
PROYECTO-3

202307705 – Josué David Velásquez Ixchop

Resumen

El proyecto 3 de ipc2 presentado consiste en desarrollar una API en Python con Flask que clasifique el sentimiento de mensajes obtenidos de redes sociales, determinando si estos son positivos, negativos o neutros. Utilizando un diccionario de palabras de sentimientos y considerando nombres de empresas y servicios específicos, la herramienta busca analizar la percepción de los usuarios sobre dichas entidades. La API recibe archivos XML como entrada y genera respuestas también en formato XML, proporcionando estadísticas detalladas de los mensajes clasificados por fecha, empresa y servicio. Este sistema tiene impactos técnicos y sociales significativos, ya que facilita el análisis de opiniones en redes sociales, útil para empresas al monitorear su reputación. Entre las conclusiones, se destaca la importancia de la programación orientada a objetos y la organización en componentes de backend y frontend, apoyados por frameworks como Flask y Django para un mejor manejo y visualización de datos en el proyecto.

Palabras clave

1. API
2. XML
3. Sentimiento
4. Análisis empresarial
5. análisis social

Abstract

The ipc2 project 3 presented consists of developing an API in Python with Flask that classifies the sentiment of messages obtained from social networks, determining whether they are positive, negative or neutral. Using a dictionary of sentiment words and considering names of specific companies and services, the tool seeks to analyze users' perception of said entities. The API receives XML files as input and generates responses also in XML format, providing detailed statistics of messages classified by date, company and service. This system has significant technical and social impacts, since it facilitates the analysis of opinions on social networks, useful for companies when monitoring their reputation. Among the conclusions, the importance of object-oriented programming and organization in backend and frontend components stands out, supported by frameworks such as Flask and Django for better management and visualization of data in the project.

Keywords

- 1.API
- 2.XML
3. Feeling
4. Business analysis
5. social analysis

Introducción

Este proyecto se centra en el desarrollo de una API en Python para analizar y clasificar sentimientos en mensajes de redes sociales, identificando si son positivos, negativos o neutros. En un contexto donde las opiniones de los usuarios en línea influyen en la reputación y el posicionamiento de las empresas, esta herramienta se vuelve esencial. Basada en técnicas de procesamiento de lenguaje natural y organizada en una arquitectura cliente-servidor, la API procesa archivos XML para extraer datos y generar reportes, permitiendo que las empresas comprendan mejor la percepción del público sobre sus servicios. El proyecto plantea la interrogante: ¿cómo se puede automatizar y optimizar el análisis de sentimiento en redes sociales para ofrecer resultados precisos? A lo largo del ensayo, se exploran los fundamentos teóricos y técnicos de este desarrollo, evidenciando el impacto de tales herramientas en la gestión de la reputación empresarial y la toma de decisiones estratégicas.

Desarrollo del tema

En el desarrollo de esta API, se buscó diseñar un sistema automatizado para el análisis de sentimientos en mensajes de redes sociales, con el fin de identificar la percepción pública hacia empresas y sus servicios. La solución utiliza Python y Flask para el backend, y sigue una estructura modular que integra varias funciones para la carga, procesamiento y clasificación de mensajes a partir de archivos XML. A continuación, se describe en detalle el funcionamiento de cada módulo y su relevancia en el proceso de análisis de sentimientos.

Procesamiento y Normalización de Datos

La primera etapa es el procesamiento del diccionario de palabras de sentimientos y la normalización de texto. En esta API, los sentimientos se organizan en dos listas globales: `sentimientos_positivos_global` y `sentimientos_negativos_global`. Estas listas se generan a partir de un archivo XML que contiene palabras clave positivas y negativas, almacenadas en la estructura `<diccionario>`. Para mantener la consistencia en el análisis, la función `normalizar`

elimina acentos y convierte el texto a minúsculas, asegurando que palabras como “bueno” y “Bueno” se traten de forma idéntica (Jurafsky & Martin, 2020).

Clasificación de Mensajes

Una vez procesado el diccionario, la API utiliza la función `clasificar_mensaje` para determinar el sentimiento de cada mensaje en función de la presencia de palabras positivas y negativas. Esta función descompone el mensaje en palabras individuales y cuenta cuántas de ellas pertenecen a cada lista de sentimientos. Si hay más palabras positivas que negativas, el mensaje se clasifica como "positivo"; si ocurre lo contrario, se clasifica como "negativo"; y si no contiene palabras de los diccionarios o hay un equilibrio, se clasifica como "neutro". Este método, conocido como "bag of words", es común en sistemas de análisis de sentimientos para grandes volúmenes de texto, aunque sus limitaciones pueden ser compensadas con un diccionario amplio y contextualizado (Parsons, 2019).

Agrupación de Mensajes por Fecha y Empresa

La función `agrupar_por_fecha` organiza los mensajes según su fecha y busca menciones específicas de empresas y servicios. Esta función es crítica para el análisis detallado de la percepción, ya que permite clasificar los datos de acuerdo con parámetros específicos de tiempo y organización, ofreciendo así una visión detallada de cómo los usuarios perciben ciertos servicios en fechas determinadas. Además, se identifican alias o sinónimos en los nombres de los servicios, lo cual mejora la precisión del análisis al incluir variaciones en el lenguaje cotidiano de los usuarios.

Cada mensaje procesado se integra en un diccionario estructurado que guarda el total de menciones y clasificaciones por sentimiento (positivo, negativo o neutro) tanto para la empresa como para sus servicios. Esto es fundamental, ya que permite analizar la percepción de la empresa a nivel general y al mismo tiempo a nivel granular en sus servicios, ofreciendo información estratégica para decisiones

empresariales basadas en la opinión pública (Turney, 2002).

Generación de la Respuesta en XML

Para cumplir con el objetivo de almacenamiento y visualización de los resultados, la API genera una respuesta estructurada en XML a través de la función `crear_xml_respuesta`. Esta función organiza la salida de acuerdo con los datos agrupados, estructurando el XML en una jerarquía que facilita la consulta y la presentación de los resultados. Cada respuesta XML incluye una sección para cada fecha analizada, con subsecciones que detallan el total de mensajes y su clasificación por empresa y servicio. Esto no solo permite una fácil visualización de los resultados, sino que además facilita la integración de estos datos en otros sistemas de reporte.

Aplicaciones y Beneficios de la Solución

El impacto técnico y empresarial de esta API es amplio. Permite a las organizaciones monitorear de manera eficiente la percepción de sus servicios y su reputación en redes sociales, proporcionándoles información clave para la toma de decisiones. Esta herramienta de análisis puede ser utilizada en una variedad de aplicaciones, como la gestión de la reputación de marca, el análisis de la satisfacción del cliente y el seguimiento de la competencia. Además, el uso de XML permite una fácil interoperabilidad con otros sistemas y garantiza que los datos puedan ser utilizados en múltiples contextos, lo que contribuye a la escalabilidad y flexibilidad de la solución.

Conclusiones

Las principales conclusiones de este proyecto reflejan el potencial del análisis automatizado de sentimientos en redes sociales como herramienta estratégica para las empresas. Esta API, desarrollada en Python y Flask, permite un procesamiento preciso y estructurado de los mensajes de los usuarios, proporcionando a las empresas datos clave para comprender mejor su percepción pública. Al integrar un diccionario de palabras de sentimientos y una clasificación por empresa y servicio, esta herramienta

facilita el monitoreo de la reputación en tiempo real y en diversos contextos.

Un aporte importante es la capacidad de la API para ofrecer un análisis detallado de cada mensaje, segmentado por fecha, empresa y servicio, lo cual es fundamental para identificar áreas específicas de mejora en las operaciones o el servicio al cliente. Sin embargo, surge la reflexión de si este enfoque basado en palabras clave es suficiente para capturar el contexto completo del mensaje o si podrían integrarse técnicas de análisis más avanzadas, como modelos de aprendizaje profundo para captar mayor complejidad emocional.

A futuro, sería recomendable expandir esta API para manejar otros idiomas y mejorar la detección de ironía o sarcasmo, características complejas en el análisis de texto que aún presentan desafíos en el ámbito de la inteligencia artificial. Este proyecto abre la puerta a futuras investigaciones y mejoras en la precisión del análisis de sentimientos, con miras a implementar soluciones aún más adaptadas y contextualizadas.

