

---

## Proyecto 3 – Backend y Frontend

---

201902672 – Jairo Adeldo Gómez Hernández

### Resumen

Este ensayo presenta una aplicación que combina un Backend en Flask con un Frontend en Django para analizar el sentimiento de los mensajes en archivos XML. La aplicación utiliza una API construida en Flask para recibir y procesar los archivos XML, extrayendo menciones a usuarios, hashtags y un archivo de configuración. Luego, utiliza el archivo de configuración para analizar el contenido de los mensajes y determinar su tono, etiquetándolos como positivos o negativos.

La aplicación también ofrece capacidades de consulta que permiten conocer cuántos usuarios han sido mencionados en cada fecha y cuáles son los hashtags utilizados en esas fechas. Esta funcionalidad es esencial para el análisis de tendencias y la comprensión de la participación de los usuarios en la plataforma.

La integración entre el Backend de Flask y el Frontend de Django permite a los usuarios interactuar de manera efectiva con la aplicación y obtener información valiosa sobre el uso de menciones y hashtags en función de las fechas. Esta arquitectura proporciona una solución sólida y completa para el análisis de contenido en redes sociales, con aplicaciones potenciales en la gestión de la reputación en línea y la toma de decisiones

empresariales basadas en datos extraídos de redes sociales.

### Palabras clave

Backend, Frontend, Flask, Django, API.

### Abstract

*This essay presents an application that combines a backend in Flask with a frontend in Django to parse the sentiment of messages in XML files. The application uses an API built in Flask to receive and process XML files, extracting user mentions, hashtags, and a configuration file. It then uses the configuration file to parse the content of the messages and determine their tone, labeling them as positive or negative.*

*The application also offers query capabilities that let you know how many users have been mentioned on each date and which hashtags were used on those dates. This functionality is essential for trend analysis and understanding user engagement on the platform.*

*The integration between the Flask backend and the Django frontend allows users to effectively interact with the application and obtain valuable*

*information about the use of mentions and hashtags based on dates. This architecture provides a robust and complete solution for social media content analysis, with potential applications in online reputation management and business decision making based on data extracted from social networks.*

## Keywords

*Backend, Frontend, Flask, Django, API.*

## Introducción

Esta aplicación de análisis de contenido en redes sociales, desarrollada con una arquitectura de dos capas, destaca por su capacidad para procesar mensajes contenidos en archivos XML. El Backend, construido en Flask, se encarga de la recopilación y procesamiento de datos, extrayendo elementos clave como menciones a usuarios, hashtags y, de manera sobresaliente, evaluando el tono de los mensajes a través de un algoritmo de análisis de sentimiento. Mientras tanto, el Frontend, diseñado en Django, proporciona una interfaz amigable para que los usuarios carguen archivos XML, realicen análisis de sentimiento y accedan a información relevante sobre menciones y hashtags, organizada cronológicamente.

Esta arquitectura de dos capas ofrece una solución completa y sólida para el análisis de contenido en redes sociales, con aplicaciones significativas en la gestión de la reputación en línea y en la toma de decisiones empresariales basadas en datos extraídos de las redes sociales. En el siguiente ensayo, exploraremos en detalle la implementación, el funcionamiento y el impacto de esta innovadora aplicación en el panorama de la inteligencia empresarial y la analítica de redes sociales.

## Desarrollo del tema

### A. Funciones del Sistema.

1. Interfaz gráfica, página web usando Django que nos dará las funcionalidades del proyecto.
2. Carga de Mensajes
3. Carga de Configuración
4. Resetear Datos
5. Consultas:
  1. Consultar Hashtags
  2. Consultar Menciones
  3. Consultar Sentimientos de los Mensajes
6. Grafica:
  1. Grafica Hashtags
  2. Grafica Usuarios Mencionados
  3. Grafica Sentimiento de los Mensajes
7. Ayuda:
  1. Información del Estudiante
  2. Documentación

### B. Explicación del Programa.

La aplicación iniciara en una interfaz web, hecha con HTML, usando estilos de Bootstrap y CSS. Usando Django para ejecutar la API hecha con Flask.

