
DOCUMENTACIÓN

201700900 – Gueslim Jhoans Fernandez Rivera

Resumen

Se desarrolló una aplicación con el fin de analizar ciertos mensajes de redes sociales para clasificarlo según los sentimientos del autor y para determinar la empresa y los servicios que se mencionan para conocer los sentimientos respecto a esto.

Para esta página web desarrollado por los frameworks Flask y Django. Del que Django se utilizó exclusivamente para el frontend y consumir la API desarrollada con Flask utilizando archivos .xml como base de datos.

Por lo que esta webapp garantiza analizar contenido de redes sociales y establecer el sentimiento de los usuarios respecto a una empresa y los servicios que provee.

Palabras clave

Flask, API, Django, HTTP, MVC

Abstract

An application was developed in order to analyze certain social media posts to classify it according to the author's sentiments and to determine the company and services mentioned to know the sentiments regarding this.

For this website developed by Flask and Django frameworks. From which Django was used exclusively for frontend and consume API developed with Flask using .xml files as database.

So, this webapp guarantees to analyze social network content and establish user sentiment about a company and the services it provides.

Keywords

Flask, API, Django, HTTP, MVC

Carga de XML

La configuración de las palabras a analizar y los mensajes vienen almacenados en un archivo de formato XML, por lo que el primer paso para que el programa funcione es cargar y almacenar el archivo en la base de datos.

Para esto se utilizó el módulo `xml.etree.ElementTree`, con el que se iterando para raíz del piso en que se fueron tomando los datos, de palabras positivas y negativas y las empresas y sus respectivos servicios.

Lista Enlazada

Para el manejo y el análisis de los datos se creó un tipo de dato abstracto para el almacenamiento y el procesamiento de la información obtenido del archivo XML. Para ello se eligió crear una Lista Enlazada Simple con POO. Esta lista enlazada consta de una clase `Nodo`, y una clase `LinkedList` como se aprecia en la figura 1.

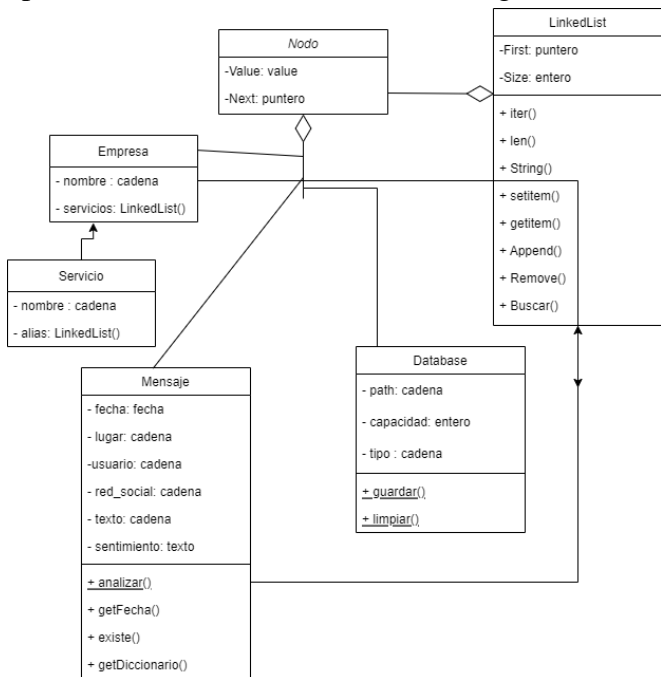


Figura 1. Diagrama de clases general.

Fuente: elaboración propia, o citar al autor, año y página.

webapp

En la carpeta `webapp` es donde esta almacenada toda la lógica del frontend, con la arquitectura modelo – vista – template. Así como va consumiendo la API del backend desarrollada con flask.

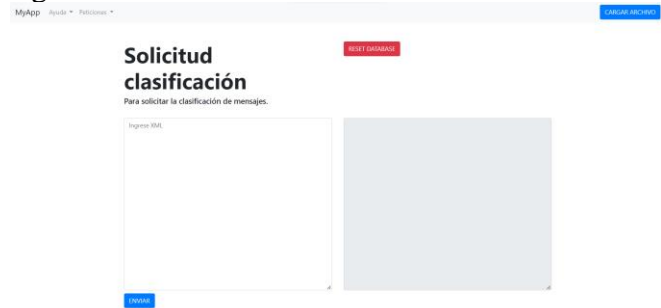
backend

En esta carpeta se ejecuta toda la lógica detrás de la app, es una API que se comunica y realiza todos los métodos a través del método `http`.

Uso

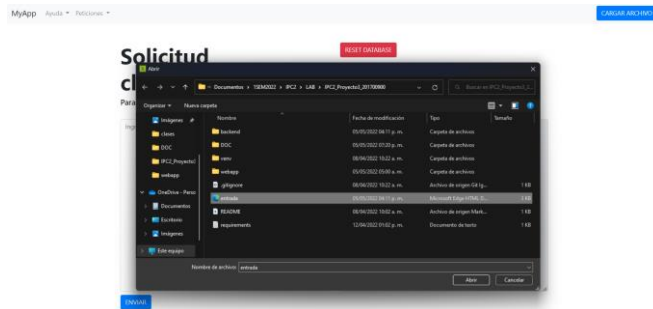
Para empezar a usar la aplicación se debe de abrir en el navegador con el `localhost` en el puerto `800`.

Al entrar al link se la pantalla de inicio será la siguiente:



En esta se encuentra un cuadro de texto donde se puede ingresar la nueva solicitud de clasificación en formato `.xml`. O también en la esquina superior derecha se encuentra un botón para hacer la carga desde el explorador de archivos. Al darle cliak se abrirá el explorador de archivos donde debemos seleccionar el `archivo.xml` a cargar.

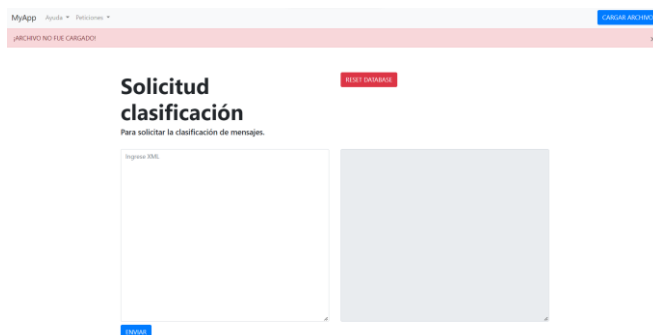
1. Cargar archivo



Al cargar el archivo o escribirlo y selecciona entrar se recibