



Trabajo de Fin de Grado

Monitorización en Facebook de la actividad de la marca, la popularidad y las emociones

Facebook monitoring of brand activity, popularity and emotions

Jorge Alonso Hernández
Grado en Ingeniería Informática

La Laguna, 6 de junio de 2018

Dña. **Rosa María Aguilar China**, con N.I.F. 43.778.956-C Catedrática de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

D. **Jesús Miguel Torres Jorge**, con N.I.F. 43.826.207-Y profesor Contratado Doctor de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

CERTIFICAN

Que la presente memoria titulada:

“Monitorización en Facebook de la actividad de la marca, la popularidad y las emociones”

ha sido realizada bajo su dirección por D. Jorge Alonso Hernández, con N.I.F. 42.221.527-J.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 6 de junio

Agradecimientos

En primer lugar, quiero agradecer a ambos tutores por ofrecerme la posibilidad de realizar este trabajo, así como la ayuda y el apoyo recibido durante el desarrollo del mismo.

También quiero agradecer a toda mi familia y amigos por los ánimos y el apoyo que me han brindado durante todos estos años. Además, por la ayuda recibida en la recta final del proyecto.

Por último, agradecer a todos los compañeros con los que he trabajado durante todos estos años, por la experiencia vivida y por la ayuda recibida por estos para sacar el trabajo adelante.

Muchas gracias a todos.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

Resumen

Este documento contiene la memoria del Proyecto de Fin de Grado titulado “Monitorización en Facebook de la actividad de la marca, la popularidad y las emociones”, en el cual se ha desarrollado un proyecto que permite el análisis de comentarios de turismo en la plataforma de Facebook. El análisis se ha centrado en las páginas de turismo de la isla de Tenerife.

Para el desarrollo de este estudio se obtienen del texto de publicaciones y comentarios de dicha página, las palabras clave, los hashtags y las frases nominales, el idioma en el que está escrito el mensaje, las emociones y sentimientos reflejados por el mismo. A su vez se obtiene el número de me gustas y comparticiones de las publicaciones de la página.

Con todos estos datos se realiza el análisis de la página revelando los aspectos a mejorar y los aspectos que más éxito tienen para poder ser potenciados.

Palabras clave

Turismo, Facebook, Análisis de sentimientos, Big Data, Tenerife.

Abstract

This document contains the report of Final Degree Project entitled “Facebook monitoring of brand activity, popularity and emotions”, in which a project has been developed that allows the analysis of tourism’s pages on the Facebook platform. The analysis has focused on the tourism’s pages of the Tenerife island.

To the development of this analysis we obtain from the text of posts and comments of the page, the keywords, the hashtags, and the noun phrases, the language in which the message is written, the emotions, and sentiments reflected by the same. In turn we obtain the number of likes and shares of the page's posts.

With all this data, we make the analysis of the page revealing all the aspects to improve them and the aspects that are the most successful in order to be boosted.

Keywords

Tourism, Facebook, Sentiments analysis, Big Data, Tenerife

Índice general

1	Introducción	11
1.1	Antecedentes	11
1.2	Objetivos	12
1.3	Big Data	13
1.4	Análisis de sentimientos	13
2	Aspectos técnicos	15
2.1	Lenguaje de programación	15
2.2	Entorno de desarrollo	16
2.3	Base de datos	17
2.4	Sistema de control de versiones	17
2.5	API Graph Facebook	17
2.6	Watson Natural Language Understanding	18
3	Desarrollo	19
3.1	Planificación del proyecto	19
3.2	Extracción y almacenamiento de datos	19
3.2.1	Token de acceso	20
3.2.2	Extracción y almacenamiento	20
3.3	Extracción de características	23
3.3.1	Hashtags	24
3.3.2	Preprocesamiento y Etiquetado	24
3.3.3	Palabras clave	26
3.3.4	Frases nominales	27
3.3.5	Extracción	28
3.4	Análisis de los contenidos	29
3.4.1	Análisis de características	29
3.4.2	Análisis de me gustas y comparticiones	30
3.5	Análisis de sentimientos	33
3.5.1	Idiomas	37
3.5.2	Emociones	38
3.5.3	Sentimientos	38
4	Paso a producción	40
4.1	Problemas detectados	40
4.2	Puesta en marcha	41

4.3	Política de uso de datos.....	41
5	Conclusiones	44
5.1	Resultados.....	44
5.1.1	Palabras clave	44
5.1.2	Hashtags	45
5.1.3	Frases nominales	47
5.1.4	Número de me gustas.....	48
5.1.5	Número de comparticiones	52
5.1.6	Idiomas.....	52
5.1.7	Emociones.....	54
5.1.8	Sentimientos.....	56
5.2	Conclusiones	63
5.3	Aplicaciones.....	64
6	Líneas futuras	66
7	Summary and conclusions.....	67
8	Presupuesto	69
9	Referencias	70
10	Bibliografía.....	72

Índice de figuras

Ilustración 1: Código de la base de datos MongoDB	21
Ilustración 2: Código de la petición a Facebook	21
Ilustración 3: Código del bucle que recorre las publicaciones obtenidas de Facebook.....	22
Ilustración 4: Código de la creación de las listas con la información	23
Ilustración 5: Código de la creación de los <i>DataFrame</i>	23
Ilustración 6: Función de obtención de hashtags.....	24
Ilustración 7: Ejemplo de uso de la función <code>get_hashtags</code>	24
Ilustración 8: Ejemplo de uso de la función <code>preprocess</code>	25
Ilustración 9: Código de la función de etiquetado	25
Ilustración 10: Ejemplo de uso de la función <code>tag_tokens</code>	26
Ilustración 11: Código de la función que obtiene las palabras clave	26
Ilustración 12: Ejemplo de uso de la función <code>get_keywords</code>	27
Ilustración 13: Ejemplo de uso de la función <code>get_keywords</code> sólo de adjetivos	27
Ilustración 14: Código de la función que obtiene las frases nominales	27
Ilustración 15: Ejemplo de uso de la función <code>get_noun_phrases</code>	28
Ilustración 16: Código de la función que extrae las características.....	28
Ilustración 17: Ejemplo de uso de la función <code>execute_pipeline</code>	29
Ilustración 18: Código de la función que muestra la nube de palabras.....	29
Ilustración 19: Código de la función que muestra los mensajes en los que aparece una determinada palabra	30
Ilustración 20: Ejemplo de uso de la función <code>print_verbatims</code>	30
Ilustración 21: Código de la creación de los <code>datetime</code>	31
Ilustración 22: Código que muestra la progresión de me gustas y comparticiones.....	31
Ilustración 23: Código de la función que genera las nubes de palabras de las características de la semana con más me gustas y comparticiones	32
Ilustración 24: Nube de palabras resultantes de utilizar la función <code>max_wordcloud</code>	32
Ilustración 25: Código de la función que realiza el análisis de sentimientos con el servicio de Watson	33
Ilustración 26: Código con la llamada a la función del análisis de sentimientos.....	34
Ilustración 27: Código con la extracción de elementos del objeto JSON devuelto por Watson Natural Language Understanding	34
Ilustración 28: Código con la obtención de las medias de los sentimientos y el análisis de características.....	35
Ilustración 29: Función que utiliza <code>plotly</code> para crear un gráfico de barras.....	35
Ilustración 30: Gráfico de barras resultantes de la función <code>bar_plotly_file</code>	36
Ilustración 31: Función que utiliza <code>plotly</code> para crear un gráfico circular	36
Ilustración 32: Gráfico circular generado por la función <code>cir_plotly_filename</code>	37
Ilustración 33: Código que muestra el gráfico de barras de los idiomas con <code>matplotlib</code>	37
Ilustración 34: Código que muestra los gráficos utilizando <code>plotly</code>	37
Ilustración 35: Código que muestra el gráfico de las emociones con <code>matplotlib</code>	38
Ilustración 36: Código que muestra el gráfico de barras y circular con <code>plotly</code>	38
Ilustración 37: Código que muestra el gráfico de barras de los sentimientos con <code>matplotlib</code>	38
Ilustración 38: Código que muestra el gráfico de barras y circular de los sentimientos con <code>plotly</code>	39
Ilustración 39: Página de Facebook creada para realizar el análisis	40

Ilustración 40: Nube de palabras con las palabras clave más utilizadas en las publicaciones	44
Ilustración 41: Nube de palabras con las palabras clave más utilizadas en los comentarios	45
Ilustración 42: Nube de palabras con los hashtags más utilizados en las publicaciones	46
Ilustración 43: Nube de palabras con los hashtags más utilizados en los comentarios	46
Ilustración 44: Nube de palabras con las frases nominales más utilizadas en las publicaciones	47
Ilustración 45: Nube de palabras con las frases nominales más utilizadas en los comentarios	47
Ilustración 46: Gráfico con el número medio de me gustas de las publicaciones	48
Ilustración 47: Nube de palabras con las palabras clave más utilizadas en las publicaciones en el día cuyas publicaciones recibieron un mayor número de me gustas	49
Ilustración 48: Nube de palabras con las palabras clave más utilizadas en los comentarios en el día cuyas publicaciones recibieron un mayor número de me gustas	49
Ilustración 49: Nube de palabras con los hashtags más utilizados en las publicaciones en el día cuyas publicaciones recibieron un mayor número de me gustas.....	50
Ilustración 50: Nube de palabras con los hashtags más utilizados en los comentarios en el día cuyas publicaciones recibieron un mayor número de me gustas.....	50
Ilustración 51: Nube de palabras con las frases nominales de las publicaciones en el día en que las publicaciones recibieron el mayor número de me gustas.....	51
Ilustración 52: Nube de palabras con las frases nominales de los comentarios en el día en que las publicaciones recibieron el mayor número de me gustas.....	51
Ilustración 53: Gráfico con el número medio de comparticiones de las publicaciones.....	52
Ilustración 54: Gráfico de barras con los idiomas utilizando matplotlib	53
Ilustración 55: Gráfico circular con los idiomas utilizando plotly	53
Ilustración 56: Gráfico de barras con los idiomas utilizando plotly	54
Ilustración 57: Gráfico de barras con las emociones utilizando matplotlib	55
Ilustración 58: Gráfico circular con las emociones utilizando plotly.....	55
Ilustración 59: Gráfico de barras con las emociones utilizando plotly	56
Ilustración 60: Gráfico de barras con los sentimientos utilizando matplotlib	57
Ilustración 61: Gráfico circular con los sentimientos utilizando plotly.....	57
Ilustración 62: Gráfico de barras con los sentimientos utilizando plotly	58
Ilustración 63: Gráfico de barras con la puntuación de cada uno de los sentimientos	59
Ilustración 64: Nube de palabras con las palabras clave de los comentarios positivos.....	59
Ilustración 65: Nube de palabras con los hashtags de los comentarios positivos	60
Ilustración 66: Nube de palabras con las frases nominales de los comentarios positivos....	60
Ilustración 67: Nube de palabras con las palabras clave de los comentarios negativos	61
Ilustración 68: Nube de palabras con los hashtags de los comentarios negativos.....	62
Ilustración 69: Nube de palabras con las frases nominales de los comentarios negativos ..	62

Índice de tablas

Tabla 1: Bibliotecas de Python utilizadas en el proyecto	16
Tabla 2: Presupuesto de las tareas desarrolladas.....	69
Tabla 3: Presupuesto de software y hardware	69

1 Introducción

El sector turístico es el principal motor de desarrollo y crecimiento de la economía en nuestro país, especialmente en Canarias, siendo la principal fuente de ingresos de las islas. Por tanto, se debe dedicar la mayor parte de los esfuerzos en conservar y aumentar el turismo de las islas, haciendo llegar a los posibles turistas del mundo que este destino es algo más que playas, que posee más encantos.

Dado que vivimos en la “era de la tecnología” deberíamos ser capaces de conseguir usar estas tecnologías para atraer y retener al turista en las islas. Siendo las redes sociales, las que nos brindan la mejor oportunidad de dar a conocer nuestro destino al resto del mundo, ya que la primera acción que realizan los turistas, hoy en día, es la de buscar su destino en internet y visitar los perfiles que pueda haber de estos en las redes sociales como Facebook, Twitter, etc.

Por tanto, en este proyecto nos centraremos en analizar la situación de las principales páginas turísticas de la isla de Tenerife en la plataforma de Facebook y comprobar si se está haciendo un uso adecuado de estas y la aceptación que tiene por el público que visita la isla, con el fin de mejorar este servicio y la imagen que se tiene de la isla en las redes sociales

1.1 Antecedentes

Facebook es una de las fuentes de información más grande actualmente, se generan billones de publicaciones, comentarios y reacciones de diferentes contenidos, de los cuales podemos obtener mucha información que se podrá utilizar de forma beneficiosa en una empresa o institución. Por tanto, el saber manejar toda esta información, que se considera el Big Data, se ha convertido en un elemento fundamental para estas.

Uno de los elementos fundamentales a la hora de analizar las redes sociales, hoy en día, es el análisis de sentimientos de texto, el cual es muy útil para extraer de forma automática el sentimiento asociado a empresas o instituciones por parte del público, para poder planificar la estrategia a llevar a cabo, es decir, potenciar determinados aspectos si el sentimiento generado es positivo o contrarrestarlo en caso de que sea negativo.

El trabajar con este Big Data va a permitir que las instituciones y empresas sean capaces de identificar nuevas oportunidades, conduciendo a movimientos de negocio más inteligentes y eficientes. De esta forma conseguiremos una reducción

de costes, una mejor toma de decisiones y la creación de nuevos productos y servicios que satisfagan las necesidades de los clientes.

Es por esto por lo que las grandes empresas ya han implantado el análisis del Big Data para mejorar los productos o servicios ofrecidos al observar el impacto que se genera por sus clientes y cada vez son más las empresas que empiezan a trabajar con esto y a aprovechar las ventajas que les proporcionan.

1.2 Objetivos

El objetivo principal de este proyecto es el de analizar las principales páginas de turismo de la isla de Tenerife en la plataforma de Facebook y comprobar la aceptación que tiene por parte de los usuarios de esta plataforma y por los visitantes de la isla.

Se analizarán, tanto de publicaciones como de comentarios de estas, las palabras clave utilizadas, las frases nominales y los hashtags, empleados, así se podrá comprobar cuales son las palabras clave que más se utilizan y cuáles son los hashtags más utilizados en las publicaciones y si los usuarios utilizan estos hashtags o prefieren utilizar otros.

Por otro lado, se analizarán el número de me gustas y de veces que se ha compartido de cada publicación, obteniendo el periodo en el que más me gustas o más veces se ha compartido una publicación, también obtendremos las palabras clave, hashtags y frases nominales de este periodo, así se sabrá cuáles son los de más éxito.

Se determinará también el idioma en el que está escrito cada comentario, ya que puede ser interesante realizar diferentes publicaciones en determinados idiomas dado el número de comentarios recibidos en dicho idioma además de obtener la cantidad de visitantes o interesados en venir a las islas de una determinada lengua.