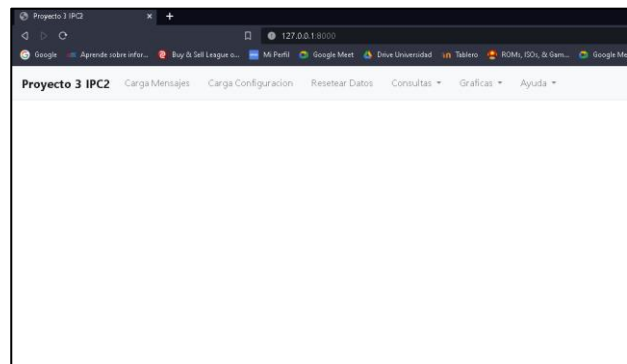


Figura 1. Interfaz Gráfica.

Fuente: elaboración propia

En caso el usuario seleccione y entre a la opción Cargar Mensajes, nos desplegara una pagina web, en la cual nos permitirá subir archivos con el formato XML, al cargar un archivo, este mostrara el contenido en el cuadro, al enviar, este consumirá nuestro método POST del API, y almacenara el contenido en una bd.xml para así clasificar la información.

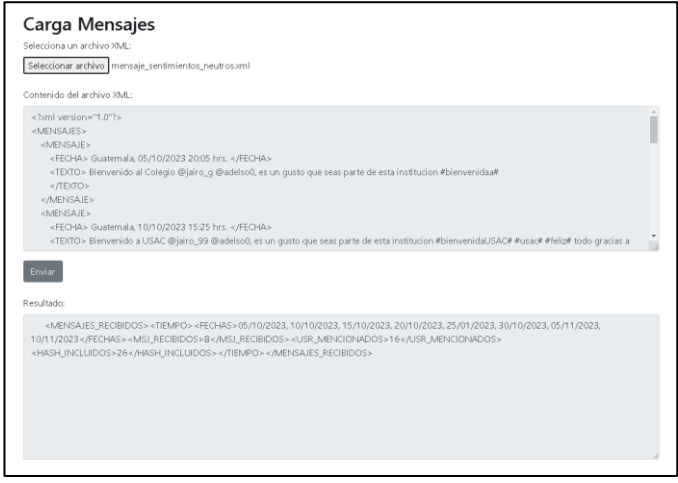


Figura 2. Carga de Mensajes.

Fuente: elaboración propia

Al darle enviar hará el uso de la API, la cual es un POST que recorrerá e ira procesando para almacenarlo.

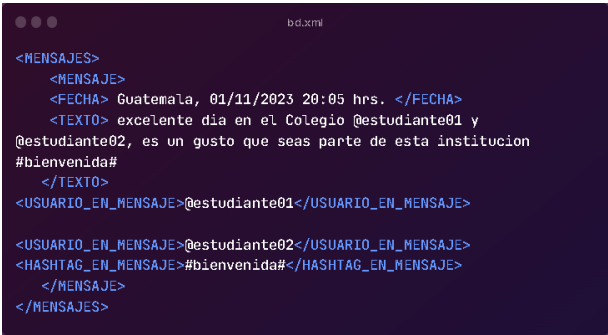


Figura 3. Guardado de Mensajes.

Fuente: elaboración propia

Al seleccionar la opción de Cargar Configuración, nos desplegara una pagina web, muy similar al cargar mensajes, pero en esta subiremos un archivo XML que contiene las palabras claves, con las cuales podremos saber si el mensaje es positivo, negativo o neutro, decidimos enviar, este consumirá nuestro método POST del API, y almacenara el contenido en una config.xml para así clasificar la información.

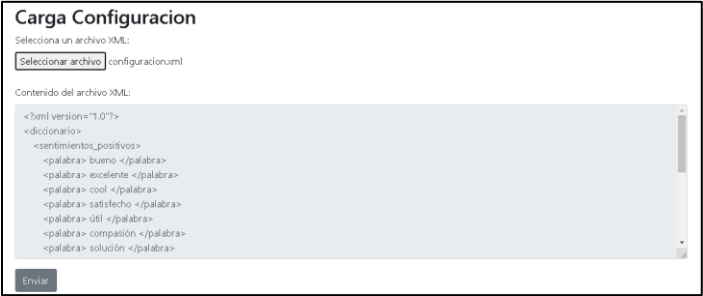


Figura 4. Carga de Configuración.

