

Dedicatoria

En primer lugar, me gustaría dedicar este trabajo a mi familia, por apoyarme en todas las etapas de mi vida y hacer posible que pudiera estudiar este grado.

También quiero hacer mención especial a Neus, por soportarme, animarme y entenderme en los momentos que más lo he necesitado.

Por último, a mis amigos y compañeros con los que he compartido estos años repletos de anécdotas y buenos momentos, ya que sin ellos no habría sido igual.

1. Introducción

En los últimos años, hemos sido testigos de como los medios de comunicación, como es el caso de la televisión, han incluido Internet, y en particular las redes sociales entre los canales de difusión de sus contenidos. Ante el crecimiento imparable de los usuarios que hacen uso de las redes sociales, las cadenas televisivas se han visto obligadas a diseñar un nuevo formato donde el usuario/público forma parte del transcurso del programa. En este nuevo formato, que integra la filosofía 2.0, el usuario puede participar e intercambiar opiniones mientras el programa se va desarrollando[1].

Los patrones de consumo y la relación con los medios se han visto alterados durante los últimos años debido a la irrupción de los medios digitales. La digitalización de nuestro entorno no solo ha modificado los canales, sino que también ha sido alterado el propio contenido, los medios, su producción y sobretodo, su propio consumo. Este cambio, estimulado por el crecimiento de la cultura digital y el uso de los dispositivos móviles, ha conseguido que “todo contenido al que denominamos *offline* está unido en directo a opiniones en redes sociales de forma bidireccional. Todo gira en torno a un contenido en vivo, que los usuarios manipulan y consumen a través del medio [...]” [1]

Estos cambios provocan el nacimiento de una nueva esfera pública que permite la actuación de manera más directa de los ciudadanos en asuntos de interés general, posibilitando canales de comunicación e interacción con programas, políticos, instituciones, marcas... así como favoreciendo la comunicación e intercambio de información entre ciudadanos sin someterse a los medios de comunicación convencionales[2].

La herramienta de *microblogging*¹ *Twitter* se ha convertido rápidamente en una de las plataformas de referencia en la creación e intercambio de opinión en Internet. Su diseño y funcionamiento se sitúan a medio camino entre blogs y las redes sociales, con mensajes de corta extensión (*Tweets*) de máximo 280 caracteres. Además, la inmediatez y la creación de debate son dos de sus características esenciales. Debido a que se trata de un

¹ “Manera de comunicación o sistema de publicación que se basa en el envío de mensajes cortos de texto, por lo general con una extensión máxima de 140 o 280 caracteres, por medio de herramientas creadas con esa misión” [6].

motor de opinión y debate, no sorprende que este punto haya despertado gran interés, sobretodo el de los analistas del periodismo y de los estudios de las campañas políticas[3].

Desde que la empresa *Twitter* publica sus datos y permite acceso a parte de su código fuente además del uso de su *API*², los investigadores han detectado que el mayor volumen mundial de *Tweets* está relacionado siempre con una retransmisión televisiva. Por este motivo, el uso de esta red social resulta de gran interés y fiabilidad en el análisis de opinión e impacto de ciertos contenidos televisivos.

Estos hábitos están presentes entre personas de todas las edades, pero en mayor medida, entre las generaciones más jóvenes. Por ejemplo, Nielsen en 2014 afirmaba que: “si se produce un incremento de los comentarios en *Twitter* de un 8,4%, el rating de ese programa de televisión mejora un 1% para ese grupo de edad.”[4] Otro informe de la CEA de 2016 afirma que: “el 85% de los jóvenes nacidos entre 1981 y 1997, utiliza las segundas pantallas mientras ve la televisión” [4]. Por tanto, estamos ante una sociedad acostumbrada al uso de las redes sociales para comentar lo que ocurre en su vida cotidiana y para comentar su experiencia televisiva[14][15].

Según *Brandwatch*, en *Twitter* hay 326 millones de usuarios activos al mes, el 80% de los usuarios activos acceden al sitio vía móvil y se envían 500 millones de *Tweets* cada día, es decir, 6000 *Tweets* por segundo [5]. Estos grandes volúmenes de datos generados pueden aportarnos mucha información si son tratados y explotados. Es por esto por lo que este proyecto pretende visualizar la información obtenida, a partir de los datos extraídos de *Twitter* sobre programas de opinión, de una manera visual y con el fin de que pueda ser utilizada por cualquier tipo de usuario.

² Application Programming Interface

1.1 Objetivos

El objetivo del proyecto es la creación de una aplicación web que permita al usuario, mediante el uso de gráficas, visualizar de manera sencilla y amigable los datos recopilados de *Twitter* sobre programas televisivos de opinión.

La finalidad de este proyecto es que la web desarrollada permita a todo tipo de usuarios, como, por ejemplo: productores televisivos, analistas de opinión, politólogos o simplemente usuarios interesados, observar qué tertulianos son los más mencionados, los que más interés generan, que programas tienen mayor audiencia, que hashtags o temas son lo más comentados e interesantes para la audiencia, la repercusión del programa en redes sociales, las horas de mayor actividad...

Para la resolución del objetivo principal, se han definido los siguientes subobjetivos:

- Análisis del impacto de los programas de televisión mediante la generación de gráficos.
- Desarrollo de una aplicación web que permite la visualización de los datos de interés para el usuario.

1.2 Estructura de la memoria

Estructuraremos la memoria de la siguiente manera:

- En el capítulo 2 se enumerarán las tecnologías utilizadas en el proyecto y se mencionarán sus principales características.
- En el capítulo 3 realizaremos un análisis del problema en el que se comentará el porque han sido utilizados los programas que se han analizado y la información disponible para la realización de este.
- En el capítulo 4 se explicará el diseño de los diferentes elementos del proyecto y el porque se han realizado de esa manera.
- En el capítulo 5 se llevará a cabo un estudio de un ejemplo concreto, que permitirá visualizar los diferentes elementos creados para un caso real.
- En el capítulo 6 contiene las conclusiones extraídas del proyecto y propuesta para posibles trabajos futuros.

2. Tecnologías utilizadas

En este capítulo se presenta el conjunto de tecnologías y librerías utilizadas para la realización del proyecto. El orden con las que las mencionaremos será: primero hablaremos de la base de datos de donde obtenemos los datos utilizados en el proyecto, en segundo lugar, el conjunto de tecnologías utilizadas en el desarrollo del *backends* y, por último, las empleadas en el diseño del *frontend*⁴.

2.1 MongoDB

MongoDB es un sistema de base de datos NoSQL que esta orientado a documentos y es de código abierto. Consideramos los sistemas NoSQL, al conjunto de sistemas que, a diferencia de los sistemas de bases de datos relacionales, no guardan su información en tablas. En este caso, MongoDB, almacena su información en objetos BSON. El BSON, es una forma binaria de JSON, la cual ocupa menos espacio y, además, es más rápida y eficiente para ser utilizada como estructura de datos de lenguajes de programación[8].

Su uso en este proyecto se debe principalmente porque los datos recopilados haciendo uso de la API de *Twitter* son devueltos en formato JSON y por su facilidad de uso y simplicidad para realizar consultas desde diferentes lenguajes de programación.



Figura 1: Logo de la base de datos MongoDB

³ Conjunto de actividades realizadas en el lado del servidor, es decir, las tareas de bases de datos y los servidores de aplicaciones que el usuario no puede visualizar en el explorador de Internet. [7]

⁴ Uso de las tecnologías con las que interactúa directamente el usuario. El objetivo es desarrollar la interfaz gráfica de usuario (GUI), buscando que sea atractiva, accesible e intuitiva. [7]

2.2 Python

Python es un lenguaje de programación interpretado con una sintaxis que favorece la legibilidad del código. Además, se trata de un lenguaje de programación fuertemente tipado con tipado dinámico, multiparadigma y multiplataforma de código abierto.

Los principales motivos por los que se ha elegido este lenguaje es por su rapidez de desarrollo, simplicidad, versatilidad, facilidad para trabajar con estructuras de datos y por su gran cantidad de librerías que nos resultan de gran utilidad durante la realización del proyecto.



Figura 2: Logo de Python

Las librerías que hemos utilizado en el desarrollo son:

2.2.1 Pymongo

Pymongo es una librería de Python que nos permite conectarnos con una base de datos MongoDB y realizar acciones CRUD⁵ de una manera muy similar a la utilizada en la consola de MongoDB, lo que facilita su aprendizaje y utilización. Se ha utilizado en este proyecto para conectar con nuestra base de datos MongoDB. El proceso de conexión solo requiere de la importación de la librería y de proporcionar por parámetro el nombre o alias asociado a la misma.

⁵ Create, Read, Update and Delete, es decir, crear, leer, actualizar y eliminar.

2.2.2 Matplotlib

Matplotlib es una biblioteca de código abierto para la generación de gráficos a partir de datos contenidos en listas o arrays del lenguaje de programación Python. Sus principales ventajas, además de la ya mencionada de ser de código abierto, es la de ser multiplataforma y que permite la generación de gráficos desde *scripts* o desde la propia consola de Python. Se ha escogido por la gran cantidad de posibilidades que nos proporciona y por la simplicidad con las que podemos construir gráficos sin perder completitud y versatilidad. Para la realización del proyecto hemos utilizado la versión 3.1.1 de Matplotlib [9].



Figura 3: Logo de Matplotlib

2.2.3 Seaborn

Seaborn es una biblioteca de visualización de datos de Python basada en Matplotlib. Matplotlib permite la personalización de las gráficas como puede ser: cambio de colores, cambios de fuentes, nuevo diseño de los ejes, definición del patrón o color del fondo. En este proyecto, se ha utilizado Seaborn porque nos proporciona de manera muy sencilla, a nuestro parecer, una visualización más atractiva a la proporcionada por defecto por Matplotlib.