Por último, se analizarán las emociones y los sentimientos generados en los comentarios por parte de los usuarios. Por un lado, obtendremos las emociones expresadas en los comentarios las cuales pueden ser:

- Alegría
- Tristeza
- Miedo
- Ira
- Asco

Y, por otro lado, obtendremos el sentimiento expresado en los comentarios por parte de los usuarios, los cuales pueden ser:

- Positivo
- Negativo
- Neutro

1.3 Big Data

El Big Data se define como el conjunto de datos que no pueden ser procesados o analizados utilizando las herramientas tradicionales, debido a su tamaño, complejidad y velocidad de crecimiento.

Hoy en día es fundamental que las empresas utilicen el Big Data a su favor, con el fin de prosperar, ya que este les proporciona una cantidad de nuevas oportunidades que deben aprovechar. Gracias a este Big Data, las empresas pueden saber la popularidad que tienen entre sus clientes, comprobando que características de la empresa son las mejor valoradas y que se deben potenciar, así como aquellas que son las peor valoradas y que se deben de mejorar. También va a permitir descubrir clientes potenciales a los que se podría llegar y descubrir nuevos productos o servicios que los clientes reclaman. Facilitará comprobar el impacto que va a tener un determinado producto o servicio nuevo en el mercado, viendo la acogida que podría tener por parte de los clientes. Esto hará que las empresas realicen una mejor y más eficiente toma de decisiones y una considerable reducción de costos.

Distintas fuentes de Big Data las encontramos en las redes sociales, tales como Facebook, Twitter, YouTube, Instagram, etc. Estas plataformas generan una cantidad ingente de datos diariamente, ya sea en forma de comentarios, publicaciones, reacciones, me gustas, tweets, etc.

1.4 Análisis de sentimientos

El análisis de sentimientos se refiere a los diferentes métodos de procesamiento de lenguaje natural, análisis de texto y lingüística computacional para identificar y extraer información subjetiva del contenido existente en el mundo digital, como las redes sociales. El análisis de sentimientos nos va a permitir extraer de un texto si este posee una connotación positiva, negativa o neutra con una determinada puntuación.

Hoy en día los usuarios expresan sus opiniones respecto a un producto o una marca en redes sociales, generándose una cantidad ingente de opiniones que pueden ser analizadas y utilizadas en favor de dicha marca o producto. El poder automatizar el proceso de analizar cada una de esas opiniones y comprobar el sentimiento que genera en los usuarios va a permitir ver si la opinión de los usuarios es positiva o negativa, en cuyo caso se deben tomar las medidas necesarias para conseguir crear un sentimiento positivo en los usuarios.

2 Aspectos técnicos

2.1 Lenguaje de programación

Durante el desarrollo del Trabajo de Fin de Grado se ha utilizado como lenguaje de programación, Python [1] debido a que posee bastantes bibliotecas que nos van a simplificar el trabajo a la hora de realizar el programa.

Python es un lenguaje de programación de código abierto que es interpretado, funcional y orientado a objetos. Tiene la poderosa capacidad de contar con una sintaxis limpia y clara siendo bastante comprensible y legible para el ser humano.

2.1.1 Bibliotecas utilizadas

Durante el desarrollo de este proyecto se ha contado con la ayuda de numerosas librerías las cuales han aportado funcionalidades que han facilitado el trabajo, por ejemplo, a la hora de mostrar los resultados por pantalla, cómo para el manejo de los datos o el procesamiento de lenguaje natural.

Nombre	Descripción	Referencia
requests	Biblioteca que permite realizar solicitudes HTTP en Python.	[2]
json	Módulo que permite codificar y decodificar objetos en el formato de intercambio de datos JSON.	[3]
pymongo	Distribución de Python que contiene herramientas para trabajar con la base de datos MongoDB.	[4]
pandas	Biblioteca de código abierto que proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento y fácil de usar para el lenguaje de programación Python.	[5]
re	Módulo que proporciona operaciones de coincidencia de expresiones regulares similares a las del lenguaje de programación Perl.	[6]
nltk	Plataforma para construir programas en Python que trabajen con datos en lenguaje natural. Proporciona interfaces fáciles de usar junto con un conjunto de bibliotecas de texto para clasificación, tokenización, etiquetado, análisis y razonamiento semántico.	[7]
	Paquete fundamental para la computación científica en Python, que contiene: Un poderoso objeto de	

numpy	matriz N-dimensional, funciones sofisticadas, herramientas para integrar código C / C++ y Fortran y álgebra lineal útil, transformada de Fourier y capacidades de números aleatorios	[8]
matplotlib	Biblioteca de trazado 2D de Python que permite la publicación de figuras de calidad en una variedad de formatos y entornos interactivos en todas las plataformas.	[9]
itertools	Módulo de Python que incorpora funciones que devuelven objetos iterables.	[10]
datetime	Módulo que proporciona clases para manipular fechas y horas.	[11]
ggplot	Sistema de trazado para Python basado en R's y gramática de gráficos.	[12]
wordcloud	Biblioteca que genera nubes de palabras en Python.	[13]
plotly	La biblioteca gráfica de Python Plotly permite la posibilidad de hacer gráficos online interactivos y de calidad, permite realizar gráficos de barras, gráficos de líneas, gráficos circulares, histogramas, etc.	[14]

Tabla 1: Bibliotecas de Python utilizadas en el proyecto

Para instalar todos los paquetes mencionados anteriormente se ha utilizado el sistema de gestión de paquetes pip [15]. Para facilitar la instalación de todos los paquetes se ha creado un fichero llamado **requirements.txt** dónde se incluyen todas las bibliotecas que se instalarán ejecutando el comando:

```
pip install -r requirements.txt
```

2.2 Entorno de desarrollo

Como entorno de desarrollo, se ha utilizado principalmente Visual Studio Code[16] para el desarrollo del código del script. Visual Studio Code es un editor de código fuente, desarrollado por Microsoft, ligero pero potente que se ejecuta en el escritorio y que está disponible para Windows, MacOS y Linux. Viene con soporte para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes como Python, Java y C++.

Por otra parte, también se ha utilizado Jupyter Notebook[17] para desarrollar, depurar y probar determinadas funcionalidades del script sin necesidad de ejecutar

el script completo. Jupyter Notebook es una aplicación web de código abierto que permite crear y compartir documentos con código fuente.

2.3 Base de datos

Como base de datos, donde almacenaremos toda la información obtenida de Facebook, ya sean las publicaciones hechas por la página analizada, como los comentarios de dichas publicaciones, utilizaremos MongoDB[18], que es una base de datos NoSQL, orientado a documentos, de código abierto que ofrece alto rendimiento, alta disponibilidad y escalado automático.

En MongoDB las bases de datos contienen colecciones de documentos, que son análogas a las tablas de las bases de datos. En MongoDB los datos se almacenan como documentos BSON, donde BSON es una representación binaria de documentos JSON. Dichos documentos están compuestos por un conjunto ordenado de claves que tienen asociados valores y que se corresponden con alguna estructura de datos típica de los lenguajes de programación como otros documentos, matrices, listas, etc.

2.4 Sistema de control de versiones

El sistema de control de versiones utilizado a lo largo del proyecto es Git[19] dado a que es una herramienta bastante cómoda y por la experiencia que se tiene de este sistema frente al resto como pueda ser Apache Subversion[20].

Por otro lado se ha utilizado GitHub[21] para el alojamiento del proyecto, ya que es un servicio de alojamiento basado en web para los proyectos que utilizan el sistema de control de versiones Git. Debido a la incertidumbre generada por la compra de esta plataforma por Microsoft, se han guardado copias del proyecto en otras plataformas como son Bitbucket[22] y GitLab[23].

El enlace al repositorio donde se aloja el proyecto en GitHub es el siguiente:

<https://github.com/alu0100767803/TFG>

2.5 API Graph Facebook

La API Graph[24] de Facebook es la herramienta que va a permitir que la aplicación pueda realizar tareas de lectura y escritura en la red social Facebook. Se trata de

una API de bajo nivel basada en HTTP que las aplicaciones pueden emplear para consultar datos, publicar historias nuevas, administrar anuncios, subir fotos y realizar varias tareas adicionales mediante programación.

Para realizar estas tareas, se va a necesitar un identificador de acceso, el cual es una cadena opaca que identifica a un usuario, una aplicación o una página, y que la aplicación puede utilizar para realizar llamadas a la API Graph. Dicho identificador incluye información acerca de la caducidad del identificador y la aplicación que lo ha generado

2.6 Watson Natural Language Understanding

Natural Language Understanding[25] es un servicio de Watson el cual utiliza el procesamiento de lenguaje natural para analizar las características semánticas de cualquier texto. Este servicio va a permitir extraer metadatos del texto como entidades, palabras clave, categorías, sentimientos, emociones, relaciones y roles semánticos

Para nuestro proyecto, este es el servicio que nos va a permitir realizar el análisis de sentimientos de las publicaciones y los comentarios de las páginas de Facebook. Además de los sentimientos también realizará el análisis de las emociones reflejadas en el texto, así como el idioma de este.

3 Desarrollo

3.1 Planificación del proyecto

El análisis de dichas páginas de turismo referentes a la isla de Tenerife consistirá en analizar las publicaciones hechas por estas páginas y sus comentarios en la plataforma de Facebook.

El desarrollo de dicho análisis se ha dividido en cuatro etapas principales, de las cuales, las dos primeras etapas son de extracción de datos de la API de Facebook y las otras dos son de análisis de esos datos obtenidos:

- Extracción y almacenamiento de datos
- Extracción de características
- Análisis de contenidos
- Análisis de sentimientos

3.2 Extracción y almacenamiento de datos

Esta primera etapa consiste en la extracción de datos de la API de Facebook y de su almacenamiento en la base de datos de MongoDB que se utiliza en el proyecto. Los datos que obtener son tanto las publicaciones de la página como los comentarios de cada una de estas, a su vez de cada publicación obtenemos los siguientes campos:

- **id:** Identificador único de cada publicación
- **message:** Mensaje de la publicación
- **shares:** Número de veces que ha sido compartida la publicación
- **from:** Nombre del autor de la publicación
- **created_time:** Fecha y hora de creación de la publicación
- **likes.summary(true):** Un objeto con el resumen de número de me gustas que tiene la publicación

Por parte de los comentarios, la información que se obtiene de la API de Facebook es:

- **id:** Identificador único de cada comentario
- **message:** Mensaje del comentario
- **created_time:** Fecha y hora de creación del comentario

3.2.1 Token de acceso

Para acceder a estos datos y extraerlos de la plataforma se utiliza la Graph API de Facebook la cual nos va a permitir a la aplicación realizar tareas de lectura de datos de Facebook mediante un token de acceso que generará dicha API. Por eso el primer paso para la extracción de datos es el de generar dicho token de acceso mediante los siguientes pasos:

1. Visitar el sitio web: <https://developers.facebook.com/apps/> y crear una aplicación.
2. Una vez creada la aplicación debemos acceder al sitio web: <https://developers.facebook.com/tools/explorer> y seleccionar la aplicación que se creó en el paso anterior.
3. Clicar en el botón de **Obtener identificador de acceso de usuario** y seleccionar los permisos que queremos darle a la aplicación.

Con estos pasos se genera un token de acceso que nos va a permitir descargar los datos mencionados anteriormente y poder analizar las páginas, el problema es que dicho token de acceso sólo será válido durante veinticuatro horas, pero es posible ampliarlo a dos meses y no tener que estar generando un token cada día, para eso debemos:

1. Clicar en el icono azul con una exclamación, que se encuentra al lado del token de acceso generado, y seleccionar **Abrir en la herramienta de identificadores de acceso**
2. Clicar en **Ampliar identificador de acceso**

3.2.2 Extracción y almacenamiento

Una vez obtenido el token de acceso se puede proceder a realizar las peticiones a la API de Facebook para que nos devuelva los datos pertinentes. Antes de realizar la petición debemos configurar la base de datos en la que se van a almacenar los datos, en este caso se ha utilizado MongoDB.

En la siguiente imagen se puede observar la definición de la base de datos:

```

#Base de datos
client = MongoClient('localhost:27017')
db = client.db_analyzer
db_analyzer = db.analyzer
collection_posts = db_analyzer.posts
collection_comments = db_analyzer.comments

#Limpiamos la base de datos para que no se repliquen los datos o interfieran con los de otras páginas
collection_posts.delete_many({})
collection_comments.delete_many({})

```

Ilustración 1: Código de la base de datos MongoDB

Se han definido dos colecciones de documentos, en **collection_posts** se almacena toda la información referida a las publicaciones y en **collection_comments** se almacena toda la información referida a los comentarios. Después de definir la base de datos nos aseguramos de que ambas colecciones están vacías al iniciar el script para evitar que los datos de otras páginas analizadas con anterioridad influyen en los resultados del análisis actual.

Una vez configurada la base de datos se pasa a realizar la petición a la API de Facebook, para eso debemos definir la URL a la que queremos realizar la petición y los datos que queremos obtener. En este caso se han definido dos URL, una para las publicaciones y otra para los comentarios.

En la siguiente imagen podemos ver las dos URL a las que se va a realizar las peticiones para obtener los datos:

```

url = 'https://graph.facebook.com/v2.12/'
page_url = 'https://graph.facebook.com/v2.12/%s/feed?fields=id,message,shares,created_time,likes.summary(true)' %id
print(page_url)
# Todos los campos deben de ser especificados
comments_url = 'https://graph.facebook.com/v2.12/{post_id}/comments?filter=stream&limit=100'

# Variable con el token de acceso de Facebook
params = {'access_token' : 'EAACQFfa9JeIBAI1wRhKxGmFTvH5BJ6GhmhrrPrWkvkh0iG29oiSGPocYWgxIYaDgkFapkxmn4ZB7bz0nVb6D0DQ2VynIZAT7gegZckixBaX1cRZBvp'}
posts = requests.get(page_url, params = params)
posts = posts.json()

```

Ilustración 2: Código de la petición a Facebook

- La variable **id** contiene el identificador de la página que vamos a analizar.
- La variable **page_url** contiene la dirección de la página a ser analizada y los campos que queremos obtener de dicha página, donde el identificador de la página se obtiene de la variable **id** descrita anteriormente.
- La variable **comments_url** contiene la dirección de los comentarios de las publicaciones, dónde **post_id** se sustituirá por el identificador de cada publicación de la que queramos obtener los comentarios.
- La variable **posts** contiene las publicaciones obtenidas de la petición realizada a la API de Facebook mediante la URL de la página y el token de acceso.

Ahora que tenemos todas las publicaciones almacenadas en la variable `posts`, se procede a recorrer todas las publicaciones almacenadas en la variable `posts` y obteniendo los comentarios de cada una de ellas.

En la siguiente imagen vemos el bucle que recorre el JSON con los datos de Facebook y va iterando por cada publicación, obteniendo dicha publicación y almacenándola en la base de datos, en la colección `collection_posts` y a su vez descargando los comentarios de la publicación y almacenándolos en la base de datos, para este caso en la colección `collection_comments`:

```
# Data extraction

while True:
    try:
        print("Recopilando datos...")
        ### Recupera un post
        for element in posts['data']:
            collection_posts.insert_one(element)
            ### Recupera todos los comentarios de ese post
            this_comment_url = comments_url.replace("{post_id}", element['id'])
            comments = requests.get(this_comment_url, params = params).json()
            # Recorre todos los comentarios hasta que la respuesta esté vacía, que no haya más comentarios
            while ('paging' in comments and 'cursors' in comments['paging'] and 'after' in comments['paging']['cursors']):
                ### Itera a través de todos los comentarios
                for comment in comments['data']:
                    comment['post_id'] = element['id']
                    collection_comments.insert_one(comment)
                    comments = requests.get(this_comment_url + '&after=' + comments['paging']['cursors']['after'], params = params).json()
            ### Vamos a la siguiente página en feed
            posts = requests.get(posts['paging']['next']).json()
    except KeyError as e:
        print(e)
        break

print("Datos recopilados")
```

Ilustración 3: Código del bucle que recorre las publicaciones obtenidas de Facebook

En el bucle se va sustituyendo `{post_id}` de la URL de la variable `comments_url` por el identificador de publicación de cada una de las publicaciones almacenadas en la variable `posts`.

Una vez almacenadas las publicaciones y los comentarios en la base de datos, crearemos dos listas en las cuales almacenaremos los datos de la base de datos, con los que vamos a trabajar, para eso iteramos a través del cursor de MongoDB y obtenemos los campos que necesitamos.

```

posts_data = []
comments_data = []

for doc in collection_posts.find({}):
    if 'message' in doc.keys():
        posts_data.append((doc['message'], doc['created_time'], doc['likes']['summary']['total_count'],
                           doc.get('shares', {'count': 0})['count'], doc['id']))
    else:
        pass

print(len(posts_data))

for comment in collection_comments.find({}):
    if 'message' in comment.keys():
        comments_data.append((comment['message'], comment['created_time'], comment['post_id']))
    else:
        pass

print(len(comments_data))

```

Ilustración 4: Código de la creación de las listas con la información

En el código, vemos como se recorren las dos colecciones definidas anteriormente, en dónde tenemos almacenadas las publicaciones y los comentarios, y vamos añadiendo a las listas **posts_data**, que guardará la información de las publicaciones, y **comments_data**, que guardará la información de los comentarios, los datos que necesitamos para el análisis.

Una vez almacenada la información en las listas se crearán dos objetos *DataFrame* de la biblioteca **pandas** a partir de las listas creadas con anterioridad, y le damos un nombre a cada columna para acceder de forma sencilla a la información:

```

df_posts = pandas.DataFrame(posts_data)
df_posts.columns = ['message', 'created_time', 'likes', 'shares', 'post_id']
df_comments = pandas.DataFrame(comments_data)
df_comments.columns = ['message', 'created_time', 'post_id']

```

Ilustración 5: Código de la creación de los *DataFrame*

3.3 Extracción de características

Una vez tengamos los *DataFrame* con toda la información que teníamos almacenada en la base de datos, el siguiente paso será el de extraer diferentes características de los mensajes tanto de las publicaciones como de los comentarios. Las características que vamos a extraer son:

- Hashtags
- Palabras Clave
- Frases nominales

Para obtener dichas características vamos a definir unas funciones que se van a apoyar en las bibliotecas **nlTK**, de procesamiento de lenguaje natural y re, de expresiones regulares.

3.3.1 Hashtags

Para extraer los hashtags de los mensajes de publicaciones y comentarios se ha definido una función la cual se encarga de buscar en el texto todas aquellas palabras que comiencen con el símbolo #, seguido de texto:

```
# Obtiene los hastags de los mensajes
def get_hashtags(text):
    hashtags = re.findall(r"#(\w+)", text)
    return(hashtags,)
```

Ilustración 6: Función de obtención de hashtags

Dicha función retornará una tupla, en cuya primera posición se encuentra una lista con el contenido de los hashtags sin el símbolo #. Un ejemplo de uso de esta función es:

```
get_hashtags('It was so fun and so beautiful that I want to go there again!! a beautiful volcano !! #Paradise #Teide')
(['Paradise', 'Teide'],)
```

Ilustración 7: Ejemplo de uso de la función get_hashtags

3.3.2 Preprocesamiento y Etiquetado

Antes de obtener las palabras clave y las frases nominales, los mensajes de las publicaciones y comentarios deben pasar una fase de preprocesamiento y posteriormente de etiquetado, para ello se han definido dos funciones.

La primera función, que se encarga del preprocesamiento, limpia el texto que recibe como parámetro de espacios en blanco, símbolos de puntuación y transforma las letras mayúsculas en letras minúsculas. Una vez se ha limpiado el texto, este se divide en tokens diferentes:

```
# Limpia comentarios y posts
def preprocess(text):
    # Limpieza básica
    # Limpia de espacios y signos de puntuación, y convierte en minúscula
    text = text.strip()
    text = re.sub(r'^\w\s', '', text)
    text = text.lower()

    # Divide en tokens
    tokens = nltk.word_tokenize(text)

    return(tokens,)
```

Ilustración 8: Función de preprocesamiento de texto

La función devuelve una tupla en cuya primera posición se encuentra una lista con los tokens obtenidos. Un ejemplo de uso de esta función de preprocesamiento es:

```
preprocess('It was so fun and so beautiful that I want to go there again!! a beautiful volcano !! #Paradise #Teide')

(['it',
 'was',
 'so',
 'fun',
 'and',
 'so',
 'beautiful',
 'that',
 'i',
 'want',
 'to',
 'go',
 'there',
 'again',
 'a',
 'beautiful',
 'volcano',
 'paradise',
 'teide'],)
```

Ilustración 8: Ejemplo de uso de la función preprocess

La segunda función, se encarga de realizar el etiquetado de los tokens devueltos por la función de preprocesamiento, por tanto, dicha función va a recibir por parámetro una lista de tokens. Dicha función utiliza la biblioteca **nltk** para realizar el etiquetado:

```
# Etiqueta los tokens preprocesados
def tag_tokens(preprocessed_tokens):
    pos = nltk.pos_tag(preprocessed_tokens)
    return(pos,)
```

Ilustración 9: Código de la función de etiquetado

Dicha función retorna una lista de tuplas, donde cada tupla alberga cada token y su etiqueta. Utilizando el resultado del ejemplo de la función anterior obtenemos que el resultado de esta es:

```
tag_tokens(['it', 'was', 'so', 'fun', 'and', 'so', 'beautiful', 'that', 'i', 'want', 'to', 'go', 'there', 'again',
           'a', 'beautiful', 'volcano', 'paradise', 'teide'])

([('it', 'PRP'),
 ('was', 'VBD'),
 ('so', 'RB'),
 ('fun', 'JJ'),
 ('and', 'CC'),
 ('so', 'RB'),
 ('beautiful', 'JJ'),
 ('that', 'IN'),
 ('i', 'NN'),
 ('want', 'VBP'),
 ('to', 'TO'),
 ('go', 'VB'),
 ('there', 'RB'),
 ('again', 'RB'),
 ('a', 'DT'),
 ('beautiful', 'JJ'),
 ('volcano', 'NN'),
 ('paradise', 'NN'),
 ('teide', 'NN')])
```

Ilustración 10: Ejemplo de uso de la función `tag_tokens`

3.3.3 Palabras clave

Una vez que tenemos todos los tokens con sus etiquetas correspondientes procedemos a la obtención de las palabras clave. Para ello hemos definido una función que recibe como primer parámetro una lista de tuplas con los tokens y sus etiquetas. Como segundo parámetro va a recibir el tipo de palabras clave que queremos obtener, se pueden obtener sólo los sustantivos, adjetivos o verbos, o si lo deseamos podemos obtener los tres tipos a la vez:

```
# Obtiene las palabras clave
def get_keywords(tagged_tokens, pos='all'):

    if(pos == 'all'):
        lst_pos = ('NN', 'JJ', 'VB')
    elif(pos == 'nouns'):
        lst_pos = 'NN' #Filtra por sustantivos
    elif(pos == 'verbs'):
        lst_pos = 'VB' #Filtra por verbos
    elif(pos == 'adjectives'):
        lst_pos = 'JJ' #Filtra por adjetivos
    else:
        lst_pos = ('NN', 'JJ', 'VB')

    keywords = [tup[0] for tup in tagged_tokens if tup[1].startswith(lst_pos)]
    return(keywords,)
```

Ilustración 11: Código de la función que obtiene las palabras clave

Dicha función devuelve una tupla, en cuya primera posición se encuentra una lista con las palabras clave. Un ejemplo del resultado de esta función utilizando los tokens etiquetados de la función tag_tokens es:

```
get_keywords([('it', 'PRP'), ('was', 'VBD'), ('so', 'RB'), ('fun', 'JJ'), ('and', 'CC'), ('so', 'RB'), ('beautiful', 'JJ'),
             ('that', 'IN'), ('i', 'NN'), ('want', 'VBP'), ('to', 'TO'), ('go', 'VB'), ('there', 'RB'), ('again', 'RB'),
             ('a', 'DT'), ('beautiful', 'JJ'), ('volcano', 'NN'), ('paradise', 'NN'), ('teide', 'NN')])

(['was',
 'fun',
 'beautiful',
 'i',
 'want',
 'go',
 'beautiful',
 'volcano',
 'paradise',
 'teide'],)
```

Ilustración 12: Ejemplo de uso de la función get_keywords

Un ejemplo de las palabras clave que sólo son adjetivos es:

```
get_keywords([('it', 'PRP'), ('was', 'VBD'), ('so', 'RB'), ('fun', 'JJ'), ('and', 'CC'), ('so', 'RB'), ('beautiful', 'JJ'),
             ('that', 'IN'), ('i', 'NN'), ('want', 'VBP'), ('to', 'TO'), ('go', 'VB'), ('there', 'RB'), ('again', 'RB'),
             ('a', 'DT'), ('beautiful', 'JJ'), ('volcano', 'NN'), ('paradise', 'NN'), ('teide', 'NN')], 'adjectives')

(['fun', 'beautiful', 'beautiful'],)
```

Ilustración 13: Ejemplo de uso de la función get_keywords sólo de adjetivos

3.3.4 Frases nominales

Partiendo de la lista de tokens y sus correspondientes etiquetas, podemos obtener las frases nominales, para ello se ha definido una función la cual recibe como parámetro dicha lista y va a obtener, a partir de ella, todos aquellos sustantivos que van seguidos de adjetivos y que puede estar precedido por un complemento directo:

```
# Obtiene las palabras clave
def get_keywords(tagged_tokens, pos='all'):

    if(pos == 'all'):
        lst_pos = ('NN', 'JJ', 'VB')
    elif(pos == 'nouns'):
        lst_pos = 'NN' #Filtra por sustantivos
    elif(pos == 'verbs'):
        lst_pos = 'VB' #Filtra por verbos
    elif(pos == 'adjectives'):
        lst_pos = 'JJ' #Filtra por adjetivos
    else:
        lst_pos = ('NN', 'JJ', 'VB')

    keywords = [tup[0] for tup in tagged_tokens if tup[1].startswith(lst_pos)]
    return(keywords,)
```

Ilustración 14: Código de la función que obtiene las frases nominales

Dicha función va a devolver una tupla en cuya primera posición tendremos una lista de frases nominales. Un ejemplo de la obtención de las frases nominales es:

```
get_noun_phrases([('it', 'PRP'), ('was', 'VBD'), ('so', 'RB'), ('fun', 'JJ'), ('and', 'CC'), ('so', 'RB'), ('beautiful', 'JJ'), ('that', 'IN'), ('i', 'NN'), ('want', 'VBP'), ('to', 'TO'), ('go', 'VB'), ('there', 'RB'), ('again', 'RB'), ('a', 'DT'), ('beautiful', 'JJ'), ('volcano', 'NN'), ('paradise', 'NN'), ('teide', 'NN')])  
  
(['a beautiful volcano'],)
```

Ilustración 15: Ejemplo de uso de la función `get_noun_phrases`

3.3.5 Extracción

Una vez definidas las funciones que nos van a permitir obtener las características, se ha implementado una función que se va a encargar de extraer todas estas características. Dicha función recibe como parámetro un *DataFrame* y almacenará en diferentes columnas de dicho *DataFrame* las características mencionadas anteriormente:

```
def execute_pipeline(dataframe):  
  
    # Obtiene hashtags  
    dataframe['hashtags'] = dataframe.apply(lambda x: get_hashtags(x['message']), axis=1)  
  
    # Preprocesamiento  
    dataframe['preprocessed'] = dataframe.apply(lambda x: preprocess(x['message']), axis=1)  
  
    # Etiquetado  
    dataframe['tagged'] = dataframe.apply(lambda x: tag_tokens(x['preprocessed'])[0], axis=1)  
  
    # Obtiene palabras clave  
    dataframe['keywords'] = dataframe.apply(lambda x: get_keywords(x['tagged'][0], 'all'), axis=1)  
  
    # Obtiene frases nominales  
    dataframe['noun_phrases'] = dataframe.apply(lambda x: get_noun_phrases(x['tagged'][0]), axis=1)  
  
    return(dataframe)  
  
df_posts = execute_pipeline(df_posts)  
df_comments = execute_pipeline(df_comments)
```

Ilustración 16: Código de la función que extrae las características

Una vez ejecutada la función en los *DataFrame*, `df_posts` y `df_comments`, se dispondrán de las nuevas columnas con los hashtags, palabras clave y frases nominales, para su posterior análisis. Un ejemplo del *DataFrame* resultante de utilizar esta función es:

```

execute_pipeline(df)
print(df)

0      message      hashtags \
0  It was so fun and so beautiful that I want to ... ([Paradise, Teide],)

0      preprocessed \
0  ([it, was, so, fun, and, so, beautiful, that, ...

0      tagged \
0  (([it, PRP], (was, VBD), (so, RB), (fun, JJ), ...

0      keywords      noun_phrases
0  ([was, fun, beautiful, i, want, go, beautiful,... ([a beautiful volcano],)

```

Ilustración 17: Ejemplo de uso de la función `execute_pipeline`

3.4 Análisis de los contenidos

3.4.1 Análisis de características

Para analizar el contenido extraído en el apartado anterior utilizaremos nubes de palabras que nos van a permitir comparar la frecuencia de hashtags, palabras clave y frases nominales de las publicaciones con las de los comentarios. Así podremos ver si coinciden las palabras clave utilizadas en las publicaciones y los comentarios o comprobar si los hashtags utilizados en las publicaciones se siguen por parte de los usuarios en los comentarios.

Para obtener dicha nube de palabras se ha definido una función, la cual recibe por parámetro el *DataFrame* y la columna en concreto de la que queremos obtener dicha nube de palabras, también recibe como parámetro el título que va a recibir. Esta función utilizará la biblioteca **wordcloud** para generar la nube:

```

def viz_wordcloud(dataframe, column_name, title):
    lst_tokens = list(itertools.chain.from_iterable(dataframe[column_name]))
    lst_p = []
    for phrase in lst_tokens:
        lst_p.extend(phrase)
    lst_phrases = [phrase.replace(" ", "_") for phrase in lst_p]

    CLEANING_LIST = [] # Lista con palabras que consideraremos como ruido
    # Quitamos los elementos considerados como ruido
    lst_phrases = [phrase.replace(" ", "_") for phrase in lst_phrases if not any(spam in phrase.lower() for spam in CLEANING_LIST)]
    # Eliminamos los tokens que son de una sola letra
    lst_phrases = [phrase.replace(" ", "_") for phrase in lst_phrases if len(phrase) > 1]

    wordcloud = WordCloud(font_path='./Fonts/Verdana.ttf', background_color="white", max_words=2000
        , max_font_size=40, random_state=42).generate(" ".join(lst_phrases))

    plt.figure()
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.title(title)
    plt.show()

```

Ilustración 18: Código de la función que muestra la nube de palabras

En dicha función existe una lista que va a servir para limpiar la nube de palabras que consideraremos ruido.