Fuente: elaboración propia

Al darle enviar en el recuadro de resultados nos mostrara cuantos mensajes positivos se guardaron y cuantos negativos fueron almacenados en config.xml

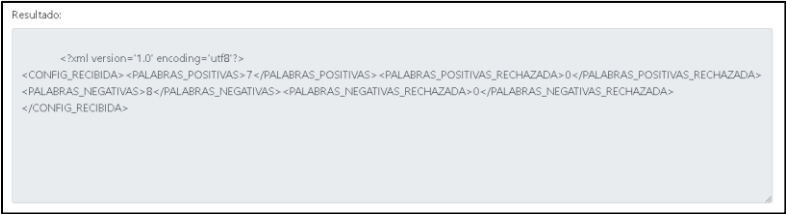


Figura 5. Carga de Configuración.

Fuente: elaboración propia

Al darle enviar hará el uso de la API, la cual es un POST que recorrerá e ira procesando para almacenarlo.



Figura 6. Guardado de Configuración.  
Fuente: elaboración propia

Al seleccionar la opción Resetear Datos, nos mostrará esta página web, en la cual su función será eliminar los archivos de bd.xml y config.xml, esto para comenzar un nuevo procedimiento sin necesidad de parar la aplicación.

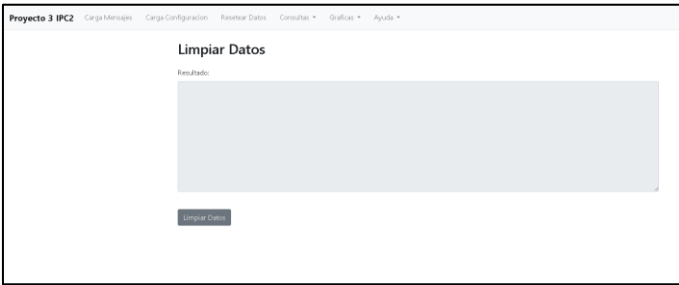


Figura 7. Pagina Limpiar Datos.  
Fuente: elaboración propia

En esta página, al seleccionar la opción Limpiar Datos haremos uso del API “limpiarDatos” el cual se encarga de eliminar los archivos de bd.xml y config.xml que son los que almacenan la información cargada, esto para eliminar los datos, sin necesidad de reiniciar la aplicación.



Figura 8. Resultado de Limpiar Datos.  
Fuente: elaboración propia

En caso seleccionar la opción Consultas, se nos desplegara una lista de consultas, las cuales son: Consultar Hashtags, Consultar Menciones, Consultar Sentimientos. Esto mostrando una pagina web para cada uno, haciendo uso de los métodos GET de nuestra API, para así extraer la información de nuestros documentos bd.xml y config.xml

Al seleccionar Consultar Hashtags, nos mostrara una pagina web en la cual veremos todos los hashtags cargados en los mensajes, ordenados por fecha. Esto haciendo uso de nuestra API, de un método GET, get\_hashtags el cual obtiene los hashtags y la cantidad de veces que fueron usados en los mensajes.

**Hashtags**

Fecha de inicio: dd/mm/aaaa Fecha de fin: dd/mm/aaaa

Buscar Resetear

Fecha	Hashtag
20/10/2023	#bienvenida#: 1 mensaje(s)
25/10/2023	#bienvenidaico#: 1 mensaje(s)
30/10/2023	#usac#: 1 mensaje(s) #feliz#: 1 mensaje(s)
05/11/2023	#usac#: 2 mensaje(s) #feliz#: 2 mensaje(s) #si#: 1 mensaje(s) #cale#: 1 mensaje(s) #ipc2#: 1 mensaje(s)
10/11/2023	#feliz#: 3 mensaje(s) #descanso#: 1 mensaje(s) #vacaciones#: 1 mensaje(s)
01/11/2023	#bienvenida#: 2 mensaje(s)

Figura 9. Consultar Hashtags.  
Fuente: elaboración propia

Esta consulta se puede hacer filtrando por fecha, usando el formato DD/MM/YYYY para encontrar los resultados correctos.

Hashtags	
Fecha de inicio:	Fecha de fin:
01/10/2023	20/10/2023
<div>Buscar Restablecer</div>	
Fecha	Hashtag
20/10/2023	#bienvenida: 1 mensaje(s)
02/10/2023	#fortnite: 2 mensaje(s) #win: 1 mensaje(s) #usac: 10 mensaje(s) #feliz: 10 mensaje(s)
02/10/2023	#friend: 1 mensaje(s) #usac: 11 mensaje(s) #feliz: 11 mensaje(s)

Figura 10. Consultar Hashtags Filtrados por Fecha.

Fuente: elaboración propia

Al seleccionar Consultar Mencionados, nos mostrara una página web en la cual veremos todos los usuarios cargados en los mensajes, ordenados por fecha. Esto haciendo uso de nuestra API, de un método GET, get\_mencionados el cual obtiene los usuarios mencionados y la cantidad de veces que fueron mencionados en los mensajes.

Menciones de Usuarios	
Fecha de inicio:	Fecha de fin:
dd/mm/yyyy	dd/mm/yyyy
<div>Buscar Restablecer</div>	
Fecha	Usuarios Mencionados
20/10/2023	edward_95: 1 mensaje(s) estudiante01: 1 mensaje(s) estudiante02: 1 mensaje(s)
25/10/2023	jro_99: 1 mensaje(s) estudiante01: 1 mensaje(s) estudiante02: 1 mensaje(s) estudiante03: 1 mensaje(s)
30/10/2023	estudiante03: 1 mensaje(s) estudiante04: 1 mensaje(s) jro_99: 1 mensaje(s)
05/11/2023	estudiante04: 1 mensaje(s) usuario01: 1 mensaje(s)
10/11/2023	estudiante04: 1 mensaje(s) usuario01: 1 mensaje(s)
01/11/2023	estudiante01: 1 mensaje(s) estudiante02: 1 mensaje(s)

Figura 11. Consultar Usuarios Mencionados.

Fuente: elaboración propia

En esta consulta se puede hacer un filtrado por fecha, usando el formato DD/MM/YYYY para encontrar los resultados correctos.

Menciones de Usuarios	
Fecha de inicio:	Fecha de fin:
01/10/2023	20/10/2023
<div>Buscar Restablecer</div>	
Fecha	Usuarios Mencionados
20/10/2023	edward_95: 1 mensaje(s) estudiante01: 1 mensaje(s) estudiante02: 1 mensaje(s)
02/10/2023	edgorn95: 2 mensaje(s) jro_99: 2 mensaje(s)

Figura 12. Consultar Usuarios Mencionados Filtrados por Fecha.

Fuente: elaboración propia

Al seleccionar Consultar Sentimientos, nos mostrara una página web en la cual veremos como se clasificaron los mensajes según el contenido, se comparan según las palabras claves cargadas en config.xml y se hace una clasificación de mensajes positivos, negativos y neutros.

Esto se hace desde nuestra API, haciendo uso del método, get\_messages el cual nos obtiene el mensaje y recorre el mensaje para comprobar si las palabras coinciden con alguna de la configuración, en caso encontrar palaras las palabras claves, lo almacenará en positivo o negativo, si no se encuentra ninguna palabra en el mensaje que coincida con la configuración se dará como mensaje neutro.

Mensajes Sentimiento

Fecha de inicio:

dd/mm/yyyy

Fecha de fin:

dd/mm/yyyy

Buscar

Restablecer

Fecha	Mensajes Positivos	Mensajes Negativos	Mensajes Neutros
20/10/2023	1	0	0
Bienvenido @estudiante01 @estudiante02, el @edward_95 esperamos que esten satisfecho en esta institucion #bienvenida#			
25/10/2023	1	0	0
Hola @estudiante01 hoy @estudiante02 no podra ir ya que esta malo, @estudiante03 deberias juntarte con @jro99 es un excelente companero #bienvenida.iceo#			
30/10/2023	0	1	0
@estudiante03 Hey @estudiante04, es triste que @jro_99 no pudo entrar a la facultad #usac# #feliz#			
05/11/2023	1	0	0
hoy @usuario01 y @estudiante04, veran su nota de parcial y todo parece que les ira cool #usac# #feliz# #SIP# #SALE# #IPC2#			

Figura 13. Consultar Usuarios Mencionados Filtrados por Fecha.