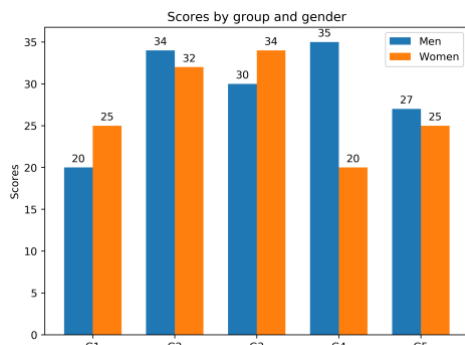


Figura 4: Gráfica generada por Matplotlib

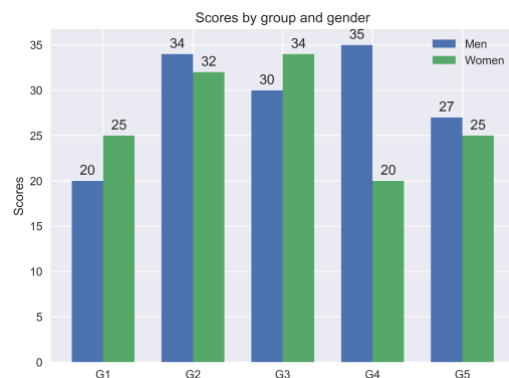


Figura 5: Gráfica generada por Matplotlib con Seaborn

2.3 Flask

Flask es un *framework*6 minimalista escrito en Python que permite la creación y despliegue de páginas web de manera sencilla y con pocas líneas de código. [11]

Entre sus ventajas destacamos que se trata de un *framework* de código abierto, que proporciona facilidades en el manejo de rutas y con gran volumen y calidad de documentación disponible. Se eligió este *framework* en lugar de otros, como puede ser *Django*, debido a su sencillez de desarrollo y porque no se requería de ninguna característica que *Flask* no pudiera proporcionarnos.



Figura 6: Logo del framework Flask

6 Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que pueden servir de base para la organización y desarrollo de software. [10]

2.4 HTML

HTML o *HyperText Markup Language* es un lenguaje de marcado para la elaboración de páginas web compuesto de una serie de etiquetas o marcas que permiten unir el contenido y la apariencia de las páginas web. [12]

Se ha elegido HTML porque es el standard a cargo del *World Wide Web Consortium* (W3C) o Consorcio WWW, organización dedicada a la estandarización de las tecnologías ligadas a la web. Se trata pues del estándar en la visualización de páginas web y del lenguaje que todos los navegadores web han adoptado.



Figura 7: Logo de HTML5

2.5 CSS

CSS o *Cascading Style Sheets* es un lenguaje de diseño gráfico y la tecnología empleada en la creación de páginas web que permite un mayor control sobre el lenguaje HTML. Permite crear hojas de estilo que definen como se debe mostrar cada elemento. El término en cascada (*Cascading*), que aparece en el nombre de la tecnología, indica que se pueden aplicar diferentes hojas de estilo sobre la misma página. [12]

Se ha utilizado esta tecnología al igual que HTML porque se trata del standard diseñado por el *World Wide Web Consortium*.



Figura 8: Logo de CSS3

2.6 Jinja2

El framework *Flask* anteriormente comentado únicamente sirve para ofrecer servicios basados en HTTP⁷. Para poder publicar contenidos dinámicos en HTML es necesario utilizar Jinja2.

Jinja2 es un lenguaje de *templates* o plantillas escrito en Python, el cual nos permite insertar datos procesados y texto predeterminado dentro de un documento de texto, motivo por el cual nos resulta de gran utilidad para la inserción de elementos dinámicos en nuestra web, como puede ser las gráficas o listados de programas, necesarios para la realización del proyecto.



Figura 9: Logo de Jinja2

⁷ HyperText Transfer Protocol, protocolo de comunicación que permite la transferencia de información en la World Wide Web.

2.7 JavaScript

JavaScript es un lenguaje de programación interpretado, imperativo, débilmente tipado, dinámico y multiplataforma, que puede ser interpretado por la mayoría de los navegadores web de la actualidad. Se puede utilizar tanto en el lado del cliente, como en el del servidor.

Para este proyecto solo se ha utilizado para la *client-side* o lado del cliente para crear interactividad dinámica en los sitios web, como es el caso de la detección de eventos de modificación o cambio en elementos HTML y su posterior acción o respuesta ante tal evento.



Figura 10: Logo de JavaScript

2.8 JQuery

JQuery es una biblioteca de JavaScript que nos permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos... JQuery, al igual que otras bibliotecas, nos ofrece funcionalidades basadas en JavaScript que de otra manera requeriría más código. De manera que con las funciones de esta biblioteca obtenemos resultados en menos tiempo y espacio. [13]



Figura 11: Logo de JQuery

3. Análisis del problema

El objetivo principal de este proyecto, como ya se ha comentado anteriormente, es el desarrollo de una página web que permita visualizar información extraída a partir de *Tweets* relacionados con programas de opinión.

Los datos utilizados para la explotación de nuestro proyecto han sido la de los dos programas de opinión y debate de referencia de ámbito nacional en España: “La Sexta Noche” del canal televisivo LaSexta y “La noche en 24 horas” del canal 24h de TVE, aunque podría utilizarse cualquier programa televisivo que utilice *Twitter*, más concretamente *hashtags*, durante su emisión.

En el caso del programa de TVE, se trata de un programa que se emite 4 días a la semana, de lunes a jueves, de 22:10 a 00:00 con una cuota de pantalla media cercana al 2,5%. El programa de LaSexta se emite semanalmente, los sábados, de 21:20 a 2:30 con una audiencia media cercana al 9,5%, y en algunos programas del 15%, convirtiéndose en un referente entre los espacios de actualidad política, análisis y debate político y social.

Los datos utilizados para la realización del proyecto son los disponibles en la BBDD mongoDB de la que se partía al inicio de este. Los datos han sido obtenidos utilizando la API de *Twitter* y se han agrupado en colecciones, cada una de las cuales representa un programa.

Por cada uno de los *Tweets*, asociados a un programa, disponemos de la siguiente información:

Visualización y seguimiento de programas de TV en Twitter

```
1 {
2   "_id" : ObjectId("566350e49913ea0e5bbb498f"),
3   "contributors" : null,
4   "coordinates" : {
5     "coordinates" : [
6       -3.7058,
7       40.4203
8     ],
9     "type" : "Point"
10  },
11  "created_at" : "Sat Dec 05 21:02:25 +0000 2015",
12  "entities" : {
13    "symbols" : [],
14    "user_mentions" : [],
15    "hashtags" : [
16      {
17        "indices" : [
18          36,
19          50
20        ],
21        "text" : "L6Ncallerajoy"
22      },
23      {
24        "indices" : [
25          103,
26          110
27        ],
28        "text" : "trndnl"
29      }
30    ],
31    "urls" : []
32  },
33  "favorite_count" : 0,
34  "favorited" : false,
35  "filter_level" : "low",
36  "geo" : {
37    "coordinates" : [
38      40.4203,
39      -3.7058
40    ],
41    "type" : "Point"
42  },
43  "id" : NumberLong(673246066041487360),
44  "id_str" : "673246066041487360",
45  "in_reply_to_screen_name" : "trendinaliaES",
46  "in_reply_to_status_id" : NumberLong(673246064359550976),
47  "in_reply_to_status_id_str" : "673246064359550976",
48  "in_reply_to_user_id" : 1179981192,
49  "in_reply_to_user_id_str" : "1179981192",
50  "is_quote_status" : false,
51  "lang" : "es",
```

Figura 12: Fragmento 1 del JSON por tuit

```
90  "retweet_count" : 0,
91  "retweeted" : false,
92  "source" : "<a href='\"http://laconversa.com\" rel='\"nofollow\">Es Tendencia en España</a>",
93  "text" : "96.563 personas podrian haber visto #L6Ncallerajoy desde su 1ª mención hasta convertirse en Tendencia. #trndnl",
94  "timestamp_ms" : "1449349345108",
95  "truncated" : false,
96  "user" : {
97    "protected" : false,
98    "profile_use_background_image" : true,
99    "default_profile" : true,
100   "default_profile_image" : false,
101   "contributors_enabled" : false,
102   "notifications" : null,
103   "name" : "Trendinalia España",
104   "follow_request_sent" : null,
105   "lang" : "es",
106   "geo_enabled" : true,
107   "profile_background_color" : "C0DEED",
108   "profile_text_color" : "333333",
109   "time_zone" : "Madrid",
110   "listed_count" : 424,
111   "utc_offset" : 3600,
112   "profile_background_image_url" : "http://abs.twimg.com/images/themes/theme1/bg.png",
113   "created_at" : "Thu Feb 14 18:39:28 +0000 2013",
114   "statuses_count" : 364603,
115   "id_str" : "1179981192",
116   "profile_sidebar_fill_color" : "DDEEF6",
117   "profile_image_url_https" : "https://pbs.twimg.com/profile_images/3256444037/7737a68b79e390c2e99f396e445a3879_normal.png",
118   "profile_link_color" : "0084B4",
119   "favourites_count" : 29,
120   "profile_background_tile" : false,
121   "verified" : false,
122   "profile_image_url" : "http://pbs.twimg.com/profile_images/3256444037/7737a68b79e390c2e99f396e445a3879_normal.png",
123   "following" : null,
124   "profile_sidebar_border_color" : "C0DEED",
125   "friends_count" : 52,
126   "id" : 1179981192,
127   "screen_name" : "trendinaliaES",
128   "url" : "http://trendinalia.com/twitter-trending-topics/spain/",
129   "description" : "Información sobre los Temas del Momento de la geografía española #TrendingTopics #trndnl",
130   "profile_background_image_url_https" : "https://abs.twimg.com/images/themes/theme1/bg.png",
131   "is_translator" : false,
132   "location" : "España",
133   "followers_count" : 34600
134 }
135 }
```