También se ha definido un método que nos va a permitir mostrar el mensaje, que alberga una determinada palabra clave, hashtag o frase nominal. Dicha función va a recibir por parámetro el *DataFrame* en el cual queremos buscar el texto, que en este caso sólo podrá ser en las publicaciones o en los comentarios, el número máximo de mensajes que queremos encontrar y la palabra contenida en los mensajes que queremos mostrar:

```
def print_verbatims(df, nb_verbatim, keywords):  
    verbatims = df[df['message'].str.contains(keywords)]  
    for i, text in verbatims.head(nb_verbatim).iterrows():  
        print(text['message'])
```

Ilustración 19: Código de la función que muestra los mensajes en los que aparece una determinada palabra

Dicha función mostrará por consola los mensajes que contienen la palabra que le indicamos. Por ejemplo, si queremos mostrar los comentarios que contengan la palabra **playa** se mostraría:

```
print_verbatims(df_comments, 10, 'playa')
```

```
Que playas tan bonitas!!  
Es una pena q esté tan masificado. El cemento se ha comido parte d la playa.
```

Ilustración 20: Ejemplo de uso de la función print_verbatims

3.4.2 Análisis de me gustas y comparticiones

El siguiente contenido a analizar es el número máximo de me gustas y comparticiones que tiene las publicaciones de la página. Para realizar este análisis lo primero que se va a realizar es convertir la fecha del *DataFrame* en un objeto **datetime** y seleccionara aquellos comentarios y publicaciones con fecha posterior al 1 de enero de 2015:

```

df_comments['date'] = df_comments['created_time'].apply(pandas.to_datetime)
df_comments_ts = df_comments.set_index(['date'])
df_comments_ts = df_comments_ts['2015-01-01':]

df_posts['date'] = df_posts['created_time'].apply(pandas.to_datetime)
df_posts_ts = df_posts.set_index(['date'])
df_posts_ts = df_posts_ts['2015-01-01']

```

Ilustración 21: Código de la creación de los datetime

Una vez creado este objeto, creamos un *DataFrame* que va a contener el promedio de me gustas y comparticiones por semana de las publicaciones. A partir de este *DataFrame* y utilizando la biblioteca **ggplot** se van a mostrar la progresión del número de me gustas y comparticiones por semana:

```

#Creamos un data frame que contiene la cantidad promedio de likes y shares por semana
dx = df_posts_ts.resample('W').mean()
dx.index.name = 'date'
dx = dx.reset_index()

#Mostramos el progreso de likes
p = ggplot(dx, aes(x = 'date', y = 'likes')) + geom_line()
p = p + xlab("Date") + ylab("Number of likes") + ggtitle("Facebook vacanzeTenerife Page")
print(p)

#Mostramos el progreso de shares
p = ggplot(dx, aes(x = 'date', y = 'shares')) + geom_line()
p = p + xlab("Date") + ylab("Number of shares") + ggtitle("Facebook VacanzeTenerife Page")
print(p)

```

Ilustración 22: Código que muestra la progresión de me gustas y comparticiones

También se ha definido una función, la cual muestra nubes de palabras clave, hashtags, y frases nominales de las publicaciones y los comentarios realizados durante la semana en la que más me gustas tienen las publicaciones y en la semana que más veces han sido compartidas las publicaciones, así podremos analizar cuáles son las características que más han gustado a los usuarios:

3.5 Análisis de sentimientos

Para el análisis de sentimientos se ha utilizado el servicio de Watson Natural Language Understanding. Dicho servicio es el que nos va a permitir obtener de cada mensaje:

- El idioma en el que está escrito.
- Las emociones expresadas en el texto, que pueden ser alegría, ira, tristeza, asco y miedo con una puntuación en un rango del 0 al 1.
- El sentimiento del mensaje, que puede ser positivo, con una puntuación en un rango del 0 al 1, neutro, con una puntuación de 0, y negativo, con una puntuación en un rango de -1 a 0.

Para realizar este análisis se ha hecho una función la cual realiza una llamada al servicio de Watson con los campos que se necesita, en este caso las emociones y los sentimientos, ya que el idioma lo devuelve siempre:

```
def get_sentiment(text):
    try:
        natural_language_understanding = NaturalLanguageUnderstandingV1(
            username="209d9a26-cca5-4bef-9690-33f5e8ee66da",
            password="SpWglqCnrHqD",
            version='2018-03-16')
        response = natural_language_understanding.analyze(
            text=text,
            features=Features(
                emotion=EmotionOptions(),
                sentiment=SentimentOptions()))

        return json.dumps(response, indent=2)
    except:
        return None
```

Ilustración 25: Código de la función que realiza el análisis de sentimientos con el servicio de Watson

Dicha función devuelve un objeto JSON con los campos solicitados, el cual será retornado.

Para obtener el análisis de sentimientos de cada uno de los mensajes de los comentarios se va a utilizar una copia del *DataFrame* **df_comments**, el cual llamaremos **df_nlu** y se va a realizar una llamada a la función definida anteriormente para cada mensaje del *DataFrame* para obtener el análisis de sentimiento de cada uno de ellos:

```
df_nlu = df_comments
df_nlu['nlu'] = df_nlu.apply(lambda x: get_sentiment(x['message']), axis=1)
```

Ilustración 26: Código con la llamada a la función del análisis de sentimientos

Una vez obtenido este objeto JSON para cada uno de los mensajes se procede a extraer los datos necesarios, en nuestro caso el idioma, los sentimientos y las emociones, para eso crearemos tres vectores para cada uno de los elementos, y posteriormente se transforman dichos vectores en *DataFrames* para un mejor manejo de los datos:

```
emotions = []
sentiments = []
languages = []

for index, p in enumerate(df_nlu['nlu']):
    if not p == None:
        fichero = json.loads(p)
        languages.append(fichero['language'])
        sentiments.append((df_nlu['message'][index], fichero['sentiment']['document']['label'], fichero['sentiment']['document']['score']))
        if 'emotion' in fichero.keys():
            emotions.append( fichero['emotion']['document']['emotion'])

df_emotions = pandas.DataFrame(emotions)
df_sentiments = pandas.DataFrame(sentiments)
df_sentiments.columns = ['message', 'label', 'score']
df_languages = pandas.DataFrame(languages)
df_languages.columns = ['languages']
```

Ilustración 27: Código con la extracción de elementos del objeto JSON devuelto por Watson Natural Language Understanding

También se halla la media de puntuación para los sentimientos positivos y negativos en función de la puntuación obtenida de cada uno de los mensajes, para eso se crean dos listas una para las puntuaciones positivas y otro para los negativos y se calcula la media de cada una de ellas, junto con las puntuaciones se almacena el mensaje de cada uno de los sentimientos, con el objetivo de realizar el análisis de características de los mensajes de uno y otro sentimiento para comprobar los temas relacionados a cada uno de ellos:

```

positive = []
negative = []

for index, i in enumerate(df_sentiments['score']):
    if i > 0:
        positive.append((df_sentiments['message'][index], i))
    elif i < 0:
        negative.append((df_sentiments['message'][index], i))

df_positive = pandas.DataFrame(positive)
df_positive.columns = ['message', 'score']
df_negative = pandas.DataFrame(negative)
df_negative.columns = ['message', 'score']

df_positive = execute_pipeline(df_positive)
df_negative = execute_pipeline(df_negative)

median_positive = df_positive['score'].median()
median_negative = df_negative['score'].median()

```

Ilustración 28: Código con la obtención de las medias de los sentimientos y el análisis de características

Por último, para mostrar los resultados se han utilizado las bibliotecas **matplotlib** y **plotly**. Para mostrar los resultados con la biblioteca **plotly** se han definido dos funciones, una para mostrar los resultados en forma de gráfico de barras y otra para mostrar los resultados en forma de gráfico circular:

```

def bar_plotly_file(x, y, title, filename):
    x = x
    y = y

    trace = go.Bar(
        x=x,
        y=y)

    data = [trace]
    layout = go.Layout(
        title=title,
    )
    fig = go.Figure(data=data, layout=layout)

    plotly.offline.plot(fig, filename=filename + '.html')

```

Ilustración 29: Función que utiliza plotly para crear un gráfico de barras

Esta primera función crea un fichero que muestra el gráfico de barras creado con la biblioteca **plotly**. Dicha función recibe por parámetro los elementos que van en el

eje x e y del gráfico, el título del gráfico y el nombre del fichero que se genera. Un ejemplo de gráfico circular generado por esta función es:

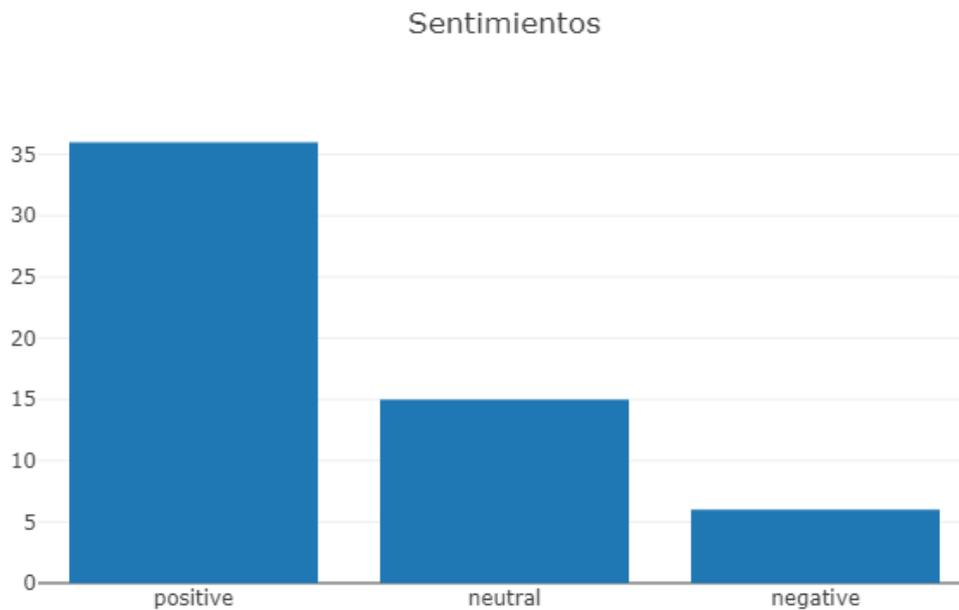


Ilustración 30: Grafico de barras resultantes de la función `bar_plotly_file`

```
def cir_plotly_file(labels, values, title, filename):  
  
    fig = {  
        'data': [{ 'labels': labels,  
                  'values': values,  
                  'type': 'pie' }],  
        'layout': { 'title': title }  
    }  
  
    plotly.offline.plot(fig, filename=filename + '.html')
```

Ilustración 31: Función que utiliza `plotly` para crear un gráfico circular

Esta segunda función crea un fichero con el gráfico circular creado con la biblioteca **plotly**. Dicha función recibe como parámetros el nombre de los elementos que se muestran en el gráfico, los valores de dichos elementos, el título del gráfico y el nombre del fichero que se va a generar con dicho gráfico. Un ejemplo de gráfico circular generado por esta función es:

Gráfico idiomas

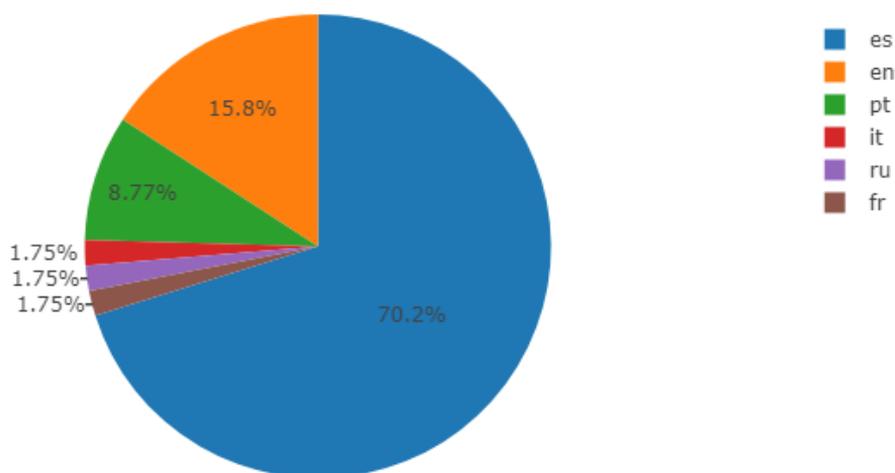


Ilustración 32: Gráfico circular generado por la función `cir_plotly_filename`

3.5.1 Idiomas

Para mostrar los resultados de los idiomas con **matplotlib**, se utiliza la función **value_counts** de la biblioteca **pandas** la cual se aplica sobre el *DataFrame* con los idiomas creado antes:

```
#Mostrando gráfica de lenguajes
df_languages['languages'].value_counts().plot(kind='bar', title='Languages')
plt.show()
```

Ilustración 33: Código que muestra el gráfico de barras de los idiomas con `matplotlib`

Para mostrarlos con **plotly**, se utilizan las funciones creadas con anterioridad y se le pasan los datos para generar dichos gráficos:

```
#Mostrando lenguajes con plotly
labels_languages = df_languages['languages'].value_counts().keys().tolist()
values_languages = df_languages['languages'].value_counts().tolist()

cir_plotly_file(labels_languages, values_languages, 'Lenguajes', 'c_lenguajes')
bar_plotly_file(labels_languages, values_languages, 'Lenguajes', 'b_lenguajes')
```

Ilustración 34: Código que muestra los gráficos utilizando `plotly`

3.5.2 Emociones

Para mostrar el resultado de las emociones con **matplotlib**, primero se calcula la media de todas las emociones y se guarda en un nuevo *DataFrame* y se muestra el resultado de las medias de cada una de ellas:

```
# Mostrando la gráfica de las emociones
df_emotions = df_emotions.apply(pandas.to_numeric, errors='ignore')
df_emotions_means = df_emotions.transpose().apply(numpy.mean, axis=1)
df_emotions_means.plot(kind='bar', legend=False)
plt.show()
```

Ilustración 35: Código que muestra el gráfico de las emociones con matplotlib

Para mostrar dichos resultados con **plotly** se utilizan las funciones definidas anteriormente, para mostrar el gráfico circular y el gráfico de barras:

```
#Mostrando emociones con plotly
labels_emotions = df_emotions_means.keys().tolist()
values_emotions = df_emotions_means.tolist()

cir_plotly_file(labels_emotions, values_emotions, 'Gráfica emociones', 'c_emociones')
bar_plotly_file(labels_emotions, values_emotions, 'Emociones', 'b_emociones')
```

Ilustración 36: Código que muestra el gráfico de barras y circular con plotly

3.5.3 Sentimientos

Para mostrar los resultados de los sentimientos con **matplotlib** se utiliza la función **value_counts** de la biblioteca **pandas** que se aplica sobre el *DataFrame* con las etiquetas del análisis de sentimientos:

```
#Mostrando gráfica de sentimientos
df_sentiments['label'].value_counts().plot(kind='bar', title="Sentiments")
plt.show()
```

Ilustración 37: Código que muestra el gráfico de barras de los sentimientos con matplotlib

Para mostrar los resultados con **plotly**, se utilizan las funciones creadas anteriormente, en este caso también se muestra el gráfico de barras con las medias de las puntuaciones de los sentimientos positivos y negativos:

```
#Mostrando sentimientos con plotly
labels_sentiments = df_sentiments['label'].value_counts().keys().tolist()
values_sentiments = df_sentiments['label'].value_counts().tolist()

cir_plotly_file(labels_sentiments, values_sentiments, 'Gráfica sentimientos', 'c_sentimientos')
bar_plotly_file(labels_sentiments, values_sentiments, 'Sentimientos', 'b_sentimientos')

labels_median = ['positive', 'negative']
values_median = [median_positive, median_negative]

bar_plotly_file(labels_median, values_median, 'Score sentiments', 'score_sentiments')
```

Ilustración 38: Código que muestra el gráfico de barras y circular de los sentimientos con plotly

4 Paso a producción

4.1 Problemas detectados

El primer problema que nos encontramos a la hora de pasar a producción el proyecto se debe a la entrada en vigor del Reglamento General de Protección de Datos[26] 25 de mayo de 2018 en la Unión Europea, que obliga a las empresas a cumplir un estricto régimen de protección de datos, con sanciones severas, de no ser cumplimentada. Se debe tener un consentimiento explícito para los datos recopilados y los fines para los que se utilizan.