Fuente: elaboración propia

En caso seleccionar la opción Graficas, se nos desplegara una lista de las gráficas, las cuales son: Grafica Hashtags, Grafica de usuarios Menciones, Grafica del Sentimientos de los Mensajes. Esto mostrando una página web para cada uno, haciendo uso de los métodos GET de nuestra API, para así extraer la información de nuestros documentos bd.xml y config.xml

Al seleccionar Grafica Hashtags, nos mostrara una página web en la cual veremos un gráfico con los hashtags cargados en los mensajes, mostrando en qué fecha se usaron más hashtags y cuantas veces fueron usados. Esto haciendo uso de nuestra API, de un método GET, stats\_hashtags el cual obtiene los hashtags y la cantidad de veces que fueron usados en los mensajes.

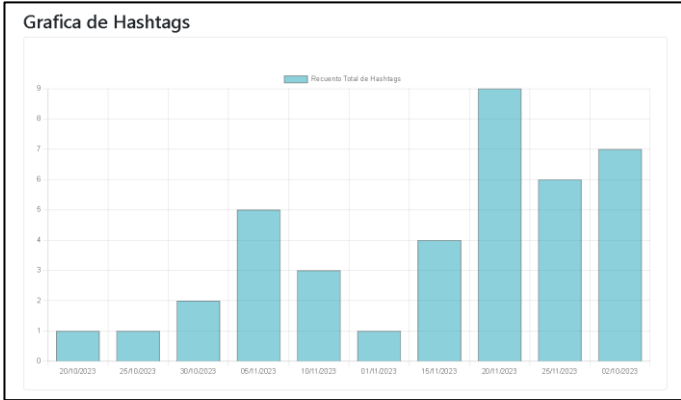


Figura 14. Grafica Hashtags  
Fuente: elaboración propia

Así mismo nos mostrara una tabla en la cual se visualiza el hashtag y cuantas veces fue usado y en qué fecha.

Hashtags Utilizados en Cada Mensaje	
Fecha	Hashtags
	<ul style="list-style-type: none"><li>#bienvenida# (1 veces)</li></ul>
	<ul style="list-style-type: none"><li>#bienvenidaloro# (1 veces)</li></ul>
	<ul style="list-style-type: none"><li>#usac# (1 veces)</li><li>#feliz# (1 veces)</li></ul>
	<ul style="list-style-type: none"><li>#usac# (1 veces)</li><li>#si# (1 veces)</li><li>#feliz# (1 veces)</li><li>#ip2# (1 veces)</li><li>#sala# (1 veces)</li></ul>
	<ul style="list-style-type: none"><li>#descanso# (1 veces)</li><li>#feliz# (1 veces)</li><li>#vacaciones# (1 veces)</li></ul>

Figura 15. Hashtags Utilizados en cada Mensaje.  
Fuente: elaboración propia

Al seleccionar Grafica Mencionados, nos mostrara una página web en la cual veremos un gráfico con los usuarios mencionados cargados en los mensajes, mostrando en qué fecha se mencionados más usuarios y cuantas veces fueron mencionados. Esto haciendo uso de nuestra API, de un método GET, stats\_menciones el cual obtiene los usuarios mencionados y la cantidad de veces que fueron mencionados en los mensajes.

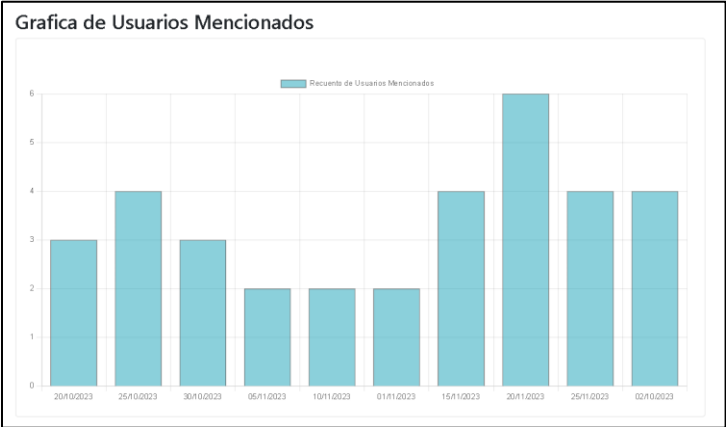


Figura 16. Consultar Usuarios Mencionados Filtrados por Fecha.  
Fuente: elaboración propia