Figura 13: Fragmento 2 del JSON por tuit

De los campos anteriormente mostrados en el JSON de la figura 12 y 13 destacamos:

- **created_at**: Indica el *datetime* en formato UTC en el que el tuit fue creado.
- **entities**: Sección con elementos comunes incluidos en los *Tweets*, entre los que destacamos por ejemplo los *hashtags* utilizados en el tuit, *urls* o menciones a usuarios (*user_mentions*).
- **favorite_count**: Indica cuantas veces este tuit ha sido marcado como me gusta/favorito.
- **id**: Se trata de un identificador único del tuit.
- **lang**: Identificador correspondiente al idioma del dispositivo que emitió el tuit. Este campo puede ser nulo.
- **in_reply_to_status_id**: Si el tuit es la respuesta a otro, contendrá el id del tuit al que responde. En caso contrario será *null*.
- **in_reply_to_user_id**: Si el tuit es la respuesta a otro, contendrá el id del usuario al que responde. En caso contrario será *null*.
- **place**: Indica que el tuit está asociado a un lugar, que no necesariamente es donde ha sido generado, y que puede ser *null*.
- **retweet_count**: Número de veces que el tuit ha sido retuiteado.
- **text**: Texto o contenido del *Tweet*.
- **user**: contiene la información relativa al usuario que ha emitido el tuit:
 - **created_at**: Indica el *datetime* en formato UTC en la que se creó la cuenta.
 - **id**: Identificador único del usuario.
 - **favourites_count**: Cantidad de *Tweets* a los que el usuario a dado me gusta/favorito.
 - **verified**: *Booleans* que indica si el usuario está o no verificado.
 - **friends_count**: Número de cuentas a las que sigue el usuario.
 - **screen_name**: También conocido como *alias*, es el texto que utilizan los usuarios para identificarse, es único, pero puede modificarse.
 - **location**: Ubicación del usuario, definida por él mismo.
 - **followers_count**: Número de seguidores del usuario.

8 En computación tipo de dato lógico, que puede representar valores de lógica binaria, esto es 2 valores, que normalmente representan verdadero o falso.



4. Diseño de la solución

En este apartado describiremos la estructura de directorios de nuestro proyecto y explicaremos el desarrollo y funcionalidad de los diferentes elementos que lo conforman.

4.1 Estructura de directorios

La estructura del proyecto viene condicionada en gran medida por el funcionamiento de Flask. Hemos definido la siguiente estructura de carpetas:

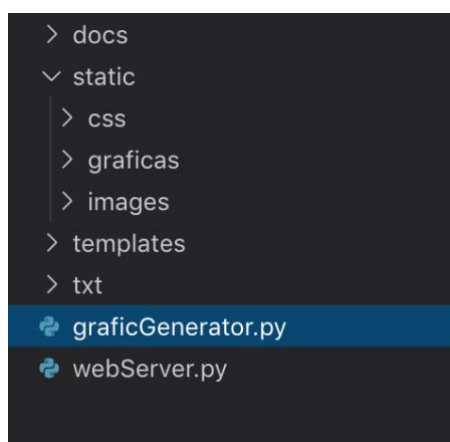


Figura 14: Estructura de directorios del proyecto

4.1.1 docs

Este directorio contiene un conjunto de ficheros de texto plano que se nos proporcionan antes de la ejecución. Existen dos ficheros por programa, aunque no necesariamente se nos deben de proporcionar, uno contiene el listado de tertulianos que asisten al programa y su nombre de usuario en *Twitter*; y en el otro se enumeran el conjunto de hashtags lanzados durante la emisión del programa y el minuto del programa en el que fueron liberados.

El nombre de estos ficheros sigue el siguiente patrón:

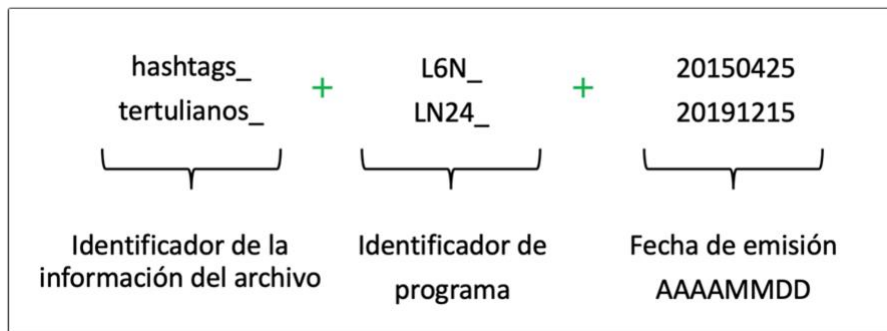


Figura 15: Patrón del nombre de los ficheros de texto del directorio "docs"

4.1.2 static

Este directorio contiene elementos relativos a la página web. Dentro de él encontramos los elementos organizados en las siguientes 3 carpetas:

1. *css*

Contiene el fichero de estilo *css* del HTML, este fichero describe como los elementos del HTML deben ser presentados.

2. *graficas*

Contiene organizadas en dos carpetas el conjunto de gráficas generadas y posteriormente mostradas en la página web. Las dos carpetas en las que se organizan estas gráficas son:

2.1. *genéricas*

Directorio en el que se almacenan el conjunto de gráficas relativas a enfrentar o comparar datos de unos programas con otros (cantidad de *Tweets* y RTs por programa, tertulianos que asisten por programa, tertulianos más recurrentes...).

2.2. específicas

En este directorio encontraremos un conjunto de gráficas iguales para cada programa y unos ficheros que contienen el código HTML para insertar el conjunto de *Tweets* más populares en la página web, pero con los datos obtenidos de las colecciones asociadas a cada uno de ellos. Este conjunto de gráficas serán las que nos permitirán conocer más información del programa seleccionado (histograma de emisión de *Tweets* durante la emisión del programa, cantidad de *Tweets* creados vs RTs, *Tweets* con mas RTs...).

Cabe mencionar que en este directorio encontramos una carpeta por programa que en su interior contiene todas las graficas comentadas y que el nombre de las subcarpetas sigue el siguiente patrón:

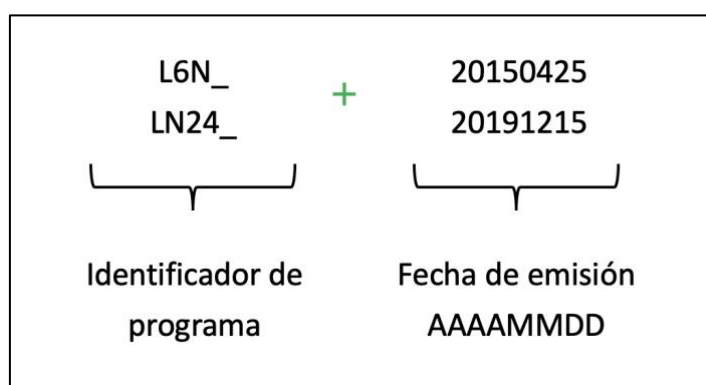


Figura 16: Patrón del nombre de subcarpetas de graficas específicas

3. images

Por último, el directorio de images posee las imágenes utilizadas en la web a nivel de iconos o elementos de diseño estáticos, como es el caso del símbolo de tuit, RT, medalla de oro para el *Tweet* con mayor número de favoritos...

4.1.3 templates

En este directorio encontramos los 3 ficheros HTML correspondientes a las 3 diferentes pestañas de la página web:

- *index*: pestaña principal dónde encontraremos el conjunto de gráficas genéricas y un listado por programa de tertulianos y hashtags de cada programa.
- *estadisticas*: fichero HTML asociado a la ventana dónde el usuario podrá visualizar las gráficas específicas asociadas al programa que seleccione.

- *info*: fichero TML estático que se corresponde con la última pestaña de la página web, la cual proporcionará información del proyecto (autor, contacto...).

4.1.4 txt

Este último directorio podría ser catalogado como auxiliar, ya que contiene un conjunto de ficheros de texto plano creado durante el proceso de generación de gráficas para su posterior lectura en el momento del despliegue de la web.

Por último, el directorio raíz contiene dos ficheros con extensión py, del lenguaje de programación Python, en los que encontramos la mayor parte del desarrollo del proyecto y que pasamos a describir:

- *graficGenerator.py*

Este fichero Python será el primero en ser ejecutado y el encargado de conectarse a la BBDD MongoDB, a la que realizará las diferentes consultas para conseguir los datos necesarios para generar el conjunto de gráficas y elementos que serán almacenados en los directorios correspondientes anteriormente mencionados.

- *webServer.py*

Con este archivo Python desplegaremos nuestra aplicación web definiendo: el conjunto de rutas del sitio web, el manejo de peticiones GET y proporcionando el conjunto de parámetros necesarios para su funcionamiento.

4.2 Desarrollo de los scripts Python

En este apartado se explica el desarrollo de los scripts Python del proyecto. En primer lugar, hablaremos sobre el encargado de la generación de gráficas (*graficGenerator.py*), y una vez desarrollado este, nos centraremos con el encargado del despliegue de la aplicación web (*webServer.py*).

4.2.1 *graficGenerator.py*

Este script Python será el encargado de conectarse a la base de datos mongoDB, realizar las consultas necesarias y generará el conjunto de gráficas y objetos pertinentes que serán almacenados en local en los directorios especificados en el apartado 4.1 y que posteriormente serán mostrados en la web.

Lo primero por tanto a realizar serán importar el conjunto de librerías necesarias para la ejecución de este. Entre ellas destacamos:

- MongoClient de pymongo: nos permite crear un objeto conexión a mongoDB sobre la que realizamos las consultas.
- Pyplot de matplotlib: necesaria para la generación de gráficas de matplotlib desde el lenguaje Python.
- Json: para el manejo de objetos JSON con los que trabaja MongoDB.
- Timedelta de datetime: nos facilita el cálculo con horas realizado para la generación de un histograma.
- Os: librería necesaria para la escritura y lectura de ficheros en la máquina en la que se va a ejecutar el script.

Una vez enumeradas las librerías de mayor importancia, destacamos la existencia de dos funciones que son definidas en este script y utilizadas por el mismo y por el script de *webServer.py* que mencionaremos más adelante. Estas funciones son:



1. ***getHashtags(nombreFichero, cod)***: Esta función, a partir de un fichero de texto con los hashtags y el tiempo transcurrido de programa hasta que el hashtag en cuestión ha sido lanzado, devuelve una lista de listas, donde cada elemento de la lista contiene una tupla (hashtag, tiempo inicio), donde tiempo inicio es el tiempo transcurrido anteriormente comentado. El parámetro *cod* es un parámetro interno definido para saber si la función ha sido llamada desde este script o desde el script encargado de desplegar el servidor web. Este parámetro *cod* es necesario debido a que si es ejecutado desde este script será necesario cargar una estructura extra con el número de hashtags emitidos en cada programa.
2. ***getTertulianos(nombreFichero, aparicionesTertuliano, cod)***: Por otro lado, esta función, a partir de un fichero de texto con los nombres y nombre de usuario de los tertulianos asistentes al programa, devuelve una lista con el nombre completo del tertuliano y su nombre de usuario en *Twitter*. El parámetro *aparicionesTertuliano* es un parámetro interno utilizado para seleccionar la variable en la que almacenamos el número de apariciones del tertuliano en el conjunto de los programas estudiados y que resulta necesario para cuando la función es llamada desde otro script. El funcionamiento del parámetro *cod* es el mismo que el explicado para la anterior función.

Por último, hay que mencionar que en caso de que el fichero de hashtags y/o tertulianos no exista, el script seguirá la ejecución ya que esta información no es obligatoria de disponer para el normal funcionamiento de la aplicación y por ello se lleva a cabo una captura del posible error surgido de intentar leer un fichero que no existe.

En caso de no existir el fichero de hashtags, se asume que el número de hashtags emitidos en el programa es 0 y en el caso de que el fichero de tertulianos no exista, asumimos que el número de tertulianos asistentes al programa también es 0. Esto puede no ser cierto, es decir, puede ser que no tengamos la información, pero si se hayan lanzado hashtags y hayan acudido tertulianos al programa. No obstante, se optó por esta decisión para no entorpecer el análisis de los otros programas que si disponen de estos ficheros.

4.2.2 webServer.py

Este script Python será el encargado de desplegar el servidor web dónde podrán mostrarse los resultados de las gráficas generadas por el script anterior. En este fichero además de definir parámetros de acceso de la página web, como puede ser el puerto, también será el encargado de resolver las peticiones web que se produzcan al mismo.

Las librerías utilizadas en este apartado y que consideramos importantes de mencionar son todas del *framework* Flask y son las siguientes:

- Flask: componente principal de la librería necesario para la utilización de esta.
- `render_template`: Función de Flask que recibe como parámetro el fichero HTML a desplegar y el conjunto de parámetros (en caso de ser necesario) que se le proporcionan. En estos parámetros es donde nosotros proveemos a la web por ejemplo de un listado de los programas, de un listado de tertulianos, una lista de hashtags, el programa seleccionado en el apartado de estadísticas detalladas...

En este script se llevará a cabo la gestión de las diferentes rutas de las que se dispone en la web y que en puntos posteriores serán explicadas con más detalle juntamente con los HTML que las componen.

Las rutas gestionadas en este apartado son las siguientes:

- Ruta home o de estadísticas genéricas ('/'): Esta ruta es la principal o inicial de la página web. El HTML asociado a esta es el encargado de mostrar las gráficas genéricas asociados al conjunto global de los programas. A esta ruta se le proporcionan como parámetros, en el método `render_template`, los siguientes parámetros:
 - El nombre del fichero HTML a renderizar: *index.html*
 - Listado de programas de la colección
 - Listado de programas “normalizado”, es una versión del nombre del programa con un formato más legible



- Número de programas que forman la colección
- Lista de listas con los diferentes hashtags de cada uno de los programas de la colección
- Lista de listas con los diferentes tertulianos de cada uno de los programas de la colección

Estos parámetros junto con el fichero HTML y Jinja2 nos permite mostrar la información necesaria en la página web de una manera dinámica e independiente de la colección o programas utilizados.

- Ruta de información (`/info`): Esta ruta es la más sencilla de la web y se compone del renderizado de un HTML estático que nos proporcionará información del proyecto. En este apartado solo se le proporciona un parámetro a la función de `render_template`: este parámetro es el nombre del fichero HTML (`info.html`) que debe ser mostrado en caso de ser esta ruta consultada.
- Ruta de estadísticas detalladas (`/estadisticas`): Esta ruta es la encargada del visionado de las estadísticas detalladas, que nos mostrará un desplegable para seleccionar el programa del que queremos visualizar sus estadísticas. En este método `render_template`, al igual que el anterior, resulta bastante simple ya se le proporcionan dos parámetros:
 - Título del fichero HTML a mostrar: `estadisticas.html`
 - Listado de programas que forman la colección y de los que podremos observar sus estadísticas específicas generadas

Cabe destacar que también se aporta un parámetro con nombre `programaSeleccionado` con valor vacío. Este parámetro será explicado a continuación con más detalle, pero cabe destacar que se le proporciona ese valor en este punto para evitar que aparezca seleccionado un programa por defecto.

- Ruta de estadísticas detalladas con parámetros ('/estadísticas/id/<programa>'): Esta ruta es la equivalente a la del apartado anterior pero cuando un programa es seleccionado entre las opciones. Una vez seleccionado, se llevan a cabo unas consultas a ficheros de texto generados con el script de *graficGenerator.py* para poder proporcionar los siguientes parámetros al *render_template*:
 - Título del fichero HTML a mostrar: *estadísticas.html*
 - Programa seleccionado: nombre del programa seleccionado del que mostrar sus estadísticas
 - Lista programas: listado del que podemos seleccionar un programa para mostrar sus estadísticas
 - Número de programa: índice que representa el programa seleccionado de la lista
 - Top 3 *Tweets* con más RTs: lista con código HTML para insertar el *blockquote*⁹ asociado a los *Tweets* con mas RTs
 - Número de RTs: lista con número de RTs asociados a los 3 *Tweets* anteriores
 - Top 3 *Tweets* con más FAVs: lista con código HTML para insertar el *blockquote* asociado a los *Tweets* con mas FAVs
 - Número de FAVs: lista con el número de FAVs asociados a los 3 *Tweets* anteriores
 - Número de *Tweets*: lista con el número de *Tweets* originales y número de los *Tweets* que son RT que forman la colección

⁹ En HTML elemento que define una sección que es citada de otra fuente (en este caso Twitter).



4.3 Desarrollo de la página web

La página web desarrollada en este proyecto se corresponde al *frontend* de la aplicación. Para ello, se han desarrollado tres ficheros HTML que representan los 3 apartados de la web: la pestaña de inicio (dónde encontramos las gráficas genéricas de la colección), el apartado de específicas (dónde podemos visualizar las estadísticas de un programa en concreto) y la ventana de información (esta ventana nos aporta información del proyecto y quien la ha desarrollado y como contactar con el mismo).

Pasamos a ver con mas detalle el desarrollo de cada una de las pestañas o ventanas mencionadas anteriormente:

4.3.1 index.html

Este archivo se corresponde con la página principal de nuestra web. En ella encontraremos el conjunto de gráficas que aglutinan información de todos los programas que forman la colección y nos permite comparar información entre ellos.

Además del menú disponible en todas las ventanas de la aplicación, que nos permite movernos entre estas, esta ventana dispone de:

- Un listado de los programas que forman la colección del tipo *detail* de HTML que nos proporciona un desplegable en caso de pulsar en un programa determinado y conocer tanto los tertulianos como los hashtags que aparecieron en el programa.
- Una imagen del tipo *img* de HTML que nos muestra el número de tertulianos y hashtags por programa. La imagen mostrada es la que ha sido generada mediante el script *graficGenerator.py*. Esta imagen se encuentra en la carpeta *static* del directorio del proyecto por necesidad del funcionamiento de *Flask* y *Jinja2*, esto ocurre con todas las imágenes mostradas en la web, aunque solo se mencionara en esta para evitar que resulte redundante.

- Una imagen del tipo *img* de HTML que nos muestra el nombre y el *username* de los 5 tertulianos con mayor número de apariciones en los programas que forman la colección.
- Una imagen del tipo *img* de HTML que nos muestra el número de interacciones generadas por programa. Con interacciones hacemos referencia a la suma de *Tweets* emitidos y *RTs* generados (la suma de ambos valores).
- Una imagen del tipo *img* de HTML que nos muestra, por separado, el número de *Tweets* creados y número de *RTs* por programa.