Referencias bibliográficas

- Date, C. J. (1991). *An Introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.
- Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing* (3rd ed.). Pearson.
- Parsons, T. (2019). *Practical Data Wrangling: Building and Testing Text Processing Pipelines*. O'Reilly Media.
- Turney, P. D. (2002). *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.

Apendices

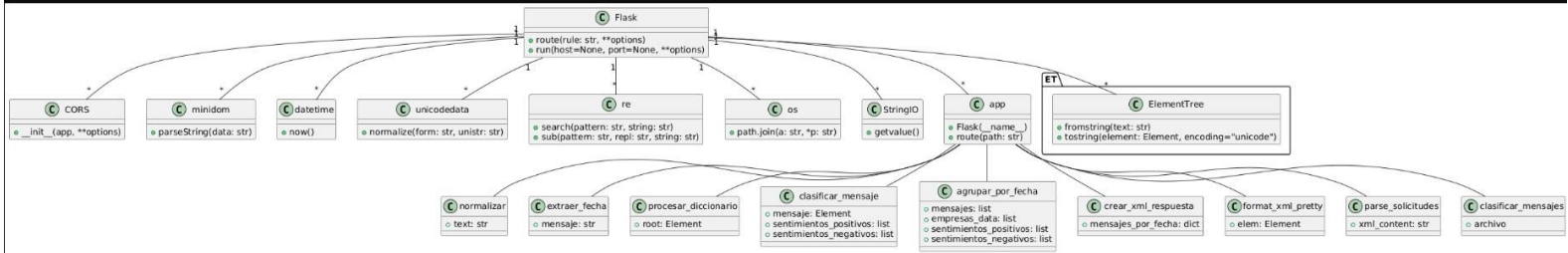


Figura 1. Diagrama de clases

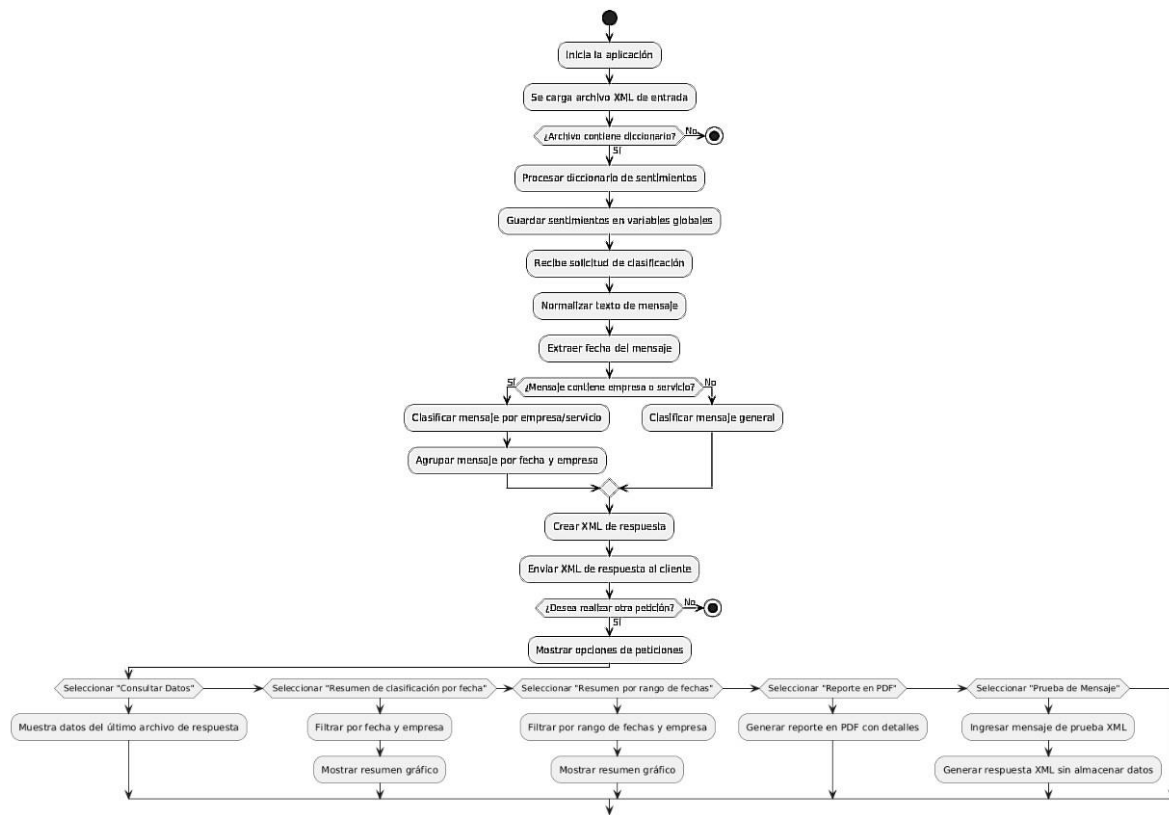


