



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Sentinel**



Presentado por Zoe Calvo Serna  
en Universidad de Burgos — 29 de junio  
de 2020

Tutores: José Manuel Galán Ordax y Virginia  
Ahedo García







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. José Manuel Galán Ordax y D<sup>a</sup> Virginia Ahedo García, profesores del departamento de Ingeniería Civil, área de Ingeniería de Organización.

Expone:

Que el alumno D. Zoe Calvo Serna, con DNI 71309260Z, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Sentinel.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 29 de junio de 2020

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. José Manuel Galán Ordax

D<sup>a</sup>. Virginia Ahedo García





## **Resumen**

Este proyecto utilizará las APIs de las redes sociales principales (instagram, twitter,...) para crear una aplicación que permita estudiar mediante sentiment analysis las series temporales de percepción de diferentes marcas comerciales, procesos sociológicos, el efecto de campañas de marketing, etc.

## **Descriptores**

Sentiment analysis, redes sociales, twitter, instagram, servidor flask, aplicación web, angular, rest.

### **Abstract**

This project will use the APIs of the main social networks (instagram, twitter,...) to create an application that allows you to study through sentiment analysis the time series of different trademarks, sociological processes, the effect of marketing campaigns...

### **Keywords**

Sentiment analysis, social networks, twitter, instagram, flask server, web application, angular, rest.



---

# Índice general

---

<b>Índice general</b>	<b>III</b>
<b>Índice de figuras</b>	<b>V</b>
<b>Índice de tablas</b>	<b>VI</b>
<b>Introducción</b>	<b>1</b>
1.1. Estructura de la memoria . . . . .	2
1.2. Estructura de los anexos . . . . .	3
<b>Objetivos del proyecto</b>	<b>5</b>
2.1. Objetivos generales . . . . .	5
2.2. Objetivos técnicos . . . . .	6
<b>Conceptos teóricos</b>	<b>7</b>
3.1. Sentiment Analysis . . . . .	7
3.2. API . . . . .	8
3.3. Web Service . . . . .	8
3.4. Protocolo HTTP . . . . .	9
3.5. REST . . . . .	9
3.6. URI . . . . .	10
3.7. API Endpoint . . . . .	10
3.8. Idempotencia . . . . .	11
3.9. Series temporales . . . . .	11
<b>Técnicas y herramientas</b>	<b>13</b>
4.1. Python . . . . .	13

4.2. PyCharm . . . . .	14
4.3. Flask . . . . .	14
4.4. MySQL . . . . .	14
4.5. GitHub . . . . .	15
4.6. ZenHub . . . . .	16
4.7. GitKraken . . . . .	16
4.8. Zotero . . . . .	16
4.9. LaTeX . . . . .	17
4.10. Overleaf . . . . .	17
4.11. Pencil Project . . . . .	17
4.12. Senti-py . . . . .	17
4.13. TextBlob . . . . .	18
4.14. Yandex Translate . . . . .	18
4.15. Heroku . . . . .	18
4.16. Drawio . . . . .	18
4.17. Statsmodels . . . . .	18
<b>Aspectos relevantes del desarrollo del proyecto</b>	<b>19</b>
<b>Trabajos relacionados</b>	<b>31</b>
6.1. MonkeyLearn . . . . .	31
6.2. Lexalytics . . . . .	33
6.3. Brandwatch . . . . .	35
<b>Conclusiones y Líneas de trabajo futuras</b>	<b>37</b>
7.1. Conclusiones . . . . .	37
7.2. Líneas de trabajo futuras . . . . .	38
<b>Bibliografía</b>	<b>39</b>

---

# Índice de figuras

---

5.1. Conexión desde Python con MySQL. . . . .	22
5.2. Conexión desde Python con MySQL. . . . .	22
5.3. Página de inicio de la aplicación. . . . .	22
5.4. Cálculos estadísticos. . . . .	23
5.5. Gráficos implementados. . . . .	23
5.6. Servicio de twitter. . . . .	24
5.7. Acceso a la información enviada del servidor a la interfaz. . . . .	24
5.8. Filtrar por fecha. . . . .	25
5.9. Funcionalidad para buscar un id que no está en la base de datos. . . . .	25
5.10. Funcionalidad para actualizar base de datos. . . . .	26
5.11. Menú de elección para el cálculo de series temporales. . . . .	26
5.12. Gráfica de series temporales. . . . .	27
5.13. Archivo json con las traducciones al inglés. . . . .	27
6.14. Monkeylearn connect your text data. . . . .	32
6.15. Monkeylearn turn your text into tags. . . . .	32
6.16. Monkeylearn put your tags to work. . . . .	32
6.17. Lexalytics análisis de texto . . . . .	33
6.18. Lexalytics entidades. . . . .	34
6.19. Lexalytics entidades subrayadas en el texto. . . . .	34
6.20. Brandwatch gráfico de líneas. . . . .	36
6.21. Brandwatch gráfico de burbujas. . . . .	36

---

# Índice de tablas

---

6.1. Precios de Brandwatch. . . . .	35
-------------------------------------	----

---

# Introducción

---

Un pilar básico en las organizaciones públicas y privadas que se dirigen a la sociedad, ya sea ofreciendo un producto o un servicio, es la opinión de las personas.

En el pasado la forma de conocer estas opiniones era mediante encuestas que después se trataban para hacer estadísticas y evaluar la calidad del producto.

Estas encuestas tiene dos grandes limitaciones:

- La muestra es limitada y a partir de esta debemos generalizar.
- La opinión de esta población viene dada por unas preguntas que realiza la organización previamente, por tanto son demasiado específicas y puede que haya aspectos que no se tengan en cuenta.

Hoy en día las redes sociales constituyen un gran avance en este ámbito. La gente puede mostrar su opinión sobre cualquier servicio o producto sin necesidad de encuestas y además pueden tener un alcance global. Esto representa un gran cambio tanto para las personas como para las organizaciones. La gente puede basarse en la opinión de personas muy dispares para tomar decisiones sobre un producto. Las organizaciones pueden aprovecharse de la diversidad de la población y de la sinceridad de la opinión ya que no está acotada por preguntas exactas.

En este punto es donde el análisis de sentimiento toma protagonismo. Este concepto no es más que el estudio del lenguaje natural, análisis de texto y lingüística computacional, para tratar de establecer métricas capaces de capturar el sentimiento general de un texto.

Es un campo muy grande que abarca diferentes técnicas. Como ocurre con otros términos, es más conocido el nombre en inglés, Sentiment Analysis, por lo que a partir de ahora nos referiremos al tema con el término anglosajón.

Estas métricas han hecho que las organizaciones complementen sus encuestas con los resultados obtenidos en las redes sociales y así poder tener un mayor feedback sobre sus servicios.

En este trabajo se propone la utilización de Sentiment Analysis para realizar estudios en Twitter e Instagram sobre temas de actualidad, productos o servicios. Para ello se recopilará información de estas redes sociales y se mostrará en gráficos y tablas para hacer más fácil la interpretación de los datos.

## 1.1. Estructura de la memoria

- **Introducción:** Descripción de la situación y el tema sobre el que el proyecto va a versar. Estructura de la memoria y de los anexos.
- **Objetivos del proyecto:** Explicación de los temas a tratar en el trabajo.
- **Conceptos teóricos:** Exposición de conceptos que facilitan la comprensión del proyecto.
- **Técnicas y herramientas:** Listado de metodologías y herramientas que han sido utilizadas para llevar a cabo el proyecto.
- **Aspectos relevantes:** Muestra aspectos a destacar durante la realización del proyecto
- **Trabajos relacionados:** Estado del arte en el ámbito del sentiment analysis y trabajos similares.
- **Conclusiones y líneas de trabajo futuras:** Conclusiones obtenidas al final del proyecto y posibles ideas futuras.

## 1.2. Estructura de los anexos

- **Plan de proyecto software:** Planificación temporal y viabilidad económica y legal.
- **Especificación de requisitos:** Objetivos y requisitos establecidos al comienzo del proyecto.
- **Especificación de diseño:** Recoge los diseños de datos, procedimental, arquitectónico y de interfaces.
- **Manual del programador:** Explica los conceptos más técnicos del proyecto como su instalación, la organización de carpetas y la ejecución.
- **Manual de usuario:** Es la guía de cómo utilizar la aplicación paso a paso.





---

# Objetivos del proyecto

---

En este apartado vamos a detallar los objetivos que queremos abordar en el proyecto, que serán denominados generales, y los objetivos necesarios para llevar a cabo los anteriores, que denominaremos como técnicos.

## 2.1. Objetivos generales

- Extraer datos de las redes sociales Twitter e Instagram.
- Estudiar mediante Sentiment Analysis la información recogida de estas APIs.
- Realizar estadísticas sobre los datos recopilados.
- Crear una aplicación web para mostrar la información.
- Mostrar las estadísticas mediante gráficos y tablas.
- Analizar mediante series temporales tendencias futuras en la evolución de las métricas.
- Crear una wiki para el manual de usuario.

## 2.2. Objetivos técnicos

- Extraer los datos de las APIs de Twitter e Instagram.
- Utilizar librerías de Sentiment Analysis para analizar los datos obtenidos.
- Realizar operaciones estadísticas sobre los resultados de Sentiment Analysis.
- Almacenar en la base de datos los datos de las APIs junto a sus resultados y cálculos de las operaciones estadísticas.
- Desarrollar una interfaz que sea adaptativa y fácil de utilizar para el usuario.
- La aplicación deberá mostrar los resultados en gráficos y tablas.
- Utilizar Git como sistema de control de versiones junto con la plataforma GitHub.
- Usar la plataforma gráfica GitKraken para Git.
- Hacer uso de ZenHub para la gestión del proyecto.
- Utilizar un sistema para las referencias bibliográficas como Zotero.
- Utilizar la librería statsmodels de Python para calcular series temporales y realizar test para ver si es estacionaria.

---

# Conceptos teóricos

---

## 3.1. Sentiment Analysis

El análisis de sentimiento [22] es una disciplina informática que se basa en identificar de forma automática las opiniones, sentimientos, pensamientos y emociones de las personas a través de textos, vídeos o audios. Esta rama de estudio pertenece al ámbito de la inteligencia artificial, ciencia social y ciencias de la computación. El análisis de sentimiento puede aplicarse a diferentes niveles:

- Detección de subjetividad: Solamente distingue entre si un comentario es subjetivo u objetivo.
- Detección de polaridad: Tras detectar que el comentario es subjetivo, determina si es positivo o negativo.

El interés de esta disciplina es que puede usarse tanto a nivel de aplicaciones destinadas a empresas y en redes sociales, pero también puede aplicarse a la educación y al cuidado de la salud.

Existen tres métodos aplicables para extraer la polaridad de un texto:

- Método basado en la lingüística: Se asocia un valor real a cada concepto que se extrae de la frase. Luego se corrigen y se combinan todos en un solo valor. En el proceso de combinación el valor irá cambiando dependiendo de palabras como 'pero' o 'muy'.
- Método basado en machine learning: Se utiliza una máquina la cual extrae los conceptos de la frase en un vector y les aplica un algoritmo de forma automática para generar el valor de la polaridad.

- Método combinado: Se basa en combinar las dos técnicas anteriores. Puede ocurrir que la primera técnica no consiga extraer la polaridad, pero sus resultados son más fiables. Por ello, en la combinación la técnica de machine learning respalda a la de lingüística para ofrecer los mejores resultados.

## 3.2. API

Una API [24] es un conjunto de código utilizado para desarrollar el software de aplicaciones. Las siglas se refieren a Application Programming Interface o Interfaz de Programación de Aplicaciones.

Son clave para la simplificación del desarrollo de aplicaciones, ya que no es necesario conocer la implementación de sus productos para poder comunicarse con ellos.

Otras ventajas importantes son:

- Dan flexibilidad.
- Simplifican el diseño y el uso de la aplicación.
- Ahorran tiempo y dinero.

## 3.3. Web Service

Un web service [9] es un conjunto de protocolos y estándares que es utilizado por aplicaciones que están escritas en distintos lenguajes de programación para intercambiar información entre ellas. Estas aplicaciones envían parámetros al servidor donde se aloja el web service y este les responde la petición.

Algunas de las mejoras que aportan a las aplicaciones son las siguientes:

- Proporcionan interoperabilidad entre aplicaciones software, sin importar la plataforma sobre la que estén instaladas.
- Permite que los servicios de una compañía se comuniquen con los de otra que está en diferente lugar geográfico.
- Se aprovecha de los sistemas de seguridad firewall del protocolo HTTP.
- Fomentan la utilización de estándares y protocolos basados en texto, los cuales facilitan su entendimiento y el acceso a su contenido.

La ventaja más importante de los web services es que son muy prácticos ya que son independientes de las aplicaciones.

## 3.4. Protocolo HTTP

El protocolo de transferencia de hipertexto [32] es el que permite las transferencias en la World Wide Web. Se trata de un protocolo de comunicación que define la sintaxis que deben utilizar los elementos software de la arquitectura web para comunicarse. HTTP no guarda la información sobre conexiones anteriores, es decir, es un protocolo sin estado. Esto es un problema ya que las aplicaciones web necesitan mantener este estado. Para solucionarlo están las cookies, que es información que el servidor almacena y es lo que permite, por ejemplo, recordar el inicio de sesión en las aplicaciones.

## 3.5. REST

El protocolo REST [7] o Representational State Transfer, es un conjunto de características que podemos usar para definir una arquitectura software que será utilizada para crear aplicaciones web que respeten el protocolo HTTP. Es una alternativa a otros protocolos como SOAP que es muy complejo.

Las características de REST son:

- Las operaciones más importantes son GET, POST, PUT y DELETE.
- El sistema debe estar dividido por capas, cada una de estas tendrá una funcionalidad de REST.
- Cada petición realizada a HTTP debe llevar toda la información necesaria ya que es un protocolo sin estado. Stateless o sin estado significa que no es capaz de guardar datos y por tanto no puede recordarlos.
- Se utilizan hipermedios, lo que permite al cliente navegar por los objetos HTML y ejecutar acciones definidas sobre los datos.
- Los objetos REST se identifican a partir de su URI, lo cual nos facilita acceder a la información para modificarla.
- El sistema REST ofrece una interfaz uniforme ya que las acciones que se realizan sobre los objetos son concretas y estos siempre se identifican por la URI.

Tras explicar las características del protocolo REST, hay que destacar las ventajas que ofrece frente a otros:

- Una API REST es independiente del lenguaje de programación o la plataforma sobre la que se trabaje, lo único que debe permanecer es

el lenguaje de las peticiones, ya sea XML o JSON. Esto ofrece gran libertad ya que permite usar distintos entornos de desarrollo.

- Separa el cliente y el servidor, es decir, la interfaz de usuario del servidor y el almacenamiento de datos son independientes, lo que mejora la portabilidad de la interfaz y aumenta la escalabilidad. Además, esta separación permite tener el front y el back en distintos servidores y esto hace que la aplicación sea más flexible.

### 3.6. URI

El Uniform Resource Identifier [11] sirve para acceder a un recurso por internet. Es el identificador que identifica la información que a la que hay que acceder y donde se encuentra.

Se compone de cinco partes, dos obligatorias, scheme y path que proporcionan la información del protocolo utilizado y la ruta al recurso, y tres opcionales, authority, query y fragment, que identifica el dominio, representa la consulta y designa una parte del recurso respectivamente.

Hay dos tipos de URI, el absoluto y el relativo. El absoluto el cual es independiente del contexto y tiene que llevar obligatoriamente el scheme, authority y path. El relativo no tiene que llevar el scheme por lo que es necesario contar con un URI base que permita que se anexe correctamente.

### 3.7. API Endpoint

Un API Endpoint [26] es el lugar donde la api y el software de la aplicación se conectan. Es decir, es el punto de conexión para enviar peticiones y recibir respuesta.