4.3.2 estadisticas.html

La ventana de estadísticas específicas está asociada a este fichero HTML. En ella encontraremos siempre, en la parte superior, un elemento del tipo *select* de HTML con el que seleccionaremos el nombre del programa cuyas estadísticas queremos visionar. Este selector, como anteriormente hemos mencionado, siempre estará visible en la parte superior pero cuando un programa haya sido seleccionado de la lista desplegable que aparece al ser pulsado, una función desarrollada con *JavaScript* y *JQuery* dentro del archivo HTML será ejecutada. Esta función provocará una petición web del mismo tipo que la producida al acceder a la ventana en primera instancia, pero proporcionando el identificador del programa seleccionado mediante una petición HTTP¹⁰ con método GET¹¹ (en este caso ha sido utilizado el método GET en lugar del POST¹² ya que no existía la necesidad de ocultar el parámetro de la petición HTTP). Esta petición será atendida por *Flask* en el script *webServer.py* y proporcionará la misma ventana que inicialmente ha sido mostrada y detallada, pero con los siguientes elementos a continuación del selector de programa:

- Un conjunto de dos imágenes del tipo *img* de HTML que se corresponden con el icono de *Tweet* y *RT* respectivamente, acompañado del número de elementos que existe en la base de datos asociados a cada una de estas dos posibilidades. Estos valores numéricos se proporcionan mediante parámetro desde el script de *webServer.py*.

¹⁰ Hypertext Transfer Protocol, es el protocolo de comunicación utilizado en la World Wide Web

¹¹ Método de petición de HTTP

¹² Método de petición de HTTP

- Una imagen del tipo *img* de HTML que se corresponde con un diagrama de tarta o gráfico circular en el que se puede apreciar el porcentaje que representan los *Tweets* y los *RTs* sobre el total de elementos de la colección, es decir, respecto de la suma de ambos valores.
- Una imagen del tipo *img* de HTML que nos muestra un histograma de la emisión de *Tweets* en intervalos de 30 minutos.
- Una imagen del tipo *img* de HTML que contiene el número de interacciones generadas por *hashtag*.
- Una imagen del tipo *img* de HTML que contiene el número de interacciones generadas por tertuliano, es decir, el número de veces que el tertuliano es mencionado.
- Grupo de tres elementos *blockquote*, este elemento nos proporciona una representación visual del *Tweet* además de proporcionarnos información extra y permitir acceder al *Tweet* directamente en la web de *Twitter* con solo pulsar sobre el mismo, que representan los 3 *Tweets* con mayor número de *RTs*.
- Grupo de tres elementos *blockquote*, este elemento nos proporciona una representación visual del *Tweet* además de proporcionarnos información extra y permitir acceder al *Tweet* directamente en la web de *Twitter* con solo pulsar sobre el mismo, que representan los 3 *Tweets* con mayor número de *FAVs* o *Me Gusta*.

4.3.3 info.html

Esta pestaña resulta la mas sencilla de todas, se trata de una imagen estática, generada previamente, dónde podemos encontrar el nombre y como contactar con el creador de este proyecto y una breve explicación de en que consiste.

5. Ejemplo de uso

En este apartado mostraremos, mediante ejemplos, el conjunto de información disponible para una colección de programas y un programa en concreto. La colección o *corpus* utilizado para esta prueba tiene las siguientes características:

- Compuesto por un total 4 programas.
- 2 entregas del programa de la Sexta Noche (años 2015 y 2016) del canal de televisión la Sexta.
- 2 entregas del programa de la Noche en 24 horas (año 2016) del canal de televisión 24h de Televisión Española.
- Se dispone de la información de los hashtags utilizados y los tertulianos que asisten al programa solo para los programas de la Sexta.

El programa utilizado para el análisis en específico que va a ser mostrado durante este apartado es el de la Sexta Noche, de la fecha 05/12/2015.

En la página web encontramos en la parte superior en todo momento una barra de navegación que nos indicará en qué pestaña, de las tres diseñadas, nos encontramos:

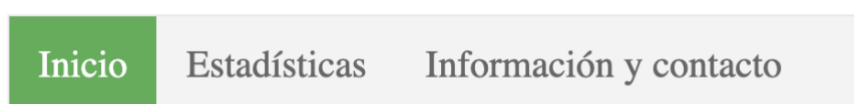


Figura 17: Barra de navegación de la página web

En la primera pestaña, la cual se corresponde con la pestaña principal, encontramos un conjunto de gráficas que nos comparan al conjunto de programas que se encuentran en la colección.

En primer lugar, en la *Figura 18*, encontramos un listado de los programas que forman esta colección. Pulsando sobre cada uno de ellos, accedemos a una lista desplegable dónde podemos consultar que tertulianos acudieron a este programa y que hashtags fueron utilizados durante la retransmisión de este:

Programas

- ▶ LN24(22/01/2016)
- ▶ LN24(14/01/2016)
- ▼ L6N(05/12/2015)

Hashtags:

- #L6Nrevilla
- #laSextanoche
- #L6Nrevilla
- #L6Ndebate7D
- #L6Ncallerajoy
- #L6Ncampaña20D
- #L6Nretoempleo
- #L6Npuertagiratoria
- #L6Nclima
- #L6Ncallerajoy

Tertulianos:

- Miguel Ángel Revilla @RevillaMiguelA
- Vicente Vallés @VicenteVallesTV
- Ana Pastor @_anapastor_
- Mariano Rajoy @marianorajoy
- Carles Campuzano @carlescampuzano
- Andrés Herzog @Herzogoff
- Antonio Hernando @AHernandoVera
- Dolors Montserrat @DolorsMM
- Ricardo Sixto @Rsixtoiglesias
- Aitor Esteban @AITOR_ESTEBAN
- Carolina Bescansa @CBescansa
- Begoña Villacís @begoñavillacis
- Jesús Maraña @jesusmarana
- Eduardo Inda @eduardoinda
- Francisco Marhuenda @pacomarhuenda
- Elisa Beni @elisabeni
- Antonio Martín Beaumont @Amartinbeaumont
- Javier Aroca @JavierArocaA
- Euprepio Padula @EuprepioPadula

- ▶ L6N(23/01/2016)

Figura 18: Listado de programas que forman la colección, desplegando los detalles asociados a uno de ellos

A continuación, en la *Figura 19* se muestra la primera de las gráficas. Se trata de un diagrama de barras dónde encontramos el número de tertulianos y hashtags de cada uno de los programas que forman la colección. Como se ve en la imagen que adjuntamos a continuación, si no se posee información de algunos de estos elementos se muestran como valor 0.

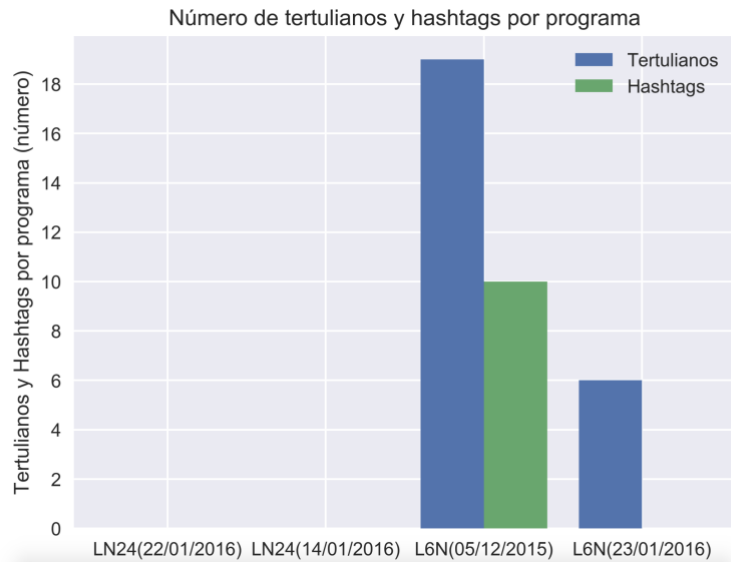


Figura 19: Gráfico del número de tertulianos y hashtags por programa

Siguiendo con esta pestaña, nos encontramos ante otro diagrama de barras en disposición horizontal, correspondiente a la *Figura 20*, el cual nos muestra a los 5 tertulianos con más apariciones en la colección y su número de apariciones. Cabe destacar, que como se trata de una pequeña colección de programas y no se dispone de información de los tertulianos de todas ellas o que en algunos no acuden tertulianos. Cabe destacar que el gráfico de la *Figura 20*, no ha sido generado con el *corpus* de la colección, ya que no todos los programas disponían del fichero de tertulianos. En su lugar, se ha generado a modo de ejemplo una casuística más real que proporciona una mejor representación del gráfico.

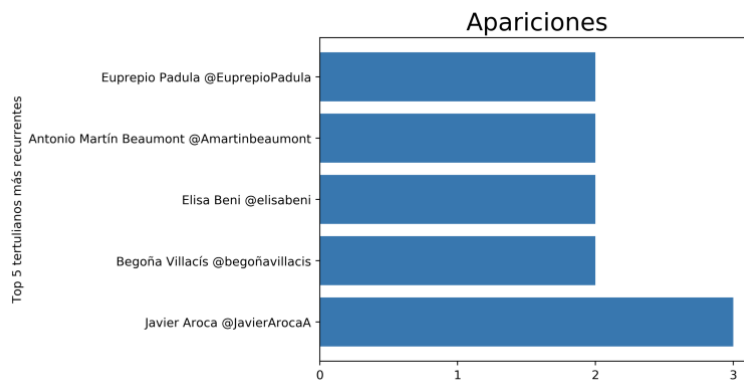


Figura 20: Gráfico de los 5 tertulianos más recurrentes y sus apariciones

En la *Figura 21*, encontramos un diagrama de barras en el cual mostramos el número total de interacciones por cada programa de la colección. Con interacciones nos referimos a la suma de *Tweets* originales, es decir, los que han sido escritos por primera vez por un usuario; y los que son *RTs* de estos.

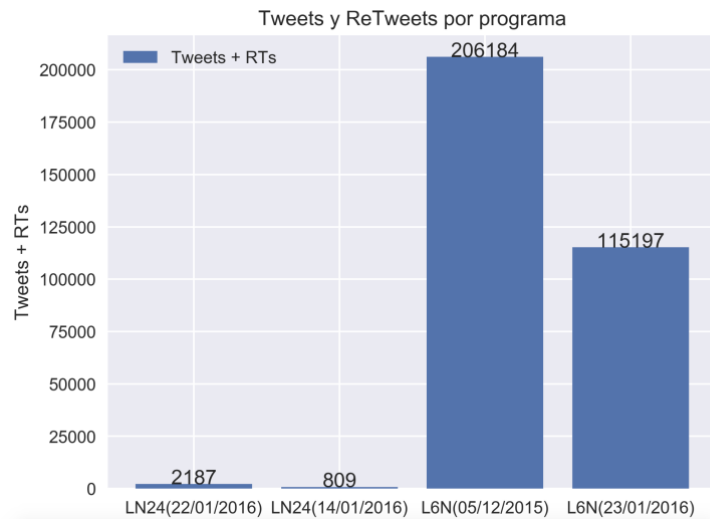


Figura 21: Gráfico de interacciones por programa

Por último, al final de esta pestaña encontramos la última de las gráficas genérica (*Figura 22*) que se corresponde con un diagrama que diferencia del total de interacciones anteriormente mostradas (*Figura 21*), cuantas se corresponden a *Tweets* originales y cuantas a *RTs*.

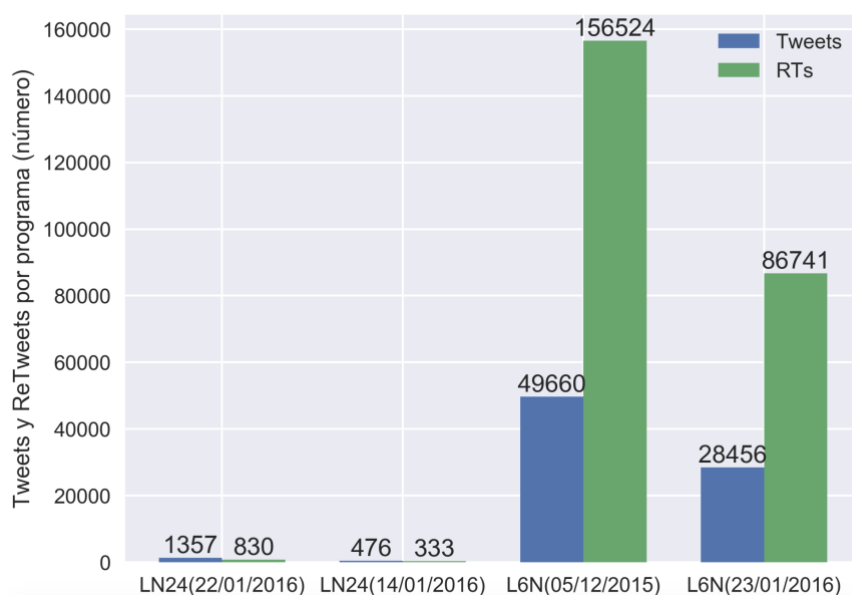


Figura 22: Gráfico de Tweets y RTs por programa

Una vez detallada la primera de las pestañas, mostremos qué elementos encontramos en la segunda. En esta, encontraremos todos los elementos necesarios para la consulta de información sobre un programa concreto. Por este motivo, el primer elemento que encontraremos será el selector desplegable que nos permite seleccionar el programa sobre el que queremos consultar información específica.



Figura 23: Selector de programa



Figura 24: Listado de programas disponibles

Hay que destacar, que una vez seleccionado un programa se realizará una recarga con las imágenes a mostrar y en el selector se nos informará del programa del cual estamos mostrando la información, pudiendo cambiar a cualquier otro cuando el usuario así lo desee.

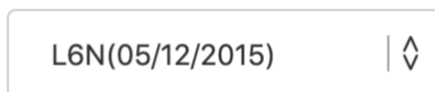


Figura 25: Programa seleccionado

Una vez seleccionado el programa, la primera información que encontraremos sobre él será la del número de *Tweets* originales que se escribieron sobre el mismo y el número de *RTs* que generaron.

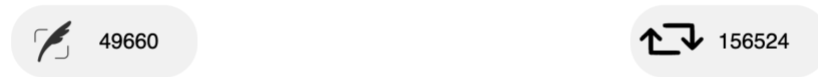


Figura 26: Número de *Tweets* escritos y *RTs* generados

A continuación, se nos muestra mediante un diagrama de tarta o gráfico circular el porcentaje que representan los *Tweets* “originales” y los *RTs* respecto al total de interacciones generadas por el programa, es decir, sobre la suma de ambos.

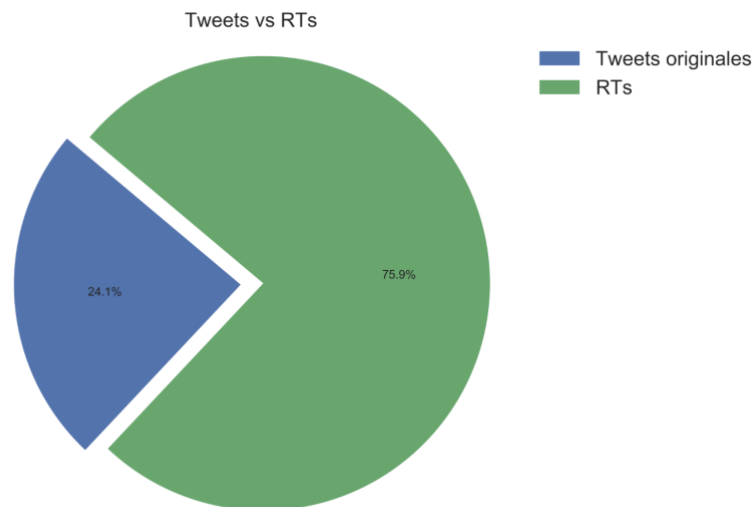


Figura 27: Gráfico del porcentaje de *Tweets* y *RTs* sobre el total de interacciones

Llegados a este punto, la página nos ofrece un histograma que nos muestra la emisión de *Tweets*, es decir, de *Tweets* originales y no *RTs*, generados en las horas previas al programa, durante la emisión y las horas posteriores a la finalización de este. Cabe destacar que algunos *Tweets* pueden ser obviados en esta representación en caso de emitirse días posteriores a la emisión del programa, ya que se realizó un análisis de las horas en los que estos eran publicados y conformaban una minoría clara y poco representativa, motivo por el cual se decidió no contemplarlos para la representación que nos ocupa.

Los *RTs* no han sido considerados para este gráfico ya que con volumetrías altas la generación del histograma resultaba imposible y solo con los *Tweets* originales se experimentó un aumento considerable del tiempo de ejecución de la generación de gráficas.

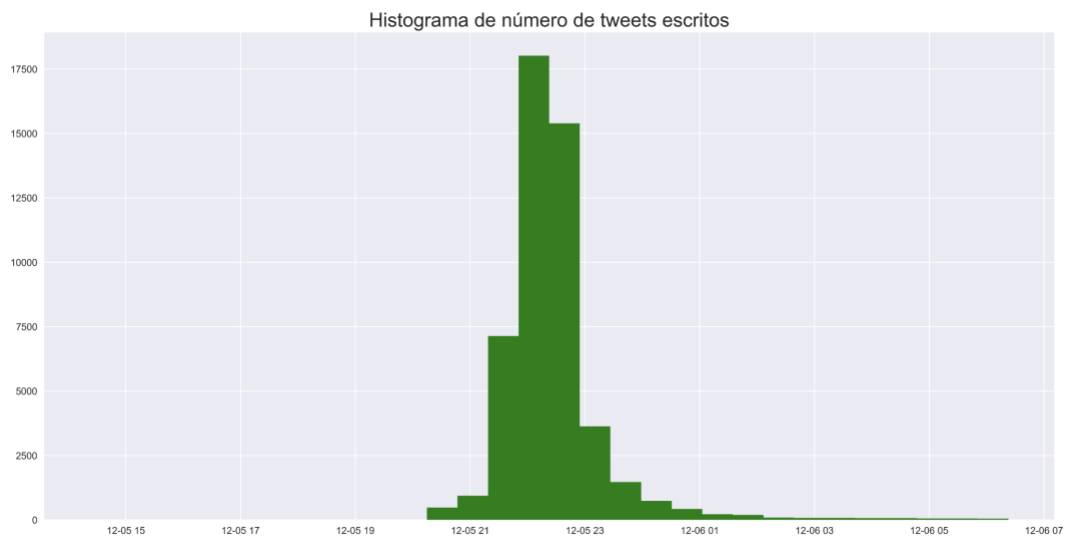


Figura 28: Histograma de creación de *Tweets* (divisiones de 30 minutos)

A continuación del histograma, de la *Figura 28*, el usuario puede encontrar un diagrama de barras que le proporciona el número total de interacciones en las que cada uno de los hashtags del programa ha sido utilizado o mencionado.

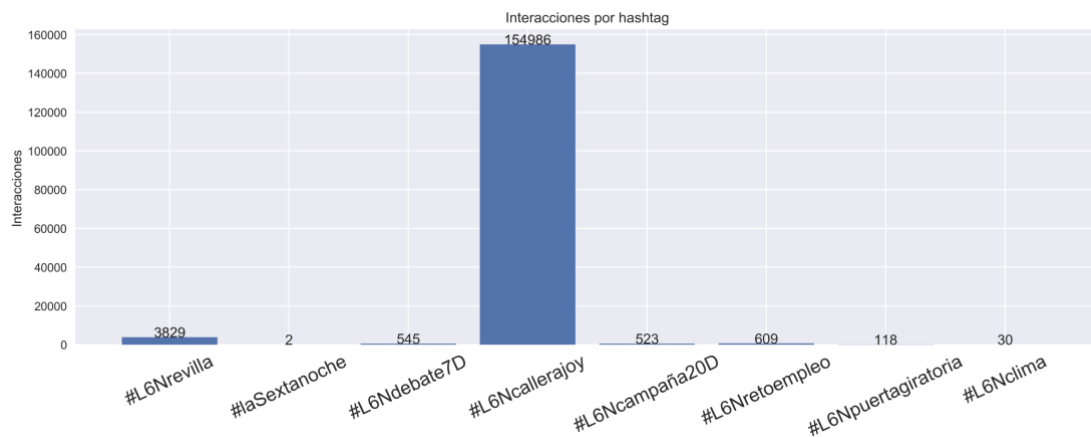


Figura 29: Gráfico de interacciones por hashtag

De manera similar al anterior gráfico (*Figura 29*), encontramos un diagrama de barras donde se nos muestran el total de interacciones dónde se mencionan a cada uno de los tertulianos asistentes al programa. Se estableció un umbral dinámico mínimo de menciones para que un tertuliano aparezca en el gráfico para facilitar su legibilidad.

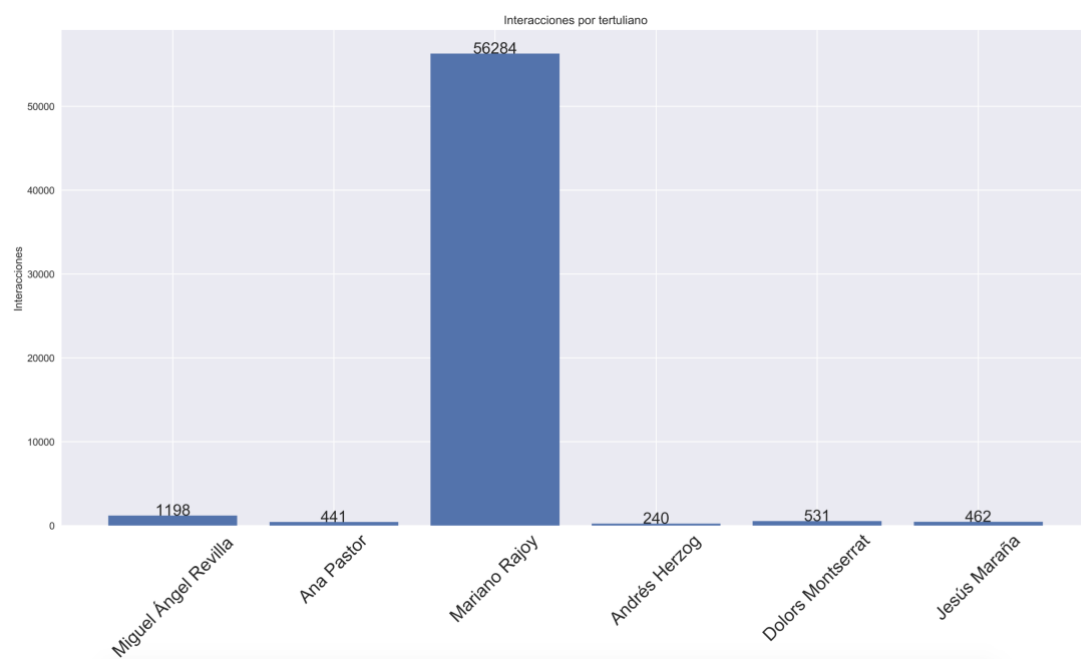


Figura 30: Gráfico de interacciones por tertuliano

Para concluir esta ventana de gráficas específicas encontramos dos bloques con los 3 *Tweets* con mayor número de *RTs* y con mayor número de *Me gusta* o *Favoritos*. Para ello se han generado los siguientes elementos:

El primero de ellos se corresponde a los 3 *Tweets* con mayor número de *RTs*. Encontramos una representación visual del mismo, la posición del *top* que ocupa y el número de *RTs* obtenidos durante la recolección de los *Tweets*, ya que es posible que después de su recopilación es posible que un usuario elimine el *RT* generado.

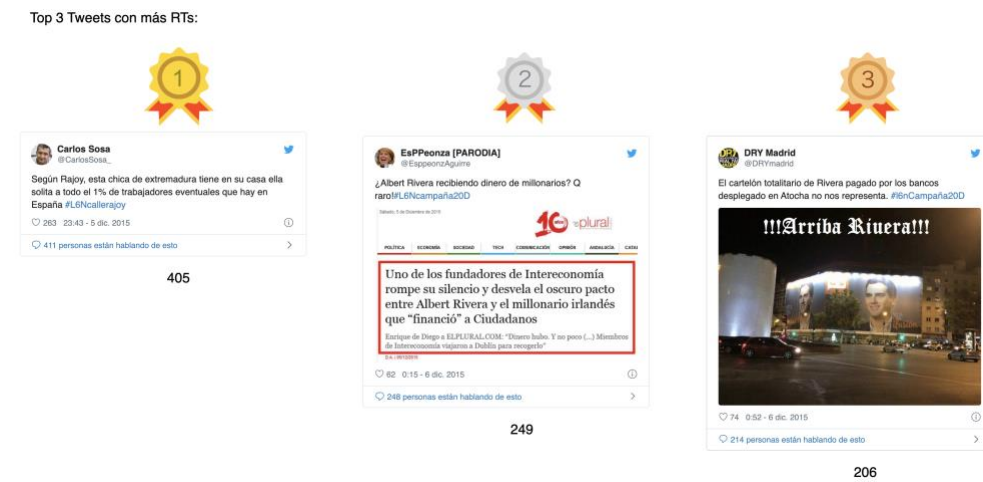


Figura 32: Top 3 de los Tweets con mayor número de RTs

En caso de pulsar sobre uno de ellos, se accede a la web de *Twitter* y al *Tweet* en concreto seleccionado por si se desea obtener más información de la mostrada o interactuar con el mismo.



Figura 33: Tweet con mayor número de RTs

En segundo lugar, de igual manera que en el anterior bloque, en este encontramos los 3 *Tweets* con mayor número de *Me gusta* o *Favoritos*. La información mostrada es la misma que en el anterior bloque y como se puede observar en el ejemplo adjuntado, el número de *Favoritos* o *Me gusta* no coincide con el número que se obtuvo en su recolección en el primero y el tercero de la clasificación que actualmente poseen menor número de *Me gusta* que en el momento en que fueron recopilados.

Top 3 Tweets con más FAVs:

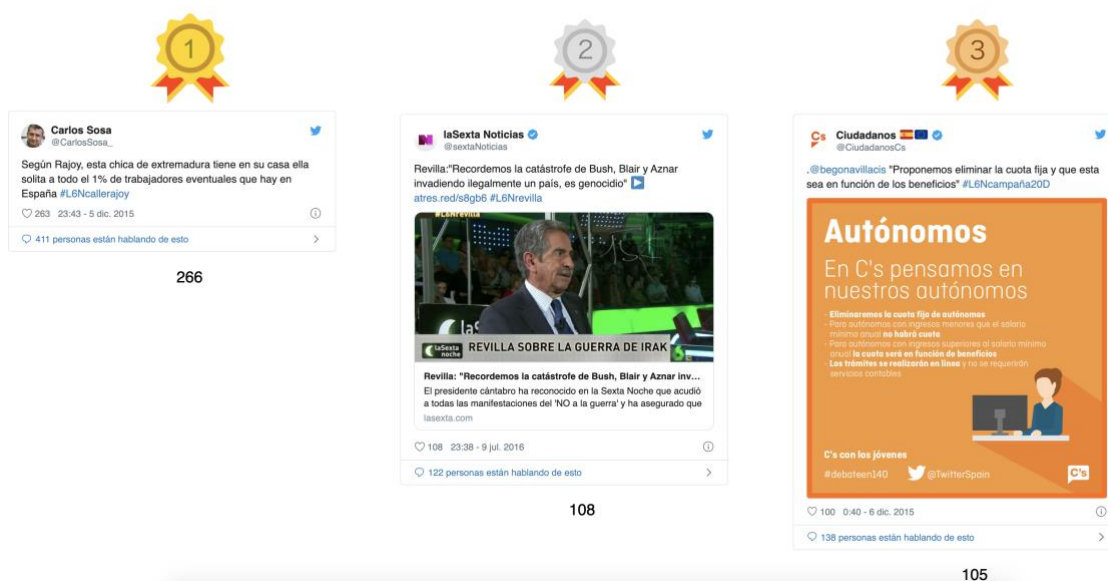


Figura 34: Top 3 de los Tweets con mayor número de FAVs

De igual manera que en el caso de los *RTs*, si pulsamos sobre uno de los *Tweets* accederemos a este en la página de *Twitter* dónde podremos interactuar con él.