Figura 2. Diagrama de actividades

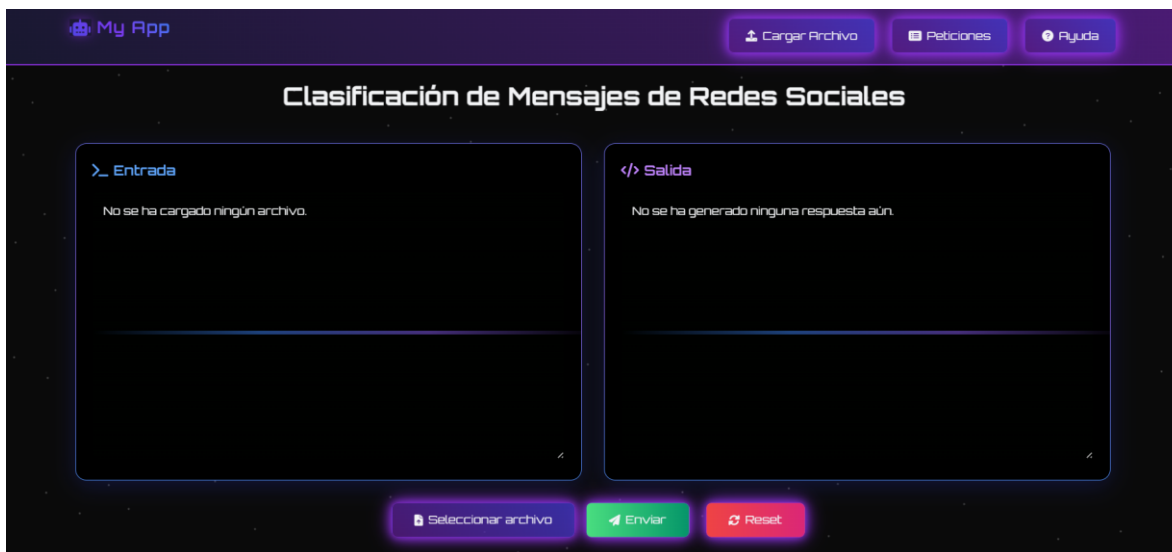


Figura 3. Clasificación de mensajes



Figura 4. Selección y envío de archivo de entra y recepción de salida



Figura 5. Reset de base de datos

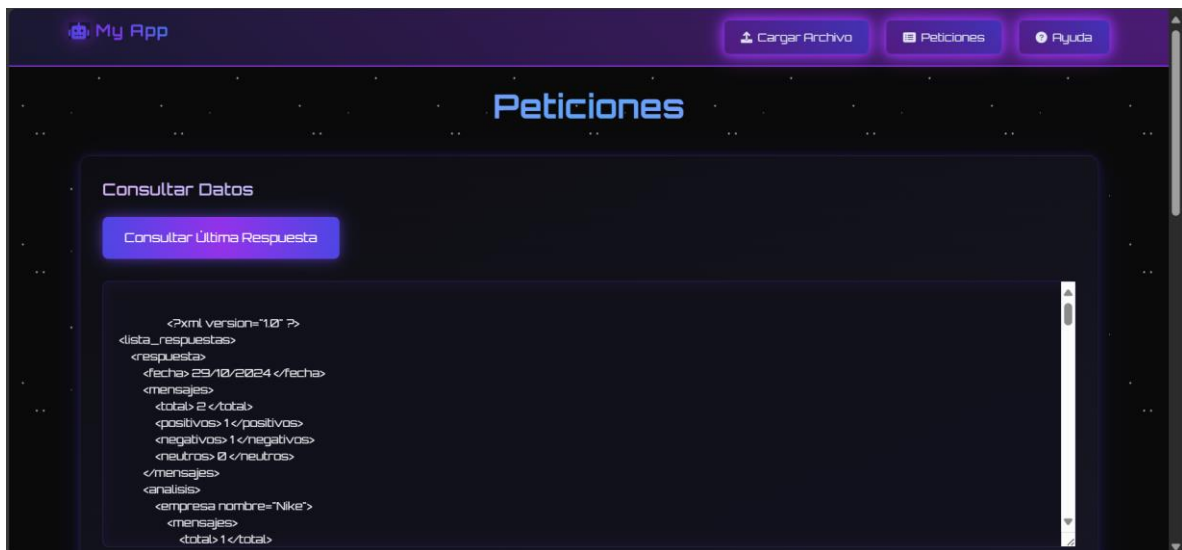