Debido a esta normativa, Facebook restringe el acceso a los datos de su plataforma, de forma que, para poder acceder a ellos, las aplicaciones deben pasar **la revisión de la aplicación** en la API de Facebook. Por tanto, se ha denegado el permiso de acceder a las páginas de turismo que se usaban para las pruebas del proyecto y a las páginas que se iban a analizar una vez concluido.

Debido al escaso tiempo y a la cantidad de requisitos pedidos por la API de Facebook, para que la aplicación pudiera acceder a esos datos, se ha decidido crear y simular una página de turismo de la isla de Tenerife con el fin de mostrar cómo sería el análisis de esta página, dado que Facebook si permite el acceso a páginas que hayan sido creadas por el propio usuario.

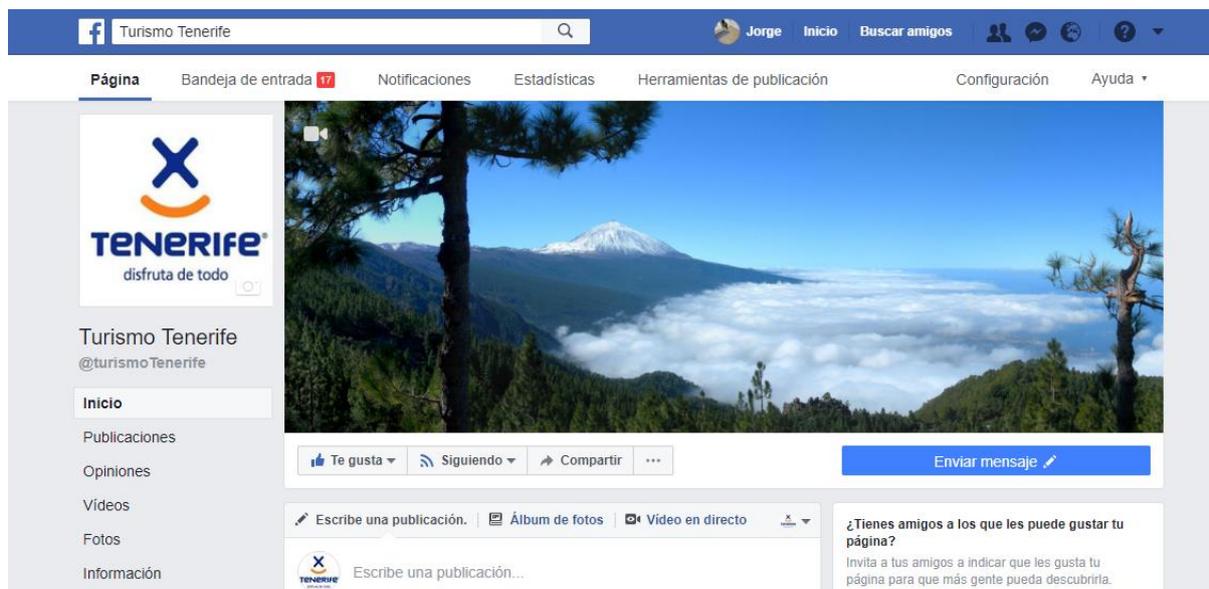


Ilustración 39: Página de Facebook creada para realizar el análisis

4.2 Puesta en marcha

Para poder poner en marcha el proyecto debemos enviar la aplicación a pasar la **revisión de la aplicación** y que Facebook la acepte, una vez Facebook la haya aceptado podremos acceder a los datos de las páginas. Para poder enviar la solicitud se deben rellenar una serie de campos de los cuales no se disponían:

- Icono de la aplicación
- URL con la política de privacidad, para el uso que vamos a darle a los datos
- La plataforma de la aplicación y la información referente al alojamiento de la aplicación, la URL si es un sitio web, identificador de la aplicación en Google Play para aplicaciones Android, entre otros datos. Se debe poder comprobar el funcionamiento de la aplicación.
- Explicación del motivo por el que se usa el acceso a los datos
- Instrucciones de cómo se usa el acceso a los datos
- Captura de vídeo en la que se muestre de forma guiada el funcionamiento de la aplicación

Una vez se haya rellenado toda esta información se podrá enviar la aplicación a revisión y si es aceptada se podrá proceder al análisis de todas las páginas de turismo de la isla de Tenerife que se deseen analizar, entre las cuales están:

- Visit Tenerife
- Tenerife
- Siam Park
- Loro Parque

4.3 Política de uso de datos

Uno de los pasos a desarrollar antes de producir la aplicación va a ser el de desarrollar una política de privacidad para los datos recopilados de Facebook, dicha política deberá ser validada por un abogado especialista. Para esta aplicación la política de privacidad podría ser la siguiente, aunque no ha pasado dicho proceso de validación:

POLÍTICA DE PRIVACIDAD

La presente Política de Privacidad establece los términos en que Analyzer usa y protege la información que es proporcionada por sus usuarios al momento de utilizar su sitio web. Esta compañía está comprometida con la seguridad de los datos de sus usuarios. Cuando le pedimos llenar los campos de información personal con la

cual usted pueda ser identificado, lo hacemos asegurando que sólo se empleará de acuerdo con los términos de este documento. Sin embargo, esta Política de Privacidad puede cambiar con el tiempo o ser actualizada por lo que le recomendamos y enfatizamos revisar continuamente esta página para asegurarse que está de acuerdo con dichos cambios.

Información que es recogida

Nuestro sitio web podrá recoger información personal, por ejemplo: Nombre, información de contacto como su dirección de correo electrónica e información demográfica. Así mismo cuando sea necesario podrá ser requerida información específica para procesar algún pedido o realizar una entrega o facturación.

Uso de la información recogida

Nuestro sitio web emplea la información con el fin de proporcionar el mejor servicio posible, particularmente para mantener un registro de usuarios, de pedidos en caso de que aplique, y mejorar nuestros productos y servicios. Es posible que sean enviados correos electrónicos periódicamente a través de nuestro sitio con ofertas especiales, nuevos productos y otra información publicitaria que consideremos relevante para usted o que pueda brindarle algún beneficio, estos correos electrónicos serán enviados a la dirección que usted proporcione y podrán ser cancelados en cualquier momento.

Analyzer está altamente comprometido para cumplir con el compromiso de mantener su información segura. Usamos los sistemas más avanzados y los actualizamos constantemente para asegurarnos que no exista ningún acceso no autorizado.

Cookies

Una cookie se refiere a un fichero que es enviado con la finalidad de solicitar permiso para almacenarse en su ordenador, al aceptar dicho fichero se crea y la cookie sirve entonces para tener información respecto al tráfico web, y también facilita las futuras visitas a una web recurrente. Otra función que tienen las cookies es que con ellas las webs pueden reconocerte individualmente y por tanto brindarte el mejor servicio personalizado de su web.

Nuestro sitio web emplea las cookies para poder identificar las páginas que son visitadas y su frecuencia. Esta información es empleada únicamente para análisis estadístico y después la información se elimina de forma permanente. Usted puede eliminar las cookies en cualquier momento desde su ordenador. Sin embargo, las cookies ayudan a proporcionar un mejor servicio de los sitios web, éstas no dan acceso a información de su ordenador ni de usted, a menos de que usted así lo quiera y la proporcione directamente. Usted puede aceptar o negar el uso de cookies, sin embargo, la mayoría de los navegadores aceptan cookies automáticamente pues sirve para tener un mejor servicio web. También usted puede cambiar la configuración de su ordenador para declinar las cookies. Si se declinan es posible que no pueda utilizar algunos de nuestros servicios.

Enlaces a Terceros

Este sitio web pudiera contener enlaces a otros sitios que pudieran ser de su interés. Una vez que usted de clic en estos enlaces y abandone nuestra página, ya no tenemos control sobre al sitio al que es redirigido y por lo tanto no somos responsables de los términos o privacidad ni de la protección de sus datos en esos otros sitios terceros. Dichos sitios están sujetos a sus propias políticas de privacidad por lo cual es recomendable que los consulte para confirmar que usted está de acuerdo con estas.

Control de su información personal

En cualquier momento usted puede restringir la recopilación o el uso de la información personal que es proporcionada a nuestro sitio web. Cada vez que se le solicite rellenar un formulario, como el de alta de usuario, puede marcar o desmarcar la opción de recibir información por correo electrónico. En caso de que haya marcado la opción de recibir nuestro boletín o publicidad usted puede cancelarla en cualquier momento.

Esta compañía no venderá, cederá ni distribuirá la información personal que es recopilada sin su consentimiento, salvo que sea requerido por un juez con un orden judicial.

Analyzer Se reserva el derecho de cambiar los términos de la presente Política de Privacidad en cualquier momento.

Hashtags en publicaciones



Ilustración 42: Nube de palabras con los hashtags más utilizados en las publicaciones

Hashtags en comentarios



Ilustración 43: Nube de palabras con los hashtags más utilizados en los comentarios

Se puede comprobar que el hashtag más utilizado en las publicaciones es también el de los comentarios, que es el de **Tenerife**, también se puede comprobar que la mayoría de los usuarios prefiere crear sus propios hashtags a seguir los que escribe la página, con la excepción de algunos como pueden ser los hashtags **Drago**, **Teide** y **Relax**.

5.1.3 Frases nominales

En las siguientes imágenes podemos ver las nubes de palabras con las frases nominales más utilizadas en las publicaciones y en los comentarios:



Ilustración 44: Nube de palabras con las frases nominales más utilizadas en las publicaciones



Ilustración 45: Nube de palabras con las frases nominales más utilizadas en los comentarios

Podemos comprobar que no existe ninguna coincidencia entre las frases nominales más utilizadas en las publicaciones de la página y las frases nominales escritas por los usuarios de Facebook en los comentarios.

5.1.4 Número de me gustas

Para comprobar las fechas en las que más me gustas reciben las publicaciones, debido a la escasa vida de la página de turismo en Facebook, no se ha podido comprobar los resultados semanalmente, por tanto, y con el fin de mostrar resultados, estos se muestran en función de los días.

En la siguiente gráfica se puede observar el número promedio de me gustas de las publicaciones en cada uno de los días en los que ha habido publicaciones. Se puede comprobar que el mayor número de me gustas se produjeron el día 12 de junio de 2018, con más de 20 me gustas en promedio:

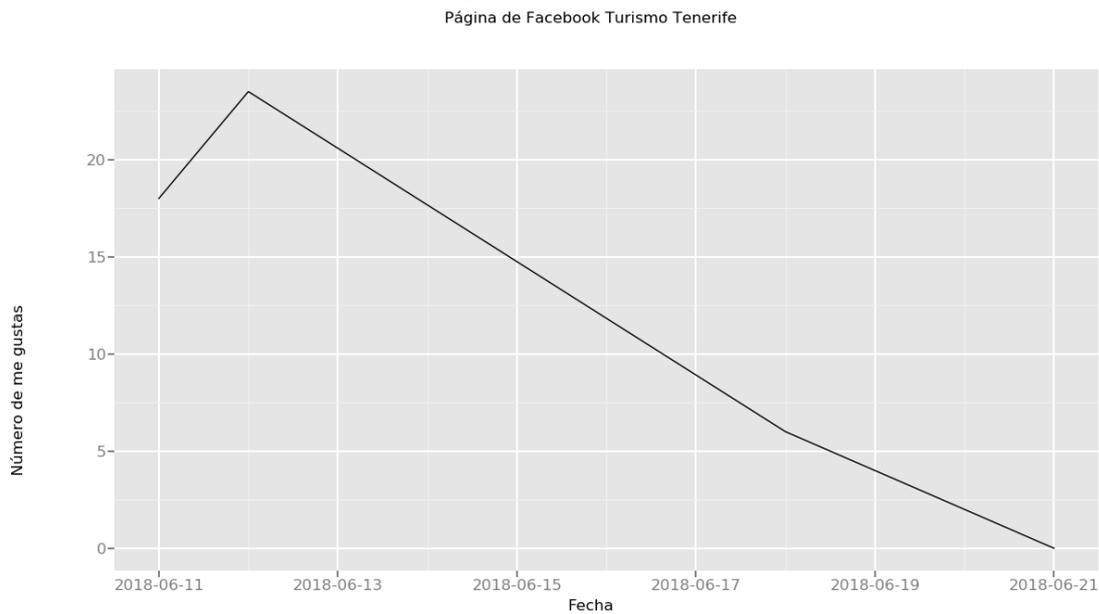


Ilustración 46: Gráfico con el número medio de me gustas de las publicaciones

Respecto a las palabras clave que se utilizan en el día que más me gustas reciben las publicaciones de las páginas nos encontramos:

Palabras clave en máximo de me gustas en publicaciones



Ilustración 47: Nube de palabras con las palabras clave más utilizadas en las publicaciones en el día cuyas publicaciones recibieron un mayor número de me gustas

Palabras clave en máximo de me gustas en comentarios



Ilustración 48: Nube de palabras con las palabras clave más utilizadas en los comentarios en el día cuyas publicaciones recibieron un mayor número de me gustas

Podemos observar que las publicaciones se centraron en los municipios de Güimar, Icod, Garachico y Los Cristianos, entre otros. Por otro lado, en los comentarios de usuario, se vuelve a comprobar en el paisaje y en lo que transmite, también nos encontramos algunas referidas al monte como pueden ser **pateo**, **chuletada** y **Teide** se puede comprobar también que no se centra en ningún municipio como si pasaba con las publicaciones.

Por otra parte, los hashtags utilizados este día son:



Ilustración 49: Nube de palabras con los hashtags más utilizados en las publicaciones en el día cuyas publicaciones recibieron un mayor número de me gustas



Ilustración 50: Nube de palabras con los hashtags más utilizados en los comentarios en el día cuyas publicaciones recibieron un mayor número de me gustas

Se puede ver, nuevamente, que el hashtag más utilizado en las publicaciones es el de **Tenerife**, y que es el más utilizado en los comentarios. También se puede comprobar que los hashtags utilizados en las publicaciones son bastante variados, en cambio, en los comentarios se vuelven a centrar en temas relacionados como la montaña, por hashtags como **Teide**, **senderismo**, **naturaleza**, **cañadas**, **pateos**.