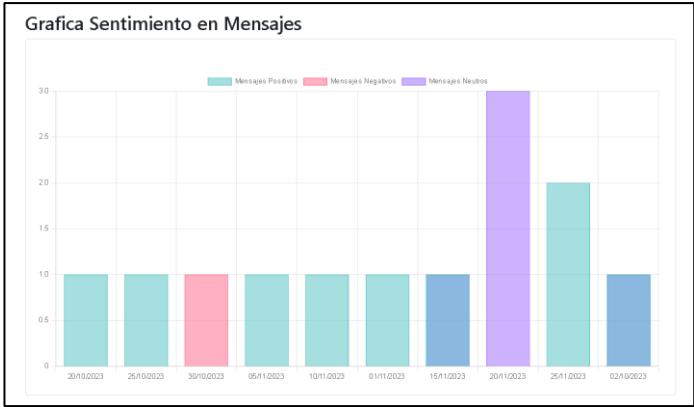
Así mismo nos mostrara una tabla en la cual se visualiza los usuarios y cuantas veces fue usado y en qué fecha.

Usuarios Mencionados en Cada Mensaje

Fecha	Usuario
	<ul style="list-style-type: none"><li>edward_95 (1 veces)</li><li>estudiante01 (1 veces)</li><li>estudiante02 (1 veces)</li></ul>
	<ul style="list-style-type: none"><li>Jiro99 (1 veces)</li><li>estudiante01 (1 veces)</li><li>estudiante02 (1 veces)</li><li>estudiante03 (1 veces)</li></ul>
	<ul style="list-style-type: none"><li>estudiante03 (1 veces)</li><li>estudiante04 (1 veces)</li><li>jiro_99 (1 veces)</li></ul>
	<ul style="list-style-type: none"><li>estudiante04 (1 veces)</li><li>usuario01 (1 veces)</li></ul>

Figura 17. Usuarios Mencionados en Cada Mensaje.  
Fuente: elaboración propia

Al seleccionar Grafica Sentimientos, nos mostrara una página web en la cual veremos un gráfico con los sentimientos de cada mensaje, este clasificado por mensajes positivos, negativos, neutros , mostrando en qué fecha se hay mensajes con sentimientos. Esto haciendo uso de nuestra API, de un método GET, stats\_sentimientos el cual obtiene los mensajes y los clasifica si son positivos, negativos o neutros.



## Conclusiones

Este ensayo presenta una aplicación de análisis de contenido en de una red sociales en el que se combina un Frontend en Django y un Backend en Flask. La aplicación ofrece diversas funciones, como carga de mensajes, configuración, consultas y generación de gráficos.

La interfaz gráfica de la aplicación es amigable y accesible, y facilita la interacción con las funciones del sistema. La aplicación también tiene la capacidad de analizar el sentimiento de los mensajes, lo que proporciona información valiosa para la toma de decisiones y la gestión de la reputación en línea.

Las consultas ofrecidas por la aplicación permiten a los usuarios comprender mejor las tendencias y la participación en las redes sociales. La generación de gráficos complementa esta funcionalidad al ofrecer representaciones visuales claras de los datos recopilados.

La aplicación también ofrece la opción de reiniciar datos, lo que permite a los usuarios realizar análisis frescos sin interrupciones.

En general, esta aplicación ofrece una solución completa y versátil para comprender el comportamiento de las redes sociales. Sin embargo, es importante considerar los aspectos éticos y de privacidad de la recopilación y análisis de datos en redes sociales.

## Referencias bibliográficas

C. J. Date, (1991). *An introduction to Database Systems*.

django. (02 de 11 de 2023). *django*. Obtenido de django:

<https://docs.djangoproject.com/es/4.2/>

Flask. (02 de 11 de 2023). *Flask*. Obtenido de Flask:

<https://flask.palletsprojects.com/en/3.0.x/>

w3schools. (02 de 11 de 2023). *w3schools*. Obtenido de w3schools:

<https://www.w3schools.com/html/default.asp>

w3schools. (02 de 11 de 2023). *w3schools*. Obtenido de w3schools:

<https://www.w3schools.com/css/default.asp>