La parte del programa que envía la información por el endpoint es el servidor y el cliente gestiona las peticiones. Los endpoints permiten operaciones como GET, POST, DELETE y PATCH, y normalmente se especifica la operación a realizar en la URL.

## 3.8. Idempotencia

La idempotencia [25] es una característica que tienen algunos métodos HTTP por la cual el resultado no se altera, no importa las veces que el método sea llamado.

En las APIs REST los métodos que tienen esta característica son:

- GET: Este método nunca debe cambiar su resultado, ya que su función es recuperar datos para su representación. Por tanto, siempre es idempotente.
- PUT: Su función es actualizar el estado un recurso, por tanto habrá ocasiones en las que se ejecute y el estado sea el mismo una y otra vez. Por ello, generalmente es un método idempotente.
- DELETE: La primera vez que se ejecuta este método da una respuesta, puede ser que borre de forma efectiva o que no encuentre el objeto, pero el resto de respuestas serán la misma, por ello es idempotente.

El método que en ningún caso puede ser idempotente es el método POST, ya que su función es crear un nuevo recurso, por tanto la respuesta no será nunca la misma.

## 3.9. Series temporales

Una serie temporal [27] es una secuencia de observaciones de una variable a lo largo del tiempo, tomando estos valores ordenados cronológicamente.

Idealmente las muestras son tomadas en intervalos regulares de tiempo.

Los componentes de una serie temporal son cuatro:

- Tendencia: Movimiento regular de la serie a largo plazo.
- Estacionalidad: Fluctuaciones a corto plazo del periodo regular.
- Residuo: Oscilaciones imprevisibles debidas a factores externos que no muestran periodicidad.
- Variaciones cíclicas: Movimientos a medio plazo que presentan cierta regularidad.

Los esquemas más utilizados son el aditivo y el multiplicativo, esto no quiere decir que todas las series temporales sean compatibles con ellos. Para saber cual es el más indicado, podemos observar si la serie es estacionaria

o no. Una serie estacionaria es aquella que tiene una tendencia constante a lo largo del tiempo y que no presenta movimiento estacional, aunque sí puede presentar movimiento residual. Por ello, en las series no estacionarias tiene más sentido utilizar el esquema multiplicativo, ya que la relación entre dos registros es más lógico que sea más parecida en terminos relativos que absolutos.

Para elegir el esquema adecuado, hay varios métodos, pero el más frecuente es representar la serie para estudiarla posteriormente.

Los tipos de series temporales que podemos ver en este proyecto son: [20]

- **Suavizado exponencial:** [2] Es la evolución de la media móvil ponderada, en este caso se utiliza un método de autocorrección para ajustar los pronósticos en dirección opuesta a las desviaciones anteriores. Dicha corrección se ve afectada por un coeficiente de suavización.
- **Método de Holt:** Pretende identificar las tendencias lineales mediante el doble alisado, sin contemplar otras componentes de la serie. Busca identificar la tendencia de la serie de tal forma que permita a la tendencia variar a lo largo del tiempo, ajustándose el modelo de forma automática.
- **Modelos Arima:** Son modelos que consiguen una representación más simple combinando términos autoregresivos y de medias móviles. Para poder modelar una serie con Arima debe convertirse primero a estacionaria.



---

# Técnicas y herramientas

---

## 4.1. Python

Python [29] es un lenguaje de programación multiplataforma orientado a objetos y además interpretado, por lo que no es necesaria su compilación para ejecutarlo, lo que aumenta la rapidez de desarrollo.

Destaca respecto a otros lenguajes por varias razones:

- Tiene una gran cantidad de librerías que incorporan varios tipos de datos y funciones, lo que facilita la realización de varias tareas.
- Es sencillo y los programas son creados con gran rapidez.
- Como ya se ha comentado puede ser utilizado en varias plataformas.
- Es gratuito.

Python fue creado por Guido Van Rossum. Buscaba un lenguaje que fuera orientado a objetos, sencillo y que realizase una serie de tareas que en esos momentos se hacían utilizando el lenguaje C.

Este lenguaje es utilizado por grandes compañías como Google, Youtube o Facebook.

## 4.2. PyCharm

Para realizar el desarrollo de la aplicación se ha utilizado PyCharm [12]. Es un entorno de desarrollo para trabajar con Python que facilita el desarrollo de aplicaciones web tanto para la parte backend, Django, Flask,... como para la parte de frontend HTML, CSS, Angular...

Es un entorno muy intuitivo y ofrece un intérprete en el editor de código en tiempo real lo que facilita reconocer los errores.

Además, cuenta con una versión gratuita para estudiantes que ofrece todos las opciones de desarrollo.

## 4.3. Flask

Flask [19] es un micro framework escrito en Python que permite crear aplicaciones web de forma rápida.

Al ser micro, flask se instala con las herramientas necesarias para hacer una web funcional, pero en caso de necesitar funcionalidades extras se pueden añadir ya que cuenta con un gran conjunto de plugins.

Además incluye un servidor web de desarrollo por lo que nos facilita la visualización de los cambios que vamos realizando.

También se estudió la opción de realizar la aplicación con Django [18], ya que tiene la misma funcionalidad, pero al final nos decantamos por Flask ya que en Django el uso de sockets es confuso, y además es un framework que está más orientado a proyectos más grandes y robustos.

## 4.4. MySQL

MySQL [31] es un gestor de bases de datos relacional desarrollado por Oracle Corporation y es considerada una de las más populares en todo el mundo.

Inicialmente fue creada por MySQLAB, la cual fue adquirida por Sun Microsystems y esta comprada por Oracle Corporation en 2010.

Para acceder a las bases de datos de MySQL se pueden usar diferentes lenguajes de programación como Java, C, Python, PHP,...

Se utiliza mayormente para aplicaciones web ya que es muy rápida en lectura y la modificación de los datos suele ser baja.

Hay varias ventajas que se deben destacar [4]:

- **Seguridad de los datos:** Es el sistema de administración de bases de datos más seguro, utilizado por ello por aplicaciones como Wordpress o Twitter.
- **Alto rendimiento:** Cuenta con un marco de motor de almacenamiento que permite a sus clientes configurar el servidor para tener un buen rendimiento.
- **Escalabilidad:** Facilita la administración de aplicaciones que estén muy integradas. Permitiendo adaptarse a lo largo del tiempo.

También se barajó utilizar SQLite, pero nos decantamos por MySQL debido a las ventajas anteriores, ya que nuestra aplicación va a utilizar varios datos de cuentas de redes sociales y queremos que el entorno sea seguro.

## 4.5. GitHub

GitHub [8] es una plataforma de desarrollo colaborativo software para albergar proyectos que utilicen Git como sistema de control de versiones.

No solo aloja tu repositorio de código, también te ofrece herramientas para facilitar el trabajo en equipo, algunas destacadas son:

- **Wiki:** Para el mantenimiento de las distintas versiones.
- **Sistema de seguimiento de problemas:** Permite a los miembros del equipo detallar los problemas que pueda haber en el software.
- **Herramienta de revisión de código:** En la que se pueden añadir notas para debatir sobre los cambios realizados.
- **Visor de ramas:** Muestra las ramas que hay creadas en un repositorio y los desarrollos de cada una.

En este proyecto GitHub es una de las partes más importantes ya que es donde se encuentra todo el código referente al proyecto y donde ha quedado registrado el trabajo que se ha ido haciendo de forma progresiva.

## 4.6. ZenHub

ZenHub [17] es una plataforma de gestión de proyectos que se integra con GitHub.

Nos permite controlar los proyectos utilizando paneles muy visuales que hacen más sencilla la gestión de tareas.

Además cuenta con diferentes gráficos e informes estadísticos que pueden sacarse por semana de trabajo para evaluar si la estimación ha sido la correcta.

## 4.7. GitKraken

Gitkraken [23] es una interfaz gráfica multiplataforma para Git. Permite llevar un seguimiento de los repositorios junto con sus ramas, tags y manejarlo todo de forma muy visual.

Ha sido imprescindible durante el desarrollo del proyecto para ver las diferentes ramas y los cambios realizados. Todos los commits se hacían en local desde la aplicación y al final de cada sprint se realizaba un push a remoto, para que quedara constancia de los cambios en GitHub.

## 4.8. Zotero

Zotero [33] es un gestor de referencias libre, gratuito y multiplataforma.

Tiene varias funciones:

- *Recopilar*: Recopila información y la guarda en la base de datos mediante capturas, ya sean individuales o colectivas.
- *Organizar*: Para poder encontrarlos de forma fácil se basa en colecciones, etiquetas, elementos relacionados y búsquedas guardadas.
- *Citar*: Permite crear referencias bibliográficas de forma muy sencilla.
- *Sincronizar*: Podemos guardar todas las referencias en nuestro ordenador y servidor.
- *Colaborar*: Permite compartir colecciones y crear grupos de colaboración.

En este proyecto se han utilizado las funciones de recopilación, organización, citar y sincronizar. Todas las referencias bibliográficas de la memoria y de los anexos han sido generadas con Zotero lo que ha facilitado mucho esta

labor, ya que permite guardar todas las webs necesarias y tiene la opción de generar citas en lenguaje LaTeX.

## 4.9. LaTeX

LaTeX [30] es un sistema de composición tipográfica basado en TeX. Fue creado con la intención de que el autor no se tuviera que preocupar por la forma del documento y se centrara más en el contenido.

Garantiza una buena estructura , casi de forma automática. Nació para ser utilizado mayormente en documentos científicos por tener que incluir fórmulas matemáticas, pero puede utilizarse en todos los ámbitos. Sobre todo, es ideal para trabajos de gran extensión ya que permite tener el documento perfectamente organizado.

## 4.10. Overleaf

Overleaf [21] es un editor de textos LaTeX online que permite además de realizar documentos en lenguaje TeX tener varios colaboradores dependiendo de la versión que tengamos. Se ha utilizado este editor por esta razón, ya que permite que el tutor pueda hacer anotaciones en el texto directamente y no sea necesario exportar el documento a un PDF, gracias a que es online.

## 4.11. Pencil Project

Pencil Project [5] es una aplicación para la creación de prototipos de interfaces gráficas. Estos prototipos pueden estar basados en aplicaciones para móviles o escritorio. Ha sido utilizado para crear el prototipo de interfaz gráfica de la aplicación web para establecer el camino a seguir en el proyecto.

## 4.12. Senti-py

Senti-py [6] es una librería de Python para realizar el análisis de sentimientos en textos en español. Para realizar el análisis debemos pasar un json y nos retornará un valor entre 0 y 1 donde 0 significa que el comentario es muy negativo y 1 que es muy positivo.

### 4.13. TextBlob

TextBlob [28] es una librería muy completa que tiene varias funcionalidades, traducción, podemos añadir palabras, nos permite clasificar el sentimiento, etc. Es la que nos va a permitir analizar los textos en inglés. Respecto a medir el nivel de sentimiento cuenta con dos parámetros, polaridad y subjetividad. En nuestro caso nos centraremos en la polaridad ya que nos ofrece el nivel de positivo o negativo que es el json pasado. La polaridad se mide entre -1 y 1 siendo 0 neutral.

### 4.14. Yandex Translate

Yandex [34] es una herramienta gratuita para realizar la detección de idioma y traducción de textos. La hemos utilizado para traducir los textos de español a inglés y poder analizar esos textos con la librería para textos en inglés. De esta forma podremos realizar el mismo análisis a todos los textos.

### 4.15. Heroku

Heroku [3] es un PaaS, plataforma como servicio, que nos brinda un servidor para poder desplegar la aplicación. A diferencia de otras plataformas, permite añadir funcionalidades extra, desarrollar en varios lenguajes de programación y desplegar versiones.

### 4.16. Drawio

Drawio [10] es una herramienta online para realizar diagramas de varios tipos, entidad-relación, UML,... Se ha utilizado para generar tanto el diagrama de secuencias como el diagrama de casos de uso.

### 4.17. Statsmodels

Statsmodels [1] es un módulo de Python para realizar operaciones estadísticas. En nuestro caso se ha utilizado para el cálculo de series temporales. Cuenta con los métodos para calcularlas, además de test para saber si es o no estacionaria y para realizar la descomposición de la serie, tanto manual como automática.

---

# Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación.

## Inicio del proyecto

El tema del proyecto surgió buscando tecnologías que pudieran destacar en un proyecto.

El análisis de sentimiento en texto está muy extendido para los textos en inglés y se nos ocurrió que hacerlo para castellano sería algo destacable.

Además, el uso de redes sociales en los últimos años está muy extendido y utilizarlo en el proyecto podría permitir la recogida de datos muy variados.

Jose Manuel, el tutor, ya había llevado varios proyectos que engloban las redes sociales por lo que era muy útil para dudas en cuanto a la extracción de datos de las APIs. Además de tener varios conocimientos en las series temporales.

Por ello pensamos en una aplicación web que uniera el análisis de sentimiento junto con redes sociales y generación de gráficos y series temporales.

Así fue como surgió la idea de crear la aplicación Sentinel.

## Metodologías

La metodología utilizada ha sido SCRUM, metodología aprendida en la asignatura de gestión de proyectos. Se basa en utilizar la gestión ágil para el proyecto, aunque no se ha seguido estrictamente, sí se han realizado algunos de los eventos que identifican esta metodología.

El desarrollo se ha dividido en sprints, en los cuales se marcaban una serie de objetivos y se estimaba su valor mediante *Story points*. Para esta manera de gestión ha sido muy útil la herramienta Zenhub, la cual nos proporciona un tablero donde podemos interactuar con los distintos estados en los que puede estar una tarea.

En cuanto a las reuniones, se han juntado la reunión de revisión del sprint, retrospectiva del sprint y planificación del sprint. Aunque se realizase en una única reunión se seguía el orden de realización.

- Revisión del sprint: Como su nombre indica se habla de cómo ha ido el sprint, los objetivos que se han conseguido y se muestran los avances.
- Retrospectiva del sprint: Se habla de los problemas que se han tenido durante el sprint y lo que sí ha funcionado. Se usa para ver qué aspectos deben cambiar para mejorar la calidad del trabajo.
- Planificación del sprint: Al comienzo de cada sprint se habla de los objetivos a conseguir durante ese sprint y la estimación de trabajo que va a llevar cada uno.

## Formación

Para la realización del proyecto se ha necesitado la investigación sobre varios conceptos con los cuales no se habían tratado durante el grado. A continuación se comenta cuáles han sido.

### Sentiment Analysis

Es el tema principal del proyecto, por lo que era esencial buscar información sobre el tema y conocer cómo funciona. Para ello se consultaron algunos libros de la biblioteca de la universidad online y varias webs.

Se buscaron librerías que realizaran los análisis y explicaran qué criterios seguían a la hora de hacerlo.



## **Flask**

Para crear la aplicación se utilizó el framework flask, ya que es muy sencillo de utilizar y de instalar. Permite crear un servidor en Python de forma rápida y favorece el patrón de diseño de MVC, ya que podemos diferenciar bien las partes del modelo, vista y controlador.

Se contempló la opción de utilizar Django, el cual nos brinda la misma funcionalidad que flask pero está orientado a proyectos más grandes.

## **Angular**

Para el desarrollo del Frontend se pensó que era interesante añadir el framework angular.

Como no se había aprendido durante el grado ha sido bastante difícil su aprendizaje, pero gracias a la web de Angular, <https://angular.io/>, que ofrece varios ejemplos con el paso a paso, se consiguió aprender cómo realizar servicios y sus comunicaciones con el Back end.

## **Series temporales**

[13] Además de los gráficos que muestran los datos almacenados, se ha decidido introducir series temporales. Para el aprendizaje, los tutores han proporcionado un curso sobre estas que se impartió para ellos en la UBU, tanto en lenguaje R como en Python.

Además del curso, también han proporcionado una serie de PDFs que permiten la comprensión y el aprendizaje de las series a nivel teórico.

## **Desarrollo del proyecto**

Durante las primeras semanas se eligió el entorno de desarrollo, se investigó sobre trabajos relacionados, sobre flask, el editor de textos para la memoria, etc.

Tras esto, había que extraer los datos de las APIs, generar el prototipo de la interfaz para la aplicación, conectar nuestra aplicación con la base de datos, conectar flask con angular.

```
user_db = os.getenv('USER_DB')
pass_db = os.getenv('PASSWD_DB')
mydb = mysql.connector.connect(host="localhost", user=user_db, passwd=pass_db, database="telusko")

mycursor = mydb.cursor()
```

Figura 5.1: Conexión desde Python con MySQL.

```
user_db = os.getenv('USER_DB')
pass_db = os.getenv('PASSWD_DB')
mydb = mysql.connector.connect(host="localhost", user=user_db, passwd=pass_db, database="telusko")

mycursor = mydb.cursor()
```

Figura 5.2: Conexión desde Python con MySQL.

Realizadas todas las conexiones necesarias, se comenzó a buscar sobre librerías de sentiment analysis y como sería su integración con el proyecto.

Mientras, se escogió el template para angular y se ajustó a la aplicación. Después se eligió el segundo template, el de gráficos y se integró con el resto de la aplicación.



Figura 5.3: Página de inicio de la aplicación.

Tras haber realizado los métodos de cálculo de análisis, se realizaron los cálculos estadísticos para almacenarlos en una nueva tabla en la base de datos.

Estadísticas

id	media	mediana	moda	varianza	desviación típica
#ubu	0.11727	0.0244381	0.00753921	0.0202485	0.142297

Figura 5.4: Cálculos estadísticos.

Teniendo ya la parte back end con los cálculos realizados y el front end con los templates bien integrados se comenzó a implementar varias funcionalidades como la de login o la del registro, y una de las principales, la visualización de los resultados en gráficos.

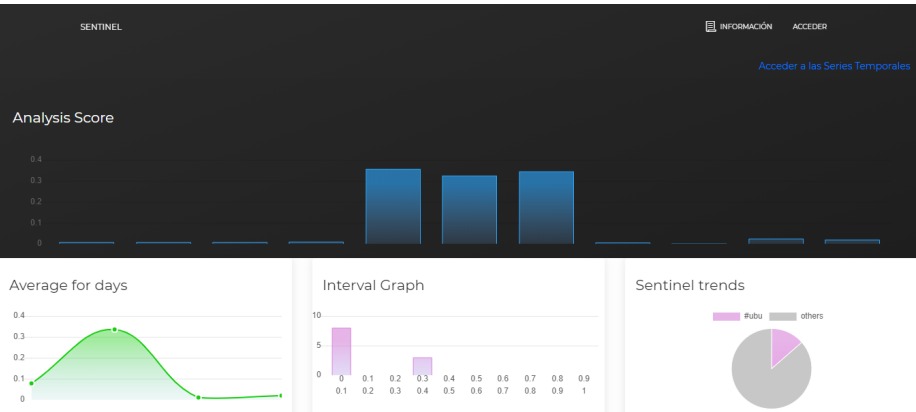


Figura 5.5: Gráficos implementados.

Para ello fue necesario aprender a realizar peticiones desde la interfaz al servidor y saber formar el json correctamente en el servidor para que la información enviada fuera accesible.

```

@Injectable({
  providedIn: 'root'
})
export class TwitterService {
  twitterUrl = 'http://127.0.0.1:5000/idTwitterInDB';
  searchTwitterUrl = 'http://127.0.0.1:5000/searchTwitter';
  constructor(private http: HttpClient) { }

  checkIdInDataBase(id: string) {
    const params = new HttpParams().set('id', id);
    return this.http.get(this.twitterUrl, {params: params});
  }

  searchIdInApi(id: string) {
    const params = new HttpParams().set('id', id);
    return this.http.get(this.searchTwitterUrl, {params: params})
  }
}

```

Figura 5.6: Servicio de twitter.

```

this.timeSeriesService.timeSeriesChart(this.id, this.since_date, this.until_date, this.is_tw,
  type, schema, num_periods).subscribe( next: response => {
  this.estacionaria = response['estacionaria'];
}

```

Figura 5.7: Acceso a la información enviada del servidor a la interfaz.

Además, se implementó la posibilidad de elegir entre qué fechas se quieren mostrar los resultados.

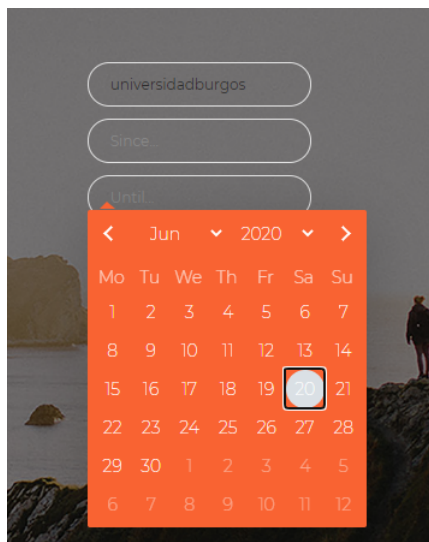


Figura 5.8: Filtrar por fecha.

Hasta el momento los resultados que se podían mostrar estaban previamente guardados, por lo que se añadió la opción de que aunque el identificador buscado no se encuentre en base de datos se busque en las APIs, se guarde en base de datos y se muestren los resultados.

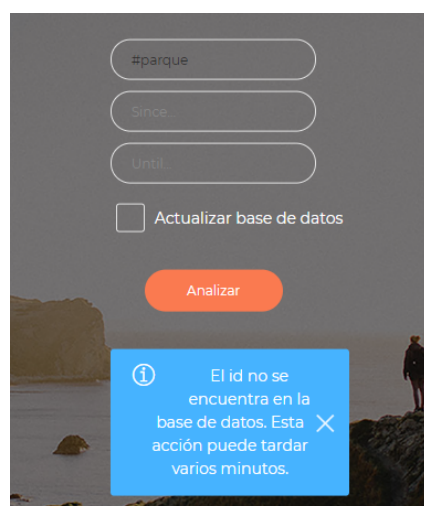


Figura 5.9: Funcionalidad para buscar un id que no está en la base de datos.

Además se añadió la opción de actualizar los identificadores que ya se encuentra en la base de datos.



Figura 5.10: Funcionalidad para actualizar base de datos.

En cuanto a la parte de desarrollo, lo último ha sido la funcionalidad de calcular y mostrar series temporales.

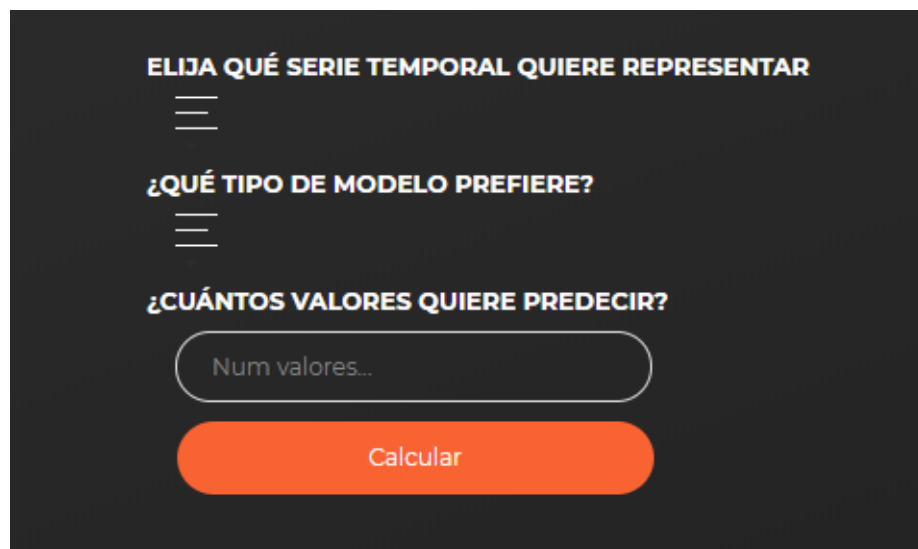


Figura 5.11: Menú de elección para el cálculo de series temporales.

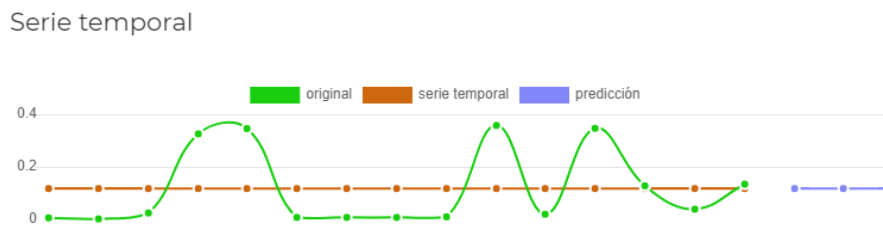


Figura 5.12: Gráfica de series temporales.

La última implementación que se ha hecho ha sido la internacionalización, para la cual se han creado dos archivos de extensión json con las traducciones, unas en español y las otras en inglés.

```
"TIME_SERIES": {
  "CHOOSE_TS": "Choose which time series you want to plot",
  "TIT_TS_MENU": "Time series",
  "OPTION1_TS": "Exponential Smoothing",
  "OPTION2_TS": "Holt Method",
  "OPTION3_TS": "Arima Method",
  "CHOOSE_SCHEMA": "What kind of model do you want?",
  "TIT_MODEL_MENU": "Model",
  "OPTION1_SCHEMA": "Additive",
  "OPTION2_SCHEMA": "Multiplicative",
  "CHOOSE_PRED": "How many values do you want to predict?",
  "BUTTON": "Calculate",
  "STATIONARY": "The time series is stationary",
  "NOSTATIONARY": "The time series isn't stationary",
  "TIMESERIE": "Time series",
  "SEASONALITY": "Seasonality",
  "TREND": "Trend",
  "RESIDUE": "Residue",
  "MENU": "Back to menu"
},
```

Figura 5.13: Archivo json con las traducciones al inglés.

Todas estas implementaciones se han complementado con la documentación sobre el proyecto en memoria y anexos.

## **Resolución de problemas técnicos**

Ha habido algunos problemas a lo largo del desarrollo del proyecto, que se han solventado con éxito.

### **Problemas con Angular**

La mayoría de los problemas que se han tenido han sido con Angular ya que es un framework que nunca se había utilizado.

El primer problema fue conseguir ajustar el template de gráficos al template de angular que se ha utilizado para el resto de ventanas. Había varios conflictos entre los módulos ya que las versiones no coincidían. Además, algunas de las últimas versiones no son soportadas por estos templates por lo que daba varios errores y no se conseguía levantar la interfaz de usuario.

Ya conseguida la unión de los templates, hubo que aprender a crear servicios y enviar peticiones al servidor y desde este mandar un objeto json bien formado. Para ello fueron muy útiles los ejemplos con pasos a seguir de la web de Angular como se ha comentado anteriormente.

Para la creación de los gráficos, ha sido necesario añadir un tiempo de parada entre los distintos servicios porque se solapaban entre ellos lo que resultaba en una petición de datos fallida con un código de error 500.

### **Problemas con series temporales**

Inicialmente las series temporales a calcular iban a ser el suavizado exponencial, método de Holt y método de Holt-Winters.

Para realizar los cálculos de series temporales es necesario un periodo, el nuestro ha tenido que ser 1 ya que los datos se recogen y almacenan basándonos en el día en el que se crearon. Por ello, los periodos que tendrían sentido serían el 1 o el 7. Por ello, se creó un desplegable para que el usuario pudiera escoger entre los dos valores. Pero, al realizar cálculos con el periodo 7 la descomposición de la serie temporal no se realizaba correctamente en ningún caso, por lo que se optó por tener un periodo de 1 siempre.

El problema de tener este valor como periodo, es que el método de Holt-Winters no se podía calcular ya que al ser los valores de análisis tan pequeños, se generaban valores no reales que no se podían representar. Por ello, decidimos eliminar esta serie temporal.

Pero en su lugar implementamos el modelo Arima, el cual es mucho más complejo y nos permite realizar predicciones más reales.



## **Problemas con Heroku**

Para la presentación de la aplicación se decidió desplegar en Heroku, que es un hosting gratuito donde se puede alojar tu página web. Hubo varios problemas desde el principio ya que el proyecto tiene dos partes, la parte del servidor en la que se realizan todas las operaciones y la interfaz de usuario. Como están escritas en diferentes lenguajes fue necesario separar las partes para que Heroku pudiera reconocer ambas. Tras conseguir esto, hubo que modificar algunas partes del código, como las URLs a las que se envían las peticiones, ya que apuntaban al Localhost y ya no iba a corresponder.

Cuando se realizaron las pruebas para ver que todo funcionaba correctamente, se notó la lentitud de la aplicación, pero al ser un hosting gratuito es normal ya que no nos ofrecen mucha memoria.

La aplicación funciona, pero al tener tantos servicios con peticiones agotamos la memoria RAM que nos ofrece Heroku y el rendimiento de la aplicación no es el esperado.

Para poder tener un buen rendimiento, sería necesario pagar una licencia que podría ascender a 250 €.

Por ello se ha optado por crear una máquina virtual en la que el tribunal pueda realizar las pruebas que considere con un rendimiento óptimo.

## **Problemas con la máquina virtual**

Como se ha indicado en el apartado anterior, se ha considerado presentar el proyecto tanto en heroku como en una máquina virtual. Inicialmente se creó una de Ubuntu, pero al empezar a instalar las dependencias se hizo imposible ya que daban fallos.

Se siguió con una Windows 7, la cual no nos permitió tampoco ya que indicaba que el sistema operativo estaba desactualizado para la version de Visual C++ Tools, las cuales son necesarias para las librerías de análisis de Python.

Por lo cual la solución ha sido comprar una licencia OEM Windows 10 Pro. Con esta se ha conseguido instalar todas las dependencias sin problema y tener la máquina virtual operativa.

Si no se ejecuta la máquina virtual en un ordenador lo suficientemente potente la aplicación puede sufrir fallos de llenado de memoria RAM, como sucedía en Heroku.



---

## Trabajos relacionados

---

En este apartado vamos a ver varias herramientas que hacen uso del sentiment analysis.

### 6.1. MonkeyLearn

MonkeyLearn [15] es una aplicación que utiliza machine learning para analizar el texto. Puede integrarse con aplicaciones como Excel, Google Sheets, Zapier, Gmail, Twitter, etc.

Además de poder utilizar su analizador de texto, te permite entrenarlo de forma personalizada añadiendo palabras que quieras destacar.

Respecto al precio, tiene tres planes diferentes:

- Un plan gratuito que está bastante reducido ya que te permite analizar 300 veces al mes, a baja velocidad y además no te permite ver los resultados en gráficos.
- Un plan llamado “Team” que permite analizar hasta 10000 veces al mes con una velocidad media, se pueden definir hasta 3000 palabras nuevas para entrenar el algoritmo, pero tampoco nos permite mostrar los resultados en gráficos. El precio es de 299\$ por mes.
- Un plan llamado “Business” que es personalizado, nosotros nos encargaremos de decidir cuántas queries queremos analizar al mes, la velocidad de la aplicación, las palabras a definir. Además, los resultados se mostrarán en gráficos y flujogramas. El precio será variable.

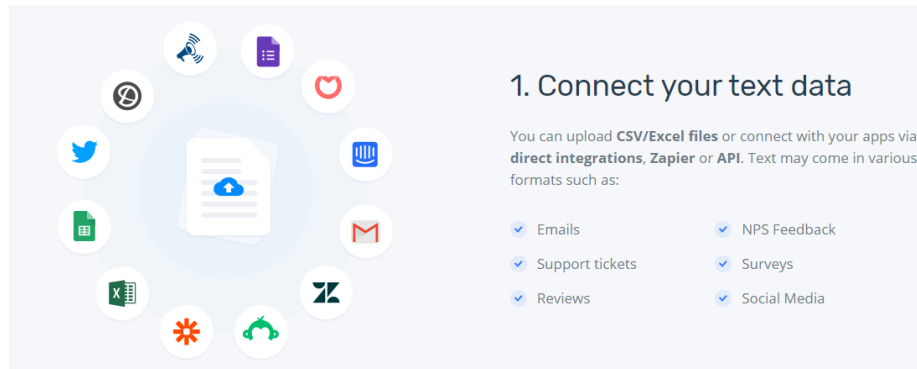


Figura 6.14: Monkeylearn connect your text data.

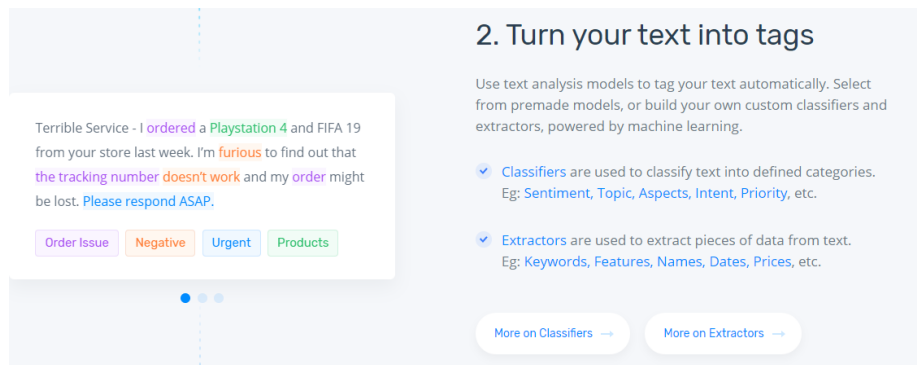


Figura 6.15: Monkeylearn turn your text into tags.

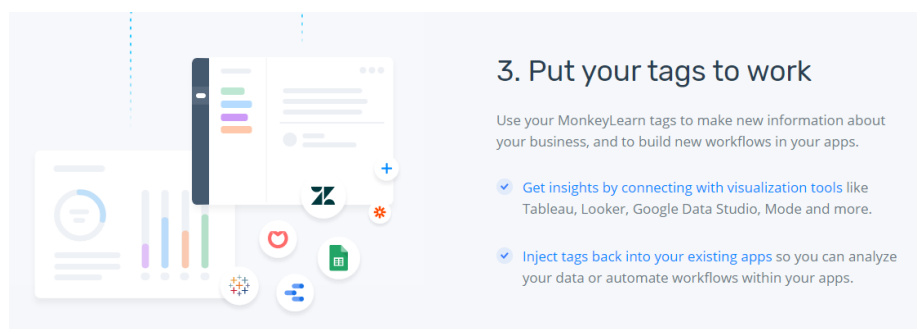


Figura 6.16: Monkeylearn put your tags to work.

## 6.2. Lexalytics

Es una aplicación que se basa en el Natural Language Processing para crear distintas librerías y modelos. Lexalytics [14] también utiliza machine learning para entrenar a los modelos.

Además de utilizar sentiment analysis en los textos, los separan en categorías dependiendo del tema que se trate en el texto y extraen las distintas entidades como la persona que ha escrito, el lugar desde donde se escribe, los productos sobre los que se escriben, etc.

El precio de la aplicación es variable dependiendo de cuántos textos queramos analizar, el tiempo que vayamos a suscribirnos, etc.

This document is: **neutral (+0.123)**

valuable assets willing love essentially  
respect be careful severe  
sophisticated precautionary injury  
worried good impressing be healed  
victory gained professional loss  
all set top

Scroll down for full report



Figura 6.17: Lexalytics análisis de texto

Extracted themes	Evidence	Sentiment
acquiring linebacker	7	+0.55
big return	7	+0.19
boarding school	7	+1.70
broken leg	7	-0.67
came time	7	+1.70
enough respect	7	+1.74
football game	7	+1.70
healthy Giants	7	-0.17
hockey year-round	7	+1.70
in-season trade	7	+0.55
junior hockey star	7	-0.17
junior level	7	-0.88

Figura 6.18: Lexalytics entidades.

**Highlight:** ☐ Phrases ☐ Themes ☒ Entities

EAST RUTHERFORD, N.J. — Leaving the practice field Monday morning, **Giants** Coach **Tom Coughlin** was asked about the limited number of running backs he had available during the workout.

“What do you mean?” Coughlin asked. “We had one. We’re all set.”

The remaining healthy **Giants** running back is the rookie **Michael Cox**, a seventh-round draft pick who was a junior hockey star growing up in **Boston** and did not play organized football until he was 17.

“It was a little strange being the only one out there for practice,” Cox said Monday. “But I feel like I’m ready to go. They said I’m going to be playing.”

If he can stay healthy until then, Cox is a near certainty to get the ball when the **Giants** play the **Minnesota Vikings** on Monday because so many other **Giants** running backs are injured. The starting running back **David Wilson** is out with a neck injury, which will not need surgery, but is likely to keep him out the next three weeks. **Andre Brown**, who was supposed to share the rushing duties with Wilson, is not scheduled to return from his broken leg until **Nov. 10**.

Da’Rel Scott, waived **Oct. 1** and re-signed a week later, has a severe hamstring strain that will not be healed for weeks. And **Brandon Jacobs**, the new starter, did not practice on Monday, resting a hamstring strain he sustained during his 22 carries in the **Giants** loss to the Bears.

Current Character Count: 4254 / 16384

Figura 6.19: Lexalytics entidades subrayadas en el texto.

## 6.3. Brandwatch

Brandwatch [16] es una aplicación que permite a las marcas saber que opina la gente sobre ellas en las redes sociales e internet mediante inteligencia artificial.

Permite categorizar los datos y entrenar el modelo de forma personalizada añadiendo palabras que queramos que identifique como positivas o negativas en caso de estar en algún comentario. Permite mostrar los datos recogidos en distintos tipos de gráficos y pudiendo personalizarlos para que muestre las categorías que queramos.

Además, nos permite analizar distintos perfiles de personas para poder realizar campañas de marketing que lleguen a las personas que queramos enfocar nuestra empresa.

Los precios varían dependiendo de los planes:

Plan	Descripción	Precio
<i>Brandwatch/Pro</i>	Está orientado a analistas de marcas pequeñas	653,59\$/mes
<i>Enterprise/M</i>	Para empresas de marcas algo más conocidas y por tanto con mayor volumen de opiniones	2614,37\$/mes
<i>Enterprise/Q</i>	Orientado a empresas importantes y que tienen muchos comentarios que analizar	2614,37\$/mes

Tabla 6.1: Precios de Brandwatch.

Además cuentan con planes personalizados para agencias.



Figura 6.20: Brandwatch gráfico de líneas.

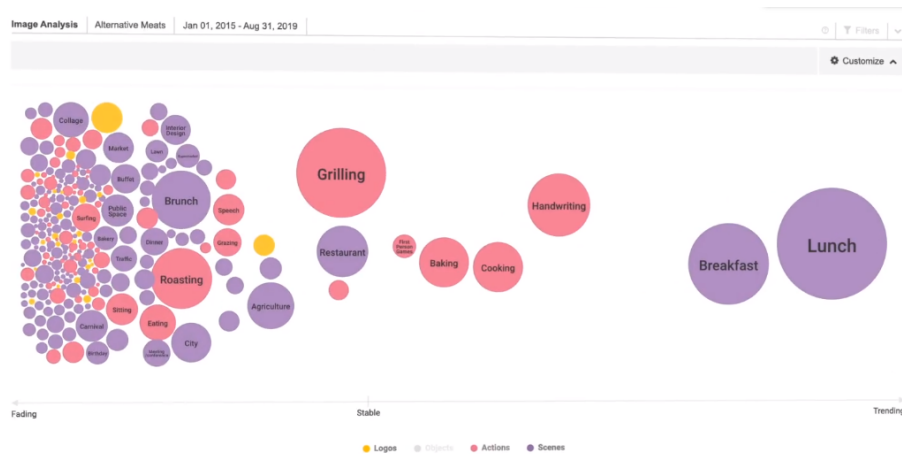


Figura 6.21: Brandwatch gráfico de burbujas.