Figura 6. Consulta de última respuesta

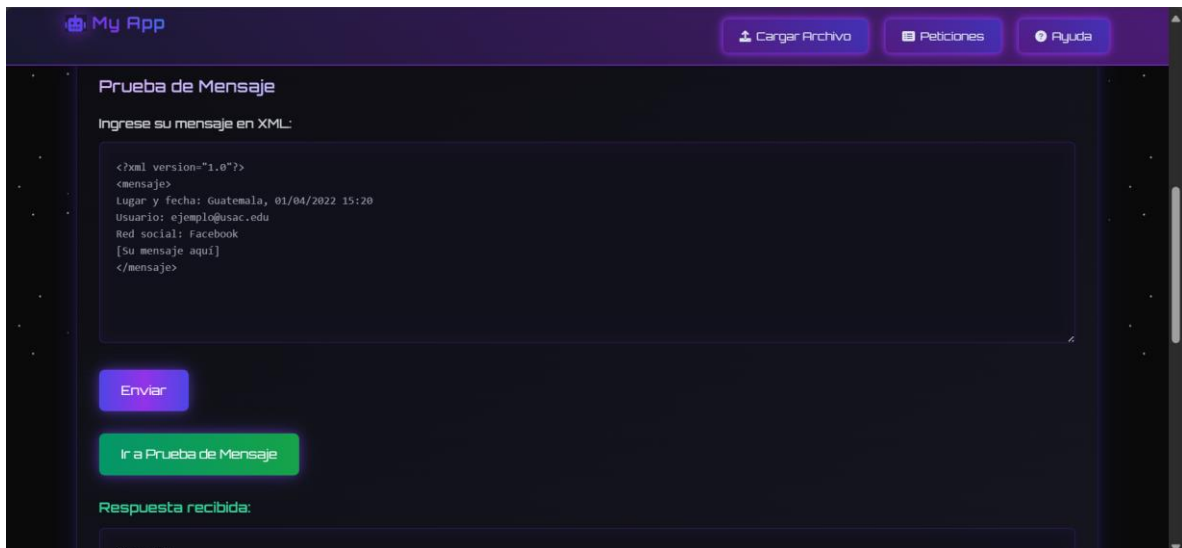


Figura 7. Prueba de mensaje individual

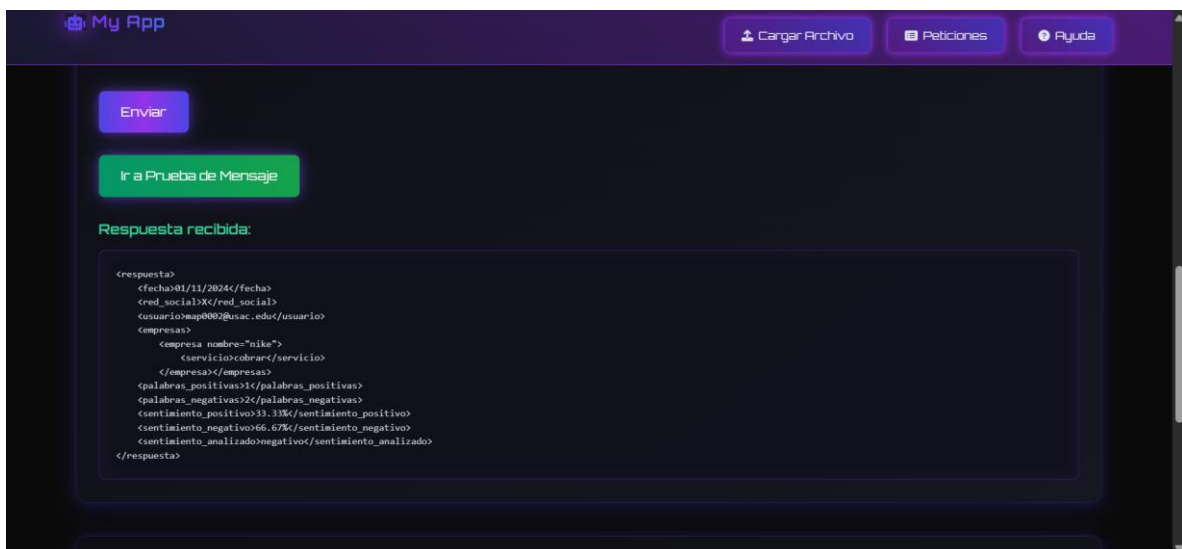


Figura 8. Respuesta de mensaje individual recibida