Por último, se pueden apreciar las diferencias entre las frases nominales de las publicaciones y comentarios utilizadas este día:

Frases nominales en máximo de me gustas en publicaciones

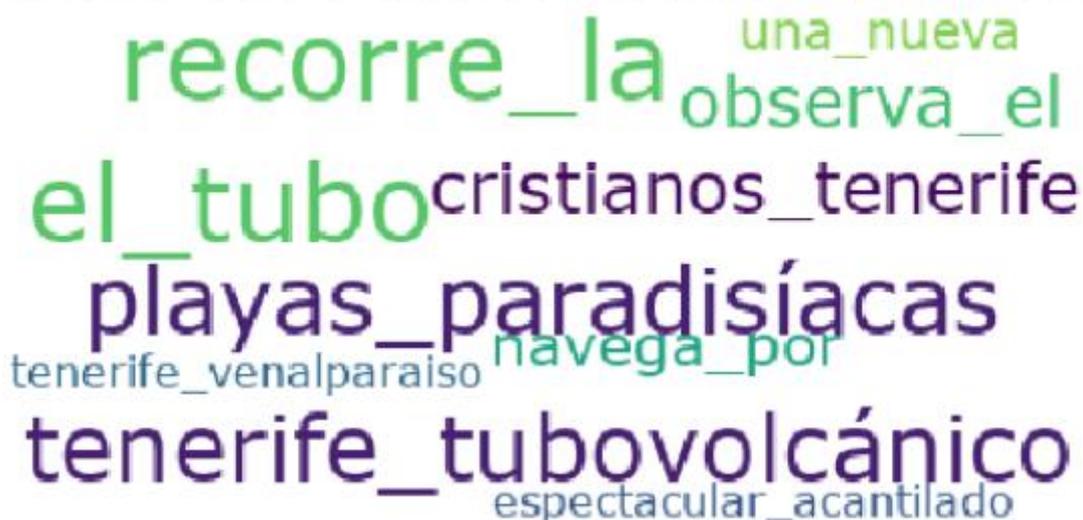


Ilustración 51: Nube de palabras con las frases nominales de las publicaciones en el día en que las publicaciones recibieron el mayor número de me gustas

Frases nominales en máximo de me gustas en comentarios



Ilustración 52: Nube de palabras con las frases nominales de los comentarios en el día en que las publicaciones recibieron el mayor número de me gustas

Se puede observar que en las publicaciones las frases nominales se han centrado en las playas paradisíacas de las islas, en el tubo volcánico y en los acantilados. En cambio, por parte de los comentarios no se ha centrado en ningún elemento reseñable quizá el más destacable son los referidos a darse un baño, ya sea en el mar o en piscinas.

5.1.5 Número de comparticiones

Al igual que sucede con el número de me gustas, para comprobar la fecha en la que más veces han sido compartidas las publicaciones, no se ha podido comprobar por semanas y se ha hecho en función de los días.

En el siguiente gráfico podemos ver el promedio de comparticiones que tienen las publicaciones en cada uno de los días en los que se han realizado publicaciones. Se puede ver que el día 12 de junio es el día donde se han recibido un mayor número de comparticiones, al igual que sucedía con los me gustas:

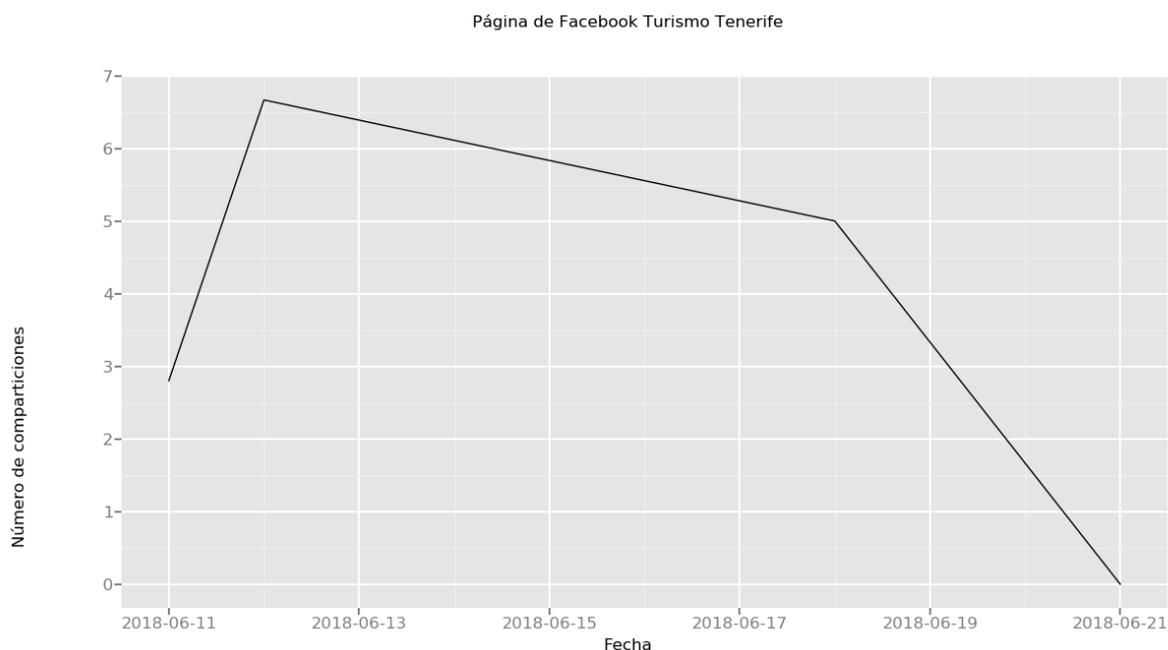


Ilustración 53: Gráfico con el número medio de comparticiones de las publicaciones

Por tanto, al ser el mismo día y las mismas publicaciones realizadas en el apartado anterior, las nubes de palabras resultantes para las palabras clave, hashtags y frases nominales de los comentarios y publicaciones de ese día, son las mismas que las mostradas en ese apartado.

5.1.6 Idiomas

Las siguientes imágenes van a mostrar los resultados obtenidos para los idiomas utilizados en los comentarios por parte de los usuarios de la página de Facebook:

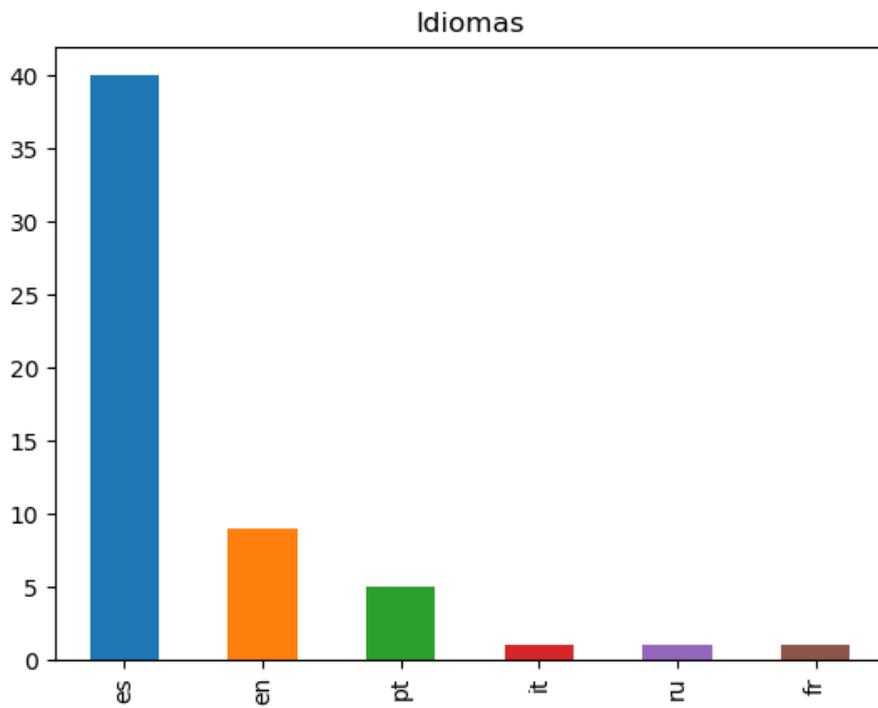


Ilustración 54: Gráfico de barras con los idiomas utilizando matplotlib

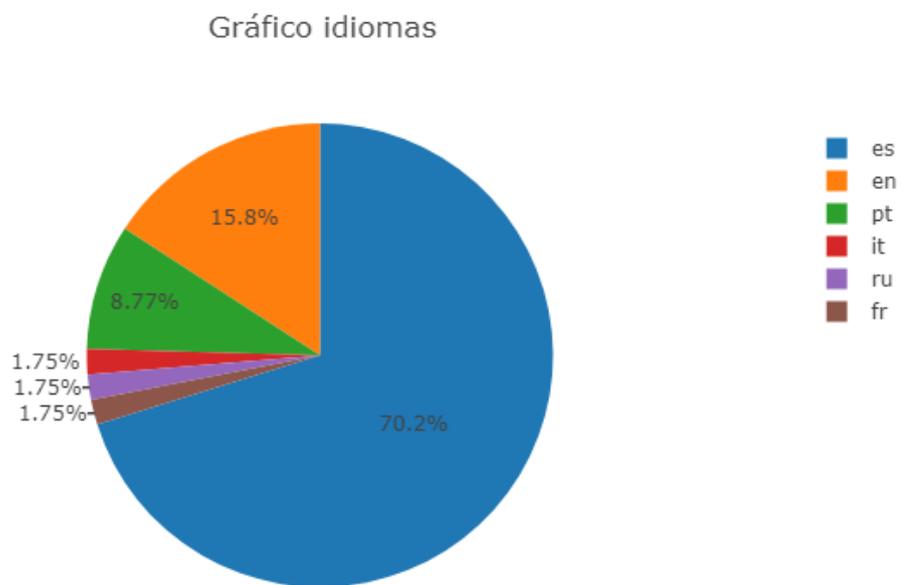


Ilustración 55: Gráfico circular con los idiomas utilizando plotly

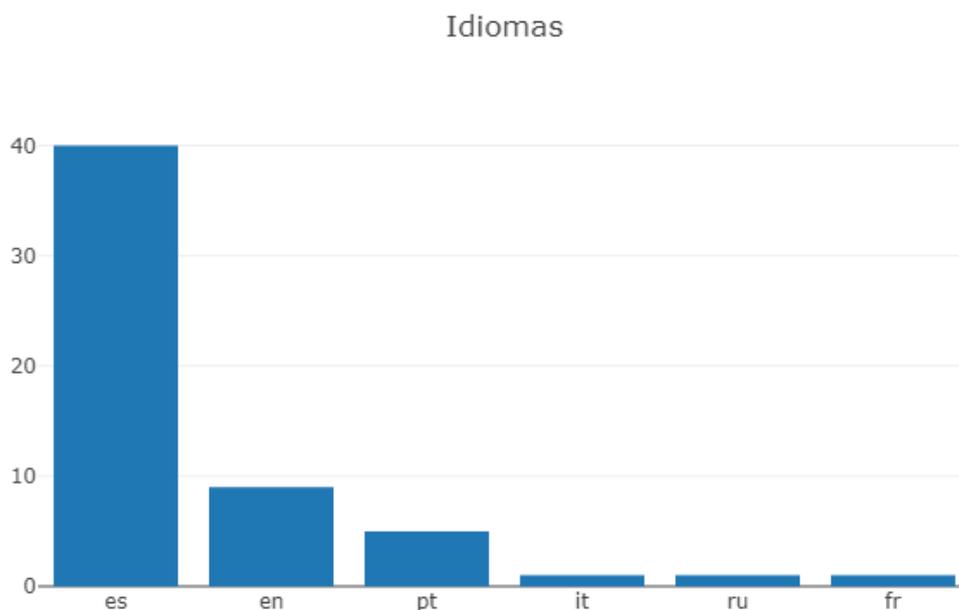


Ilustración 56: Gráfico de barras con los idiomas utilizando plotly

Como se puede ver en las imágenes el idioma predominante es el español abarcando el 70% seguido del inglés y del portugués, por tanto, sería interesante publicar contenido en la página en dichos idiomas dado que hay bastantes usuarios hablantes de estos idiomas. También cabe destacar la inexistencia de usuarios alemanes en la página de Facebook dado que una gran parte de los visitantes de la isla son alemanes.

5.1.7 Emociones

Respecto a las emociones las siguientes imágenes van a reflejar la puntuación de alegría, tristeza, ira, miedo y asco que se refleja en los comentarios de los usuarios de la página:

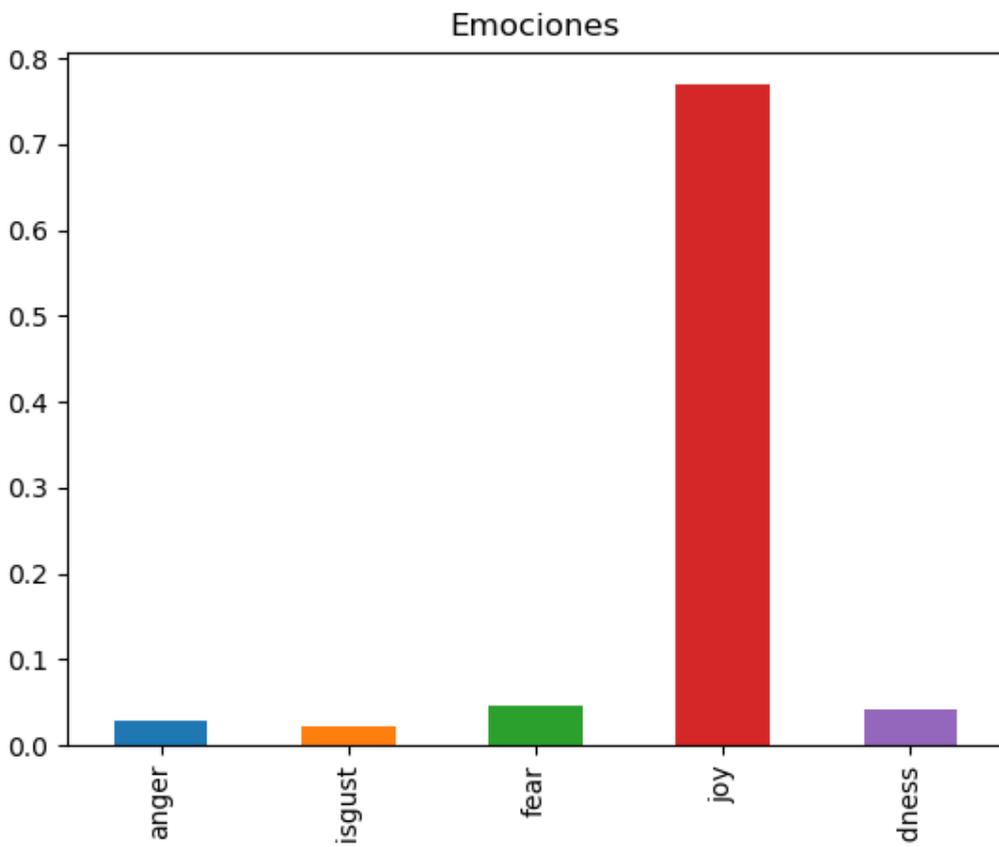


Ilustración 57: Gráfico de barras con las emociones utilizando matplotlib

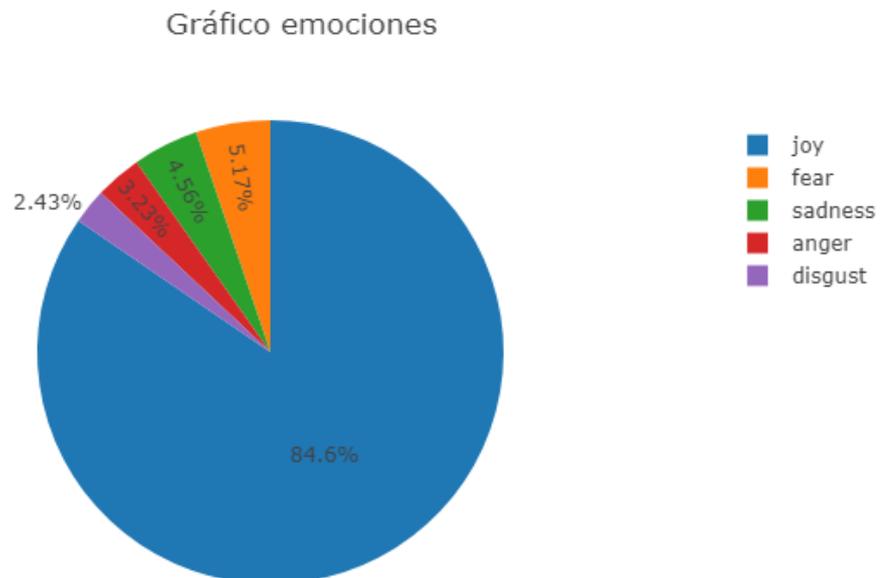


Ilustración 58: Gráfico circular con las emociones utilizando plotly

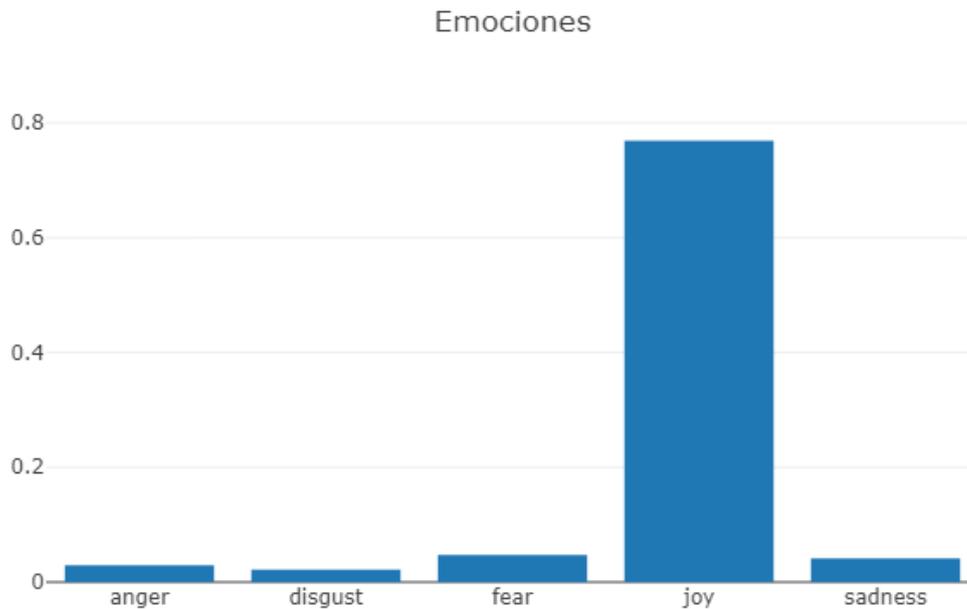


Ilustración 59: Gráfico de barras con las emociones utilizando plotly

En las imágenes se puede comprobar que la emoción que más se refleja en los comentarios es alegría por parte de los usuarios abarcando el 84% de la puntuación de las emociones, también se puede comprobar que la emoción que menos se refleja en los comentarios es asco con un 2.43% de la puntuación.

5.1.8 Sentimientos

Los últimos resultados que se reflejan en el proyecto son los relativos a los sentimientos de los comentarios, en las siguientes imágenes se muestran los gráficos respecto a la cantidad de comentarios positivos, neutros y negativos recibidos por los usuarios:

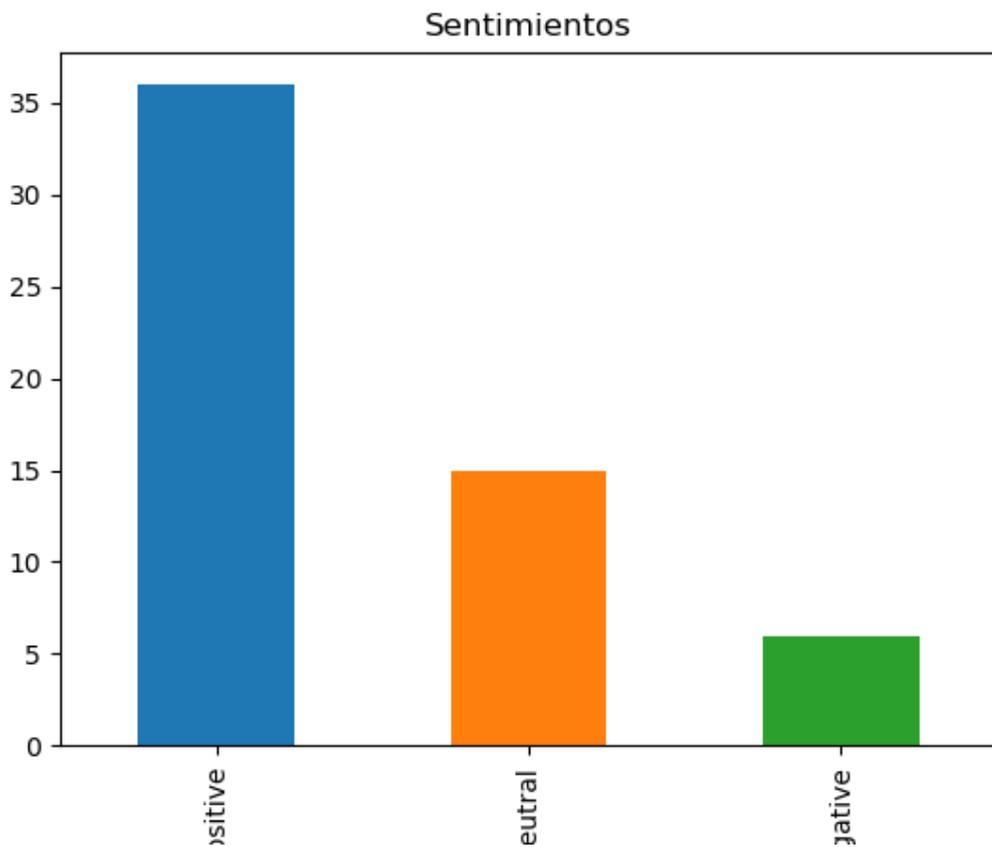


Ilustración 60: Gráfico de barras con los sentimientos utilizando matplotlib

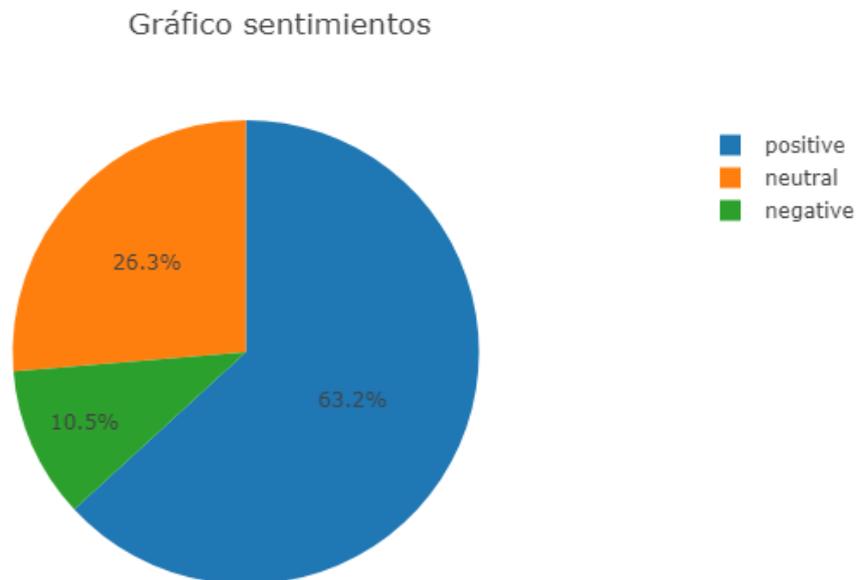


Ilustración 61: Gráfico circular con los sentimientos utilizando plotly

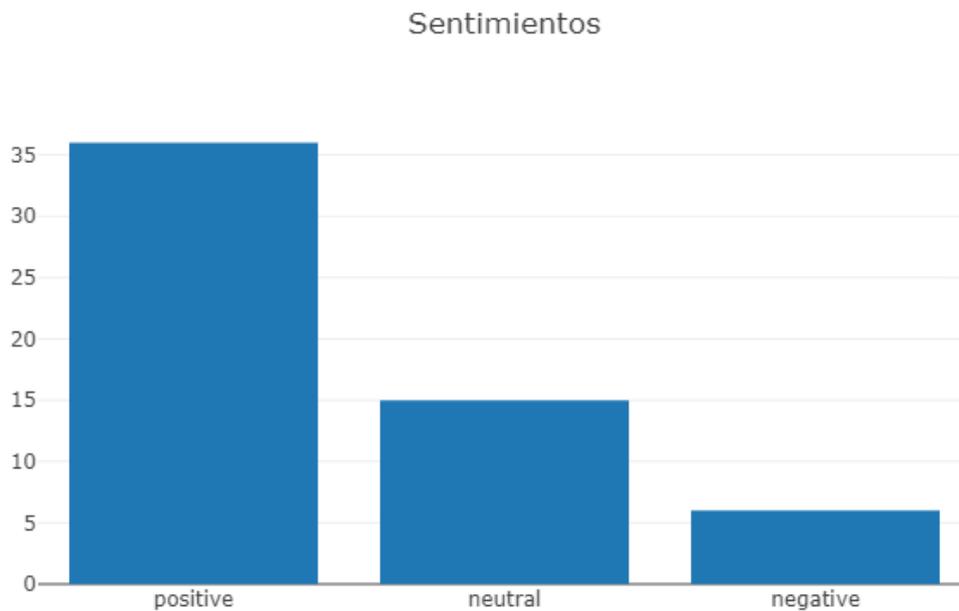


Ilustración 62: Gráfico de barras con los sentimientos utilizando plotly

En dichas imágenes podemos comprobar que el sentimiento predominante es el positivo con un 63% del total. También hay que destacar, que hay un porcentaje considerable de comentarios neutros siendo un 26% del total.

Respecto a la puntuación de los comentarios positivos y negativos, se puede comprobar en la siguiente imagen, que tanto los positivos y los negativos poseen una alta puntuación:

Puntuación sentimientos

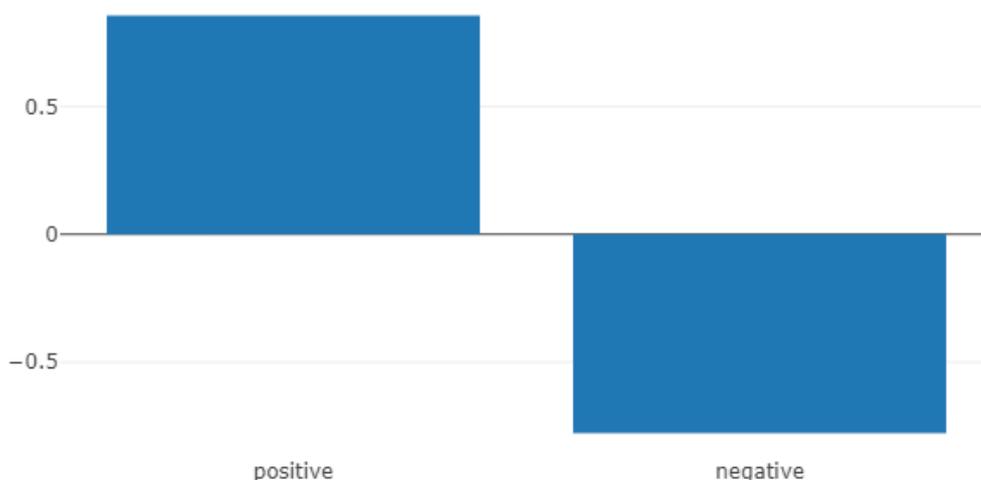


Ilustración 63: Gráfico de barras con la puntuación de cada uno de los sentimientos

5.1.8.1 Comentarios positivos

Para analizar los comentarios positivos y determinar cuáles son los temas que tratan y que se deben reforzar, ya que son los que más gustan a los usuarios de Facebook, se van a analizar las características, palabras clave, hashtags y frases nominales, de estos comentarios:



Ilustración 64: Nube de palabras con las palabras clave de los comentarios positivos

Con relación a las palabras clave, se puede destacar el Teide y lo relacionado con la espectacularidad de los paisajes de las islas, por tanto, se debe seguir destacando los paisajes naturales de nuestra isla, ya que es uno de los factores que más gusta a los turistas.



Ilustración 65: Nube de palabras con los hashtags de los comentarios positivos

Respecto a los hashtags se confirma que el más utilizado en los comentarios positivos es el mismo que más se utiliza en la página que es el de **Tenerife**, también se puede comprobar, que el Teide vuelve a triunfar entre los hashtags. También se puede ver que destacan hashtags relacionados con la naturaleza y el paraíso.



Ilustración 66: Nube de palabras con las frases nominales de los comentarios positivos

Por último, respecto a las frases nominales se puede ver que destacan los relacionados con los pueblos y con la idealidad de realizar determinadas cosas en estos lugares.