---

# Conclusiones y Líneas de trabajo futuras

---

## 7.1. Conclusiones

Una vez finalizado el proyecto y remontándome al comienzo del mismo puedo afirmar que he adquirido muchos conocimientos.

El resultado de la aplicación cumple con todos los objetivos marcados inicialmente y además se han añadido implementaciones que no se pensaron en un primer momento pero que dan al proyecto mayor funcionalidad.

Por ejemplo, en un primer momento se pensó que la aplicación simplemente iba a obtener los resultados que estuvieran en la base de datos ya almacenados. Pero a mitad del proyecto se pensó que podía ser muy interesante tener una opción de 'Actualizar la base de datos' y también la opción de buscar un identificador que no se encontrara almacenado. De esta forma, nuestra aplicación accede a las APIs de Twitter e Instagram en tiempo real y recolecta datos nuevos para almacenarlos en la base de datos.

He aprendido mucho sobre las aplicaciones web, los protocolos utilizados y las peticiones a través de ellos. En cuanto a los lenguajes, he reforzado los conocimientos que tenía sobre Python y SQL, ya que se cursan a lo largo del grado, y he aprendido el lenguaje TypeScript. También he aprendido sobre HTML, sobre el cual tenía unos conocimientos muy vagos.

Personalmente, lo que más me ha costado y creo que he aprendido mucho, ha sido cómo conectar la interfaz de usuario de una aplicación con el controlador de la misma. Para mí utilizar servicios y realizar peticiones era algo totalmente nuevo, pero al final he conseguido aprender, y aunque

me queda mucho por mejorar, he conseguido realizar las peticiones de forma exitosa.

En cuanto a las metodologías, las cuales se tratan en la asignatura de gestión de proyectos, se ha usado Scrum que actualmente está muy presente en la mayoría de los trabajos. Utilizarla ha sido clave para conseguir una buena organización y aprender a estimar cuanto tiempo gastamos en cada tarea.

Por todo ello, creo que el balance ha sido bueno aun con los problemas que se han tenido durante el desarrollo del proyecto.

## **7.2. Líneas de trabajo futuras**

- Crear varios roles de usuario.
- Implementar el acceso con cuentas de otras aplicaciones como Google, Facebook, etc.
- Implementar seguridad en la aplicación.
- Realizar sentiment analysis mediante técnicas de machine learning.
- Añadir más redes sociales de las que recabar información.
- Utilizar scraping para obtener información.

---

## Bibliografía

---

- [1] Introduction — statsmodels. <https://www.statsmodels.org/stable/index.html>, June 2020.
- [2] Suavización exponencial simple. <https://www.ingenieriaindustrialonline.com/pronostico-de-la-demanda/suavizacion-exponencial-simple/>, June 2020. Library Catalog: [www.ingenieriaindustrialonline.com](http://www.ingenieriaindustrialonline.com).
- [3] ¿Qué es Heroku y para qué me sirve? <https://platzi.com/blog/que-es-heroku-y-para-que-me-sirve/>, June 2020.
- [4] Andrés. MySQL vs SQLite: ¿Cuál es mejor? Comparativa. <https://guiadev.com/mysql-vs-sqlite/>, February 2020.
- [5] Gari Araolaza. Pencil: herramienta para prototipado web. <https://www.codesyntax.com/es/blog/pencil-herramienta-para-prototipado-web>, February 2020.
- [6] ayllote. ayllote/senti-py. <https://github.com/ayllote/senti-py>, April 2020.
- [7] Bbva. API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>, April 2020.
- [8] Luciano Castillo. Introducción — Conociendo GitHub 0.1 documentation. <https://conociendogithub.readthedocs.io/en/latest/data/introduccion/>, February 2020.

- [9] Culturacion. ¿Qué es y para qué sirve un web service? <https://culturacion.com/que-es-y-para-que-sirve-un-web-service/>, April 2020.
- [10] draw.io. draw.io - Diagrams For Everyone, Everywhere. <https://drawio-app.com/>, June 2020. Library Catalog: drawio-app.com.
- [11] Ionos. URI (identificador de recursos uniformes). <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/uri-identificador-de-recursos-uniformes/>, April 2020.
- [12] JetBrains. Desarrollo web full stack - Funcionalidades | PyCharm. [https://www.jetbrains.com/pycharm/features/web\\_development.html](https://www.jetbrains.com/pycharm/features/web_development.html), February 2020.
- [13] Manoj K. Malhotra Lee J.Krajewski, Larry P.Ritzman. *Operations Management - Processes and Supply Chains*. June 2020.
- [14] Lexalytics. Lexalytics Intelligence Platform | Lexalytics. <https://www.lexalytics.com/platform>, January 2020.
- [15] MonkeyLearn. MonkeyLearn - How it Works. <https://monkeylearn.com/how-it-works/>, January 2020.
- [16] Neoattack. ¿Qué es Brandwatch y para qué sirve? - Neo Wiki. <https://neoattack.com/neowiki/brandwatch/>, January 2020.
- [17] WWWhat's new. zenhub, plataforma de gestión de proyectos integrada en Github, ahora gratis para estudiantes. <https://wwwwhatsnew.com/2015/09/24/zenhub-plataforma-de-gestion-de-proyectos-integrada-en-github-ahora-gratis> February 2020.
- [18] Openwebinars. Django vs Flask. <https://openwebinars.net/blog/django-vs-flask/>, February 2020.
- [19] Openwebinars. Qué es flask y ventajas que ofrece. <https://openwebinars.net/blog/que-es-flask/>, February 2020.
- [20] José Manuel Galán Ordax. Evaluación integradora de políticas de agua: modelado y simulación con sociedades artificiales de agentes. tesis doctoral, June 2020.
- [21] Overleaf. About us. <https://www.overleaf.com/about>, February 2020.

- [22] Soujanya. Poria. *Multimodal Sentiment Analysis*. Socio-Affective Computing, 8. Springer International Publishing, Cham, 1st ed. 2018. edition, April 2020.
- [23] Sergio Gutiérrez Pérez. Control de Versiones: ¿Por qué GitKraken? <https://medium.com/@sergupe6/control-de-versiones-por-qué-gitkraken-eelf30b4a18f>, February 2020.
- [24] RedHat. ¿Qué es una API? <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>, April 2020.
- [25] RestfulApi. Idempotent REST APIs – REST API Tutorial. <https://restfulapi.net/idempotent-rest-apis/>, April 2020.
- [26] Margaret Rouse. What is an API endpoint? Definition from WhatIs.com. <https://searchapparchitecture.techtarget.com/definition/API-endpoint>, April 2020.
- [27] UAM Santiago de la Fuente Rodríguez. Series temporales. <http://www.estadistica.net/ECONOMETRIA/SERIES-TEMPORALES/modelo-clasico.pdf>, June 2020.
- [28] TextBlob. TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation. <https://textblob.readthedocs.io/en/dev/>, April 2020.
- [29] Desarrollo web. Qué es Python. <https://desarrolloweb.com/articulos/1325.php>, February 2020.
- [30] Wikilibros. Manual de LaTeX - Wikilibros. [https://es.wikibooks.org/wiki/Manual\\_de\\_LaTeX](https://es.wikibooks.org/wiki/Manual_de_LaTeX), February 2020.
- [31] Wikipedia. MySQL. <https://es.wikipedia.org/w/index.php?title=MySQL&oldid=123033647>, February 2020.
- [32] Wikipedia. Protocolo de transferencia de hipertexto. [https://es.wikipedia.org/w/index.php?title=Protocolo\\_de\\_transferencia\\_de\\_hipertexto&oldid=125068560](https://es.wikipedia.org/w/index.php?title=Protocolo_de_transferencia_de_hipertexto&oldid=125068560), April 2020.
- [33] Wikipedia. Zotero. <https://es.wikipedia.org/w/index.php?title=Zotero&oldid=123344086>, February 2020.
- [34] Yandex. About - Yandex.Translate. Help. <https://yandex.com/support/translate/>, April 2020.