Figura 35: Segundo Tweets con mayor número de Favoritos o Me gusta

Por último, en la última pestaña, encontramos solo una imagen de información de quien ha realizado el proyecto, que se corresponde a un proyecto de Trabajo de Final de Grado de la Escuela Técnica Superior de Ingeniería Informática y un correo electrónico de contacto con el creador de este.



Figura 36: Imagen de información y contacto del proyecto

6. Conclusiones y trabajos futuros

En este apartado, se comentarán las conclusiones obtenidas a la realización del proyectos y posibles líneas de futuros trabajos que surgen llegados a este punto.

Como dijimos al principio de esta memoria, el principal objetivo de este proyecto consistía en la creación de una aplicación web para la visualización de gráficas generadas a partir de unos datos almacenados en un base de datos. Por este motivo, podemos considerar que el objetivo principal del proyecto ha sido logrado, pero no de la forma esperada o no alcanzando y abarcando todos los conceptos inicialmente propuestos.

Durante el desarrollo del proyecto, nos hemos encontrado con diferentes obstáculos, que inicialmente no habían sido considerados. A continuación, destacamos algunos:

- En la mayoría de los *Tweets* no se disponía de la ubicación desde dónde se había emitido, incapacitando la realización de un análisis de las zonas dónde mayor número de interacciones habían sido generadas.
- Limitaciones computacionales de la biblioteca *Matplotlib* anteriormente comentadas en la creación de histogramas, por lo que se debió reducir la cantidad de datos y no considerar los *RTs* para poder generar el gráfico.

Pese a estos problemas, entre otros, surgidos durante el desarrollo del proyecto, se ha conseguido la consecución de este. No obstante, estos problemas, limitaciones computacionales o limitaciones temporales en cuanto al desarrollo del proyecto, nos abren un abanico de oportunidades para futuros trabajos y mejoras del proyecto entre las que destacamos:

- Migración a bibliotecas de generación de gráficas más potentes y completas, con generación de gráficos más atractivos visualmente y con los que se pueda interactuar. Una de ellas podría ser [Matplot.ly](https://matplot.ly) o [Google Charts](https://www.google.com/charts/)...



- Etiquetar los *Tweets* con *sentiment analysis*¹³ y poder mostrar gráficamente esta información. Línea de desarrollo que inicialmente se planteó, pero requería de un incremento muy considerable de desarrollo.
- Aumento de la funcionalidad de la página web, aportando la posibilidad de que el usuario proporcione un corpus de datos y que la aplicación automáticamente genere el conjunto de gráficas.
- Generación de documento PDF resumen de cualquiera de las pestañas del proyecto para facilitar su explotación y difusión por cualquier tipo de canal.
- Creación de apartados y gráficas dinámicas, sobre las que se puedan realizar búsquedas y obtener resultados calculados al instante.
- Mejora de la interfaz, de manera que esta resulte más atractiva para el usuario.

Como resumen, podemos destacar que la aplicación web desarrollada aporta las siguientes utilidades: Por una parte, se puede utilizar como una herramienta visual y sencilla para estudiar el impacto de los temas tratados o de la repercusión de los invitados. Además, también se le puede otorgar una función meramente informativa, para usuarios “curiosos” que no se conforman solo con el contenido emitido en televisión y quieren profundizar en el impacto de los asuntos tratados y la popularidad o controversia generada por los asistentes. Por último, su consulta puede ser de gran utilidad para cadenas televisivas que deseen estudiar el impacto de su programa en redes sociales.

¹³ Técnica que hace uso de procesamiento de lenguaje natural y análisis de texto para identificar y extraer información subjetiva.

6.1 Relación del trabajo realizado con los estudios cursados

Durante la realización de este proyecto, ha sido necesario la adquisición de nuevos conocimientos en tecnologías no utilizadas hasta la fecha. Sin embargo, se han utilizado conocimientos adquiridos durante el transcurso del Grado de Ingeniería Informática, de la Universitat Politècnica de València, tanto en asignaturas comunes como en asignaturas de la rama de Computación cursada. De las asignaturas cursadas, destacamos por su aportación de conocimientos en relación con el proyecto:

- Estadística: En esta asignatura de primero de grado se nos proporcionaron los conceptos básicos de estadística. Estos conceptos nos han permitido trabajar con gran volumen de información estadística, valorar la importancia de ésta y los diferentes tipos de representación utilizados en su estudio.
- Interfaz persona computador: Durante el transcurso de esta asignatura de segundo se nos aportaron las normas de diseño y usabilidad que debería poseer cualquier interfaz de usuario para que su uso por parte del usuario final sea lo mas funcional, intuitiva y completa posible.
- Estructura de datos y algoritmos: En esta asignatura de segundo, aprendimos la necesidad o importancia de los costes computacionales y la importancia de encontrar un buen algoritmo para la resolución de problemas.
- Gestión de proyectos: Con el conocimiento adquirido en esta asignatura fuimos capaces de detectar las fases de un proyecto, como crear una planificación y como abordar un proyecto en el transcurso de vida de éste. Aunque en las prácticas y temario nos centramos en proyectos de mayor volumen, que el que puede ser un TFG, las bases de planificación para la realización de un proyecto resultan de vital importancia para su correcto transcurso.
- Tecnología de sistemas de información en la red: En esta asignatura se nos proporcionaron conceptos de MongoDB. Además, en las prácticas de la asignatura aprendimos los elementos esenciales del lenguaje JavaScript.

- Base de datos y sistemas de información: Durante esta asignatura de tercero de grado, se estudiaron los conceptos principales de bases de datos y sus tipologías. Y aunque nos centramos con las bases de datos SQL, se nos introdujeron conceptos de bases de datos NoSQL como es el caso de MongoDB utilizado para este proyecto.
- Ingeniería del Software: En la asignatura se nos aportaron los métodos, técnicas y herramientas necesarias para el desarrollo de software de calidad, necesario para la realización de cualquier proyecto de software actual.
- Sistemas de almacenamiento y recuperación de la información y Algorítmica: Estas asignaturas de Computación, de tercer y cuarto curso respectivamente, estaban centradas en el conocimiento y diseño de algoritmos. En estas asignaturas utilizamos y aprendimos los elementos más importantes del lenguaje de programación Python con el que se ha desarrollado la mayor parte del proyecto.

Bibliografía

1. *El uso de hashtags en Twitter por parte de los programas de televisión españoles.* **Castello Martínez, Araceli.** Logroño : I Congreso Internacional “Comunicación y Sociedad Digital”, 2013.
2. *Audiencia social versus audiencia creativa: caso de estudio Twitter.* **Deltell Escolar, Luis.** 1, Madrid : Universidad Complutense de Madrid, 2014, Vol. 20. 1134-1629.
3. *Los issues de los candidatos en Twitter durante la campaña de las elecciones generales de 2011.* **Zugasti, Ricardo y Sabes, Fernando.** 38, Vitoria-Gasteiz : Revista de Estudios de Comunicación del País Vasco, 2015, Vol. 20. 1137-1102.
4. *Los programas de televisión deportivos y su relación con la audiencia social en Twitter en España.* **Gallardo Camacho, Jorge, Lavin, Eva y Fernandez , Paula.** 71, Santa Cruz de Tenerife : Revista Latina de Comunicación Social, 2016. 1138-5820.
5. **Smith, Kit.** 58 estadísticas de Twitter que debes saber. *Brandwatch.* [En línea] Brandwatch, 29 de Abril de 2019. [Citado el: 20 de Octubre de 2019.] <https://www.brandwatch.com/es/blog/58-estadisticas-twitter/>.
6. **Jimenez, Marimar.** El fenómeno Twitter engancha a las compañías. *CincoDías.* 2009.
7. *Modelo de procesos para el desarrollo del front-end en aplicaciones web.* **Valdivia Caballero, Jose Jesus.** 009, Lima : Ulima, 2016. 1993-4912.
8. *Conociendo Big Data.* **Camargo Vega, Juan Jose, Camargo Ortega, Jonathan Felipe y Joyanes Aguilar, Luis .** 38, s.l. : Revista Facultad de Ingeniería, 2014, Vol. 24. 0121-1129.
9. Matplotlib. *Matplotlib 3.1.1 Documentation.* [En línea] 19 de Octubre de 2019. [Citado el: 23 de Octubre de 2019.] <https://matplotlib.org/3.1.1/contents.html>.
10. **Galindo Haro, José María .** *Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE.* 2008.
11. Flask. **The Pallets Projects.** [En línea] [Citado el: 1 de Noviembre de 2019.] <https://palletsprojects.com/p/flask/>.
12. Programación de aplicaciones web. **Repositorio Institucional de la Universidad de Alicante. RUA.** [En línea] 2002. [Citado el: 1 de Noviembre de 2019.] http://rua.ua.es/dspace/bitstream/10045/16995/1/sergio_lujan-programacion_de_aplicaciones_web.pdf.
13. Introducción a AJAX. **uniwebsidad.**

14. *Audiencia social en Twitter: hacia un nuevo modelo de consumo televisivo.*

Florencia Claes, Lluís Deltell. 36, Universidad Ramón Llull : Trípodos, 2015. 1138-3305.

15. *El Impacto del Rating Televisivo Sobre la Actividad en Twitter: Evidencia para Chile Sobre la Base del Evento Teletón 2012.* **Fabrega, Jorge y Vega, George.** 33, Santiago de Chile : Cuad.inf, 2012. 0719-367X.