5.1.8.2 Comentarios negativos

Los comentarios negativos también deben de ser analizados con el fin de mejorar esos aspectos que los usuarios consideran negativos de la isla, con el fin de hacerlos más atractivos para estos y conseguir que estos temas negativos se vuelvan positivos:

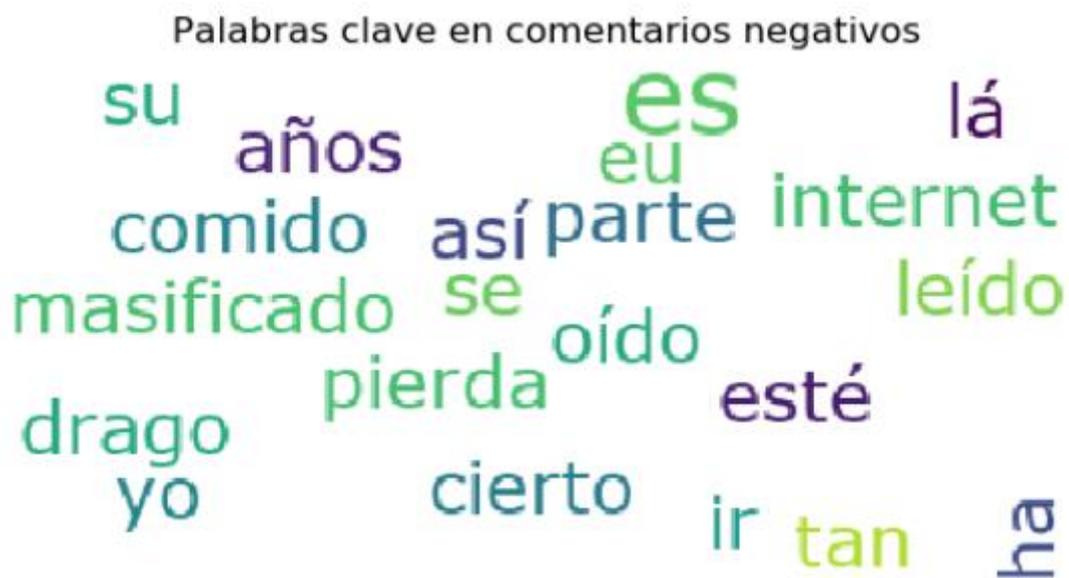


Ilustración 67: Nube de palabras con las palabras clave de los comentarios negativos

Respecto de las palabras clave cabe destacar que uno de los aspectos más negativos que se encuentran en los comentarios es la masificación de la isla y también van relacionados con el drago.

Hashtags en comentarios negativos



Ilustración 68: Nube de palabras con los hashtags de los comentarios negativos

En cuanto a los hashtags utilizados en estos comentarios tenemos solo tres hashtags. Nuevamente destaca el hashtag de **Tenerife**, como ha sido durante todo el análisis. Los otros dos son el **Drago**, como ya ocurría con las palabras clave, este destaca también en los hashtags, y el de **nostalgia**.

Frases nominales en comentarios negativos

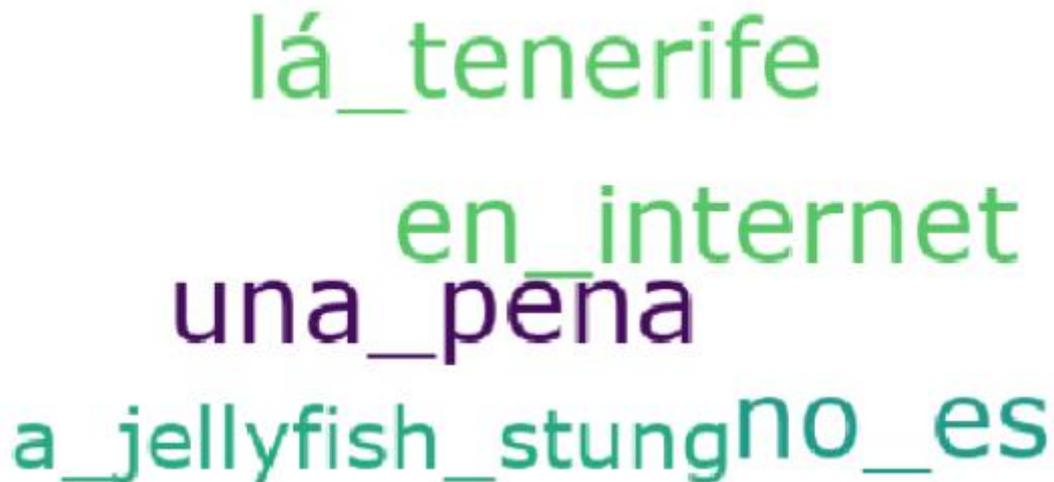


Ilustración 69: Nube de palabras con las frases nominales de los comentarios negativos

Por último, respecto de las frases nominales no hay nada reseñable, la única es la relacionado con medusas, donde se muestra el descontento de usuarios por este tema.

5.2 Conclusiones

Los resultados obtenidos de este análisis realizado sobre la página de Facebook **Turismo Tenerife** indican que tanto las publicaciones como los comentarios de la página se centran en el paisaje y en la hermosura de este, la gran mayoría de contenido referencia este aspecto, se echa en falta contenido respecto a la gastronomía, por ejemplo, la cual es uno de los aspectos más importantes de nuestra isla. También se echa en falta un mayor contenido respecto de las playas y piscinas de nuestra isla y de disfrutar del mar para darse un baño, si es cierto que hay contenido de este tema, pero está muy ensombrecido por el tema del paisaje.

Se puede comprobar también que por parte de los comentarios se trata el tema de la montaña y el senderismo, existiendo una demanda de contenido de este tema por parte de los usuarios, ya que en las publicaciones no encontramos casi contenido relacionado con esto.

Respecto a las publicaciones que más han recalado en los usuarios, ya que poseen el mayor número de me gustas y comparticiones, se pueden destacar los municipios de Güimar, Garachico e Icod entre otros, por tanto, la página podría considerar el explotar estos tres municipios ya que han recalado bastante bien por el público. Por otra parte, el contenido de estas publicaciones gira entorno al paisaje nuevamente, pero más en concreto al paisaje volcánico, pudiendo explotar este tema ya que ha recalado bastante bien entre los usuarios.

Entrando en el idioma en el que se han escrito los comentarios, se puede ver que la gran mayoría de usuarios son hispanohablantes, pero se puede ver que existe una variedad de idiomas en los comentarios, encontrándonos algunos en francés, italiano y ruso. También nos encontramos con que los idiomas extranjeros más utilizados en los comentarios son el inglés y el portugués, por tanto, sería interesante publicar contenido en estos idiomas, ya que hay bastantes usuarios a los que llegaría y se podría atraer a más gente que hable estos idiomas. Los sitios turísticos deberían tener en cuenta estos idiomas y tener información preparada para visitantes que hablen estos idiomas. Cabe destacar la inexistencia de contenido en alemán, siendo Alemania uno de los principales países desde los que llegan turistas a la isla, así que habría que tomar medidas para intentar hacer llegar el contenido de la página a los turistas de este país.

De las emociones emitidas por los usuarios en sus comentarios, se puede destacar que la mayor emoción reflejada es la alegría con una puntuación bastante alta respecto a las demás emociones, en ese sentido, las emociones generadas por la página en los usuarios son bastante buenas.

Por último, del análisis de sentimientos, podemos concluir con que el sentimiento generado por la página en sus usuarios mayoritariamente positivo, aunque hay un porcentaje de comentarios neutros considerable, que se deberían tratar de convertir en positivos, al igual que los comentarios que generan un sentimiento negativo deberían ser contrastados. De los comentarios positivos podemos destacar que se centran nuevamente en el paisaje y en la espectacularidad de este, así como en el Teide y la naturaleza, los cuales son aspectos que la página debe seguir potenciando ya que poseen una buena acogida por el público. De los comentarios negativos, se puede ver que giran en torno a la masificación de algunos puntos de la isla y al drago, por lo tanto, habría que tomar las medidas pertinentes para contrarrestar este sentimiento de los usuarios.

Una vez concluido este análisis, lo ideal sería compararlo con el análisis de otras páginas de Facebook de turismo de la isla y ver que aspectos del resto de páginas podrían mejorar esta y viceversa y qué temas son bien acogidos por los usuarios de las otras páginas, con el fin de mejorar cada una de estas páginas de turismo y conseguir atraer al mayor número de visitantes posibles. Dicha comparación no se ha podido realizar debido al cambio de política de privacidad de Facebook con la entrada en vigor del Reglamento General de Protección de Datos en la Unión Europea.

5.3 Aplicaciones

Este proyecto tiene diversas aplicaciones, ya no sólo en el ámbito del turismo, donde podemos comprobar el uso que se da a las páginas de turismo de cualquier lugar, en la plataforma de Facebook, podemos ver que lugares o atracciones turísticas son las más valoradas y demandadas por parte de los visitantes, también se puede comprobar la imagen que se da del lugar en esta red con el fin de atraer al mayor número de visitantes posibles, también se puede ver que aspectos deben ser mejorados, ya no sólo en la plataforma, sino en los servicios que se da a los visitantes y que aspectos se debe potenciar.

Otra de las aplicaciones, las podemos encontrar en las empresas, dónde podemos comprobar la popularidad de una determinada empresa por parte de los usuarios de la plataforma de Facebook, se puede comprobar el sentimiento general por parte de estos, respecto de la empresa y mejorar los aspectos que sean necesarios en función del estudio realizado mejorando la calidad de la empresa y de la imagen de esta a los usuarios de la plataforma.

También se puede comprobar la aceptación que va a tener un determinado producto o servicio en el mercado antes incluso de haberlo lanzado, comprobando el sentimiento y las reacciones que se producen en los usuarios. En función del estudio se podría mejorar algún aspecto del producto o servicio, o incluso decidir si

cancelar el lanzamiento de este debido a la enorme negativa por parte de los usuarios, consiguiendo una mejor toma de decisiones y reducción de costes.

6 Líneas futuras

Una vez terminado el desarrollo del proyecto, el objetivo es el de pasar la **revisión de la aplicación** requerida para acceder al resto de páginas de turismo de la isla de Tenerife y obtener un estudio mejor de estas, pudiendo mejorar el servicio dado por estas a los usuarios y atraer al mayor número de turistas. También se considera la realización de una web donde reflejar los resultados reflejados por el estudio, ya que es necesario para la aprobación de dicha revisión.

También se considera el realizar el estudio para el resto de las islas del archipiélago individualmente y para el conjunto de canarias, con el fin de mejorar el turismo en nuestras islas. También es ampliable a nivel nacional, analizando el turismo a este nivel o en cada una de las comunidades autónomas.

Otra idea es adecuar el estudio para las empresas, adecuando los aspectos obtenidos a estas con el fin de mejorar las pequeñas y medianas empresas de la isla, con el fin de mejorarlas y que puedan prestar un mejor servicio, así como mejorando su visibilidad de cara al público y la imagen que se tiene de estas.

7 Summary and conclusions

The results obtained from this analysis made on the **Turismo Tenerife** Facebook page indicate that both the publications and the comments of the page focus on the landscape and the beauty of this, the vast majority of content reference this aspect, There is a lack of content regarding gastronomy, for example, which is one of the most important aspects of our island. There is also a lack of more content on the beaches and pools of our island and enjoy the sea to take a bath, if it is true that there is content of this subject, but is very overshadowed by the theme of the landscape.

You can also check that on the part of the comments it is about the topic of the mountain and the hiking, there being a demand of content of this topic by the users, since in the publications we find almost content related to this.

Regarding the publications that have been most popular in users, because they have the greatest number of likes and shares, you can highlight the municipalities of Güimar, Garachico and Icod among others, therefore, the page might consider exploiting these three municipalities since they have pretty well made up for the public. On the other hand, the content of these publications revolves around the landscape again, but more specifically to the volcanic landscape, being able to exploit this topic because it has made quite success among users.

Entering in the language in which the comments have been written, you can see that the vast majority of users are Spanish speakers, but you can see the variety of languages exist in the comments, finding some in French, Italian and Russian. We also find that the foreign languages most used in the comments are English and Portuguese, so it would be interesting to publish content in these languages, as there are many users who would arrive and could attract more people who speak these languages. Tourist sites should have in mind these languages and have information prepared for visitors who speak these languages. It is worth noting the non existence of German content, being Germany one of the main countries from which tourists arrive to the island, so we would have to take measures to try to get the content of the page to tourists from this country.

Of the emotions emitted by users in their comments, it can be noted that the greatest emotion reflected is the joy with a fairly high score compared to other emotions, in that sense, the emotions generated by the page in users are quite good.

In relation to the future lines of the project, we study the possibility of expanding the study to the rest of the islands of the archipelago, to improve the tourism of the Canary Islands, as well as to apply it at national level. On the other hand, we also study the possibility of adapting the project for companies improving the image of these in the face of the public

8 Presupuesto

Tareas desarrolladas			
Tarea	Horas dedicadas	Costo unitario	Costo total
Extracción y almacenamiento de datos	30 h.	30 €	900 €
Extracción de características	60 h.	30 €	1800 €
Análisis de contenidos	30 h.	30 €	900 €
Análisis de sentimientos	40 h.	30 €	120 €
		Cuantía total	4800 €

Tabla 2: Presupuesto de las tareas desarrolladas

Software			
Nombre	Cantidad	Costo unitario	Costo total
Visual Studio Code	1	0 €	0 €
Jupyter Notebook	1	0 €	0 €
MongoDB	1	0 €	0 €
Github	1	0 €	0 €
Bitbucket	1	0 €	0 €
Gitlab	1	0 €	0 €
API Graph Facebook	1	0 €	0 €
Watson Natural Language Understanding	1	0 €	0 €
		Cuantía total	0 €
Hardware			
Nombre	Cantidad	Costo unitario	Costo total
Ordenador portátil (12 GB de RAM, 320 GB HDD, win10 64 bits, CPU i7-3610QM, 2.30GHz)	1	600 €	600 €
		Cuantía total	600 €

Tabla 3: Presupuesto de software y hardware

9 Referencias

[1] Python:

<https://www.python.org/doc/essays/>

[2] Biblioteca requests:

<http://docs.python-requests.org/en/master/>

[3] Biblioteca json:

<https://docs.python.org/3.4/library/json.html>

[4] Biblioteca pymongo:

<https://api.mongodb.com/python/current/>

[5] Biblioteca pandas:

<https://pandas.pydata.org/>

[6] Biblioteca re:

<https://docs.python.org/3/library/re.html>

[7] Biblioteca nltk:

<https://www.nltk.org/>

[8] Biblioteca numpy:

<http://www.numpy.org/>

[9] Biblioteca matplotlib:

<https://matplotlib.org/>

[10] Biblioteca itertools:

<https://docs.python.org/3/library/itertools.html>

[11] Biblioteca datetime:

<https://docs.python.org/3/library/datetime.html>

[12] Biblioteca ggplot:

<http://ggplot.yhathq.com/>

[13] Biblioteca wordcloud:

https://amueller.github.io/word_cloud/index.html

[14] Biblioteca plotly:

<https://plot.ly/python/>

[15] Pip:

<https://pypi.org/project/pip/>

[16] Visual Studio Code:

<https://code.visualstudio.com/>

[17] Jupyter Notebook:

<http://jupyter.org/>

[18] MongoDB:

<https://www.mongodb.com/>

[19] Git:

<https://git-scm.com/>

[20] Apache Subversion:

<https://subversion.apache.org/>

[21] Github:

<https://github.com/>

[22] Bitbucket:

<https://bitbucket.org/product>

[23] Gitlab:

<https://about.gitlab.com/>

[24] Graph API Facebook:

<https://developers.facebook.com/docs/graph-api/>

[25] Watson Natural Language Understanding:

<https://www.ibm.com/watson/services/natural-language-understanding/>

[26] Reglamento General de Protección de Datos

<https://www.boe.es/doue/2016/119/L00001-00088.pdf>

10 Bibliografía

- Siddhartha Chatterjee; Michael Krystyanczuk, *Python Social Media Analytics*, Birmingham: Packt Publishing, 2017