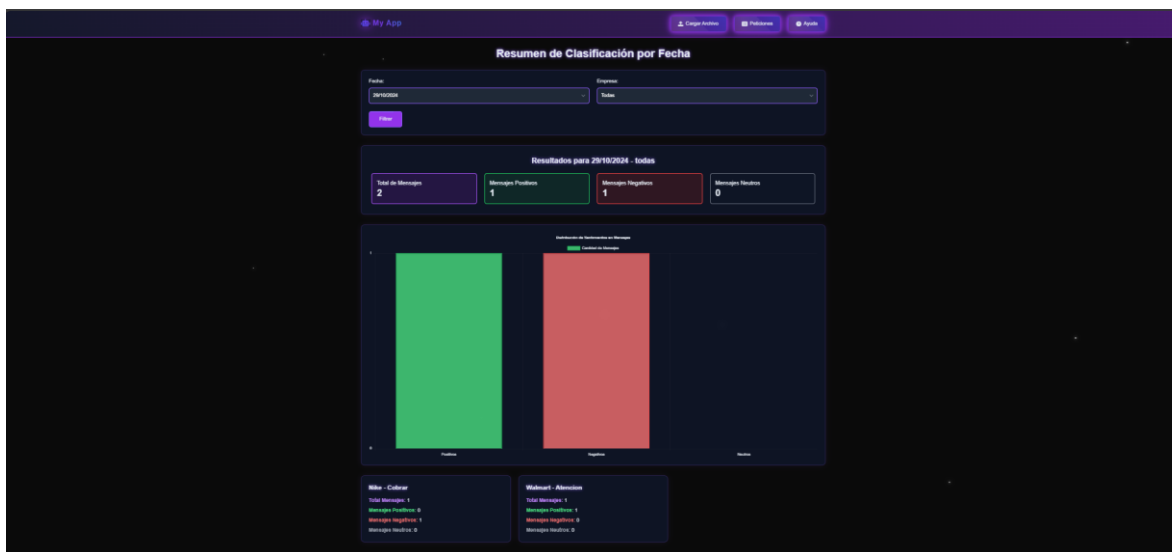


Figura 9. Resumen de clasificación por fecha

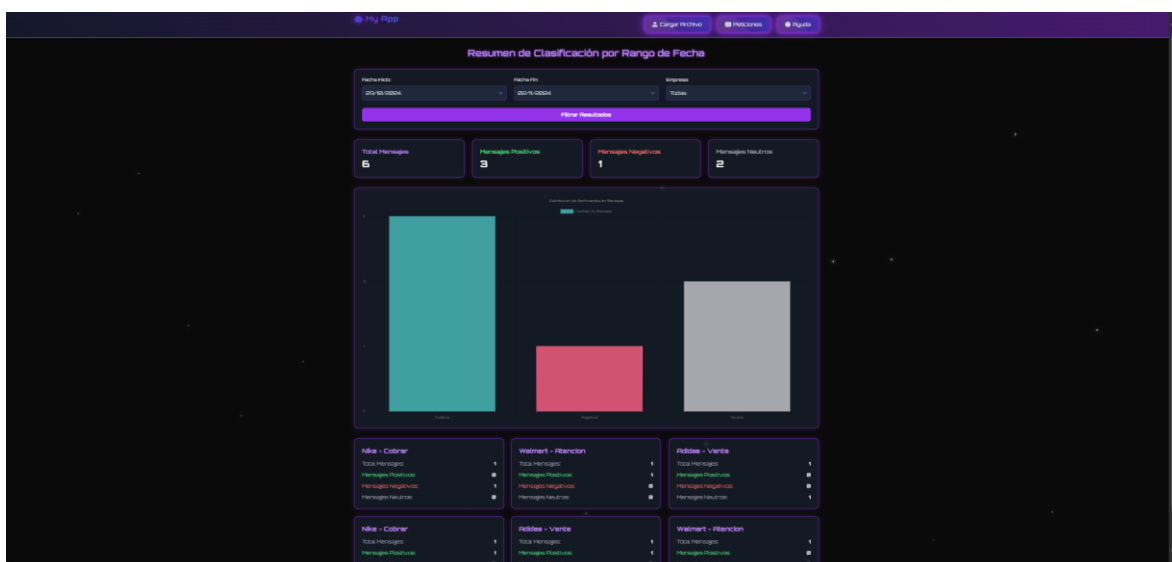


Figura 10. Resumen de clasificación por rango de fechas

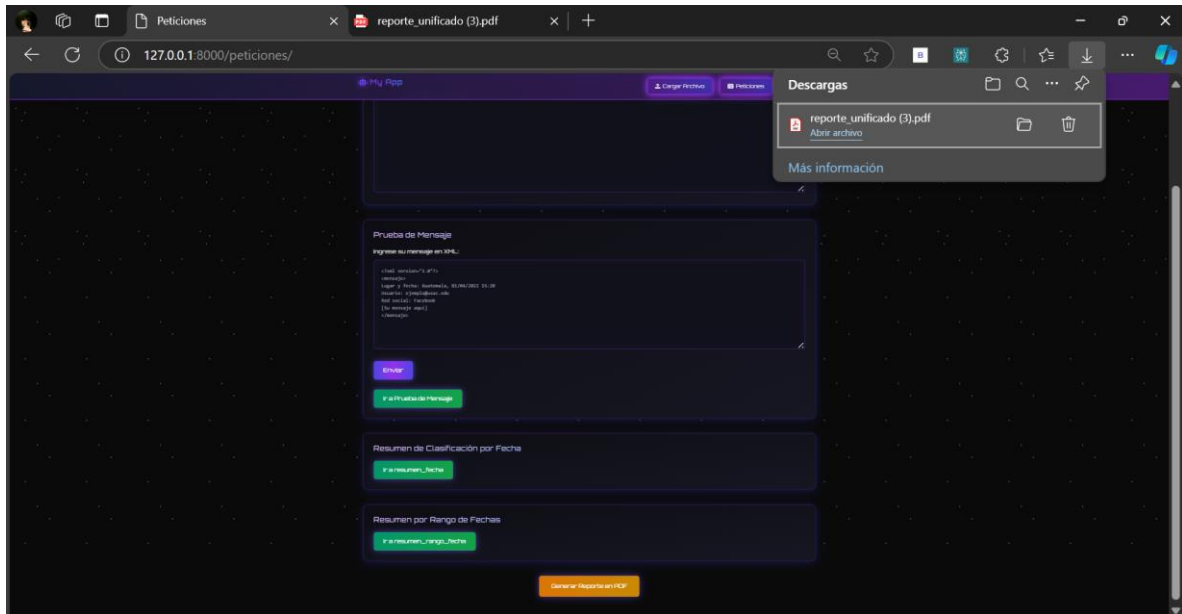


Figura 11. Generar Reporte Pdf de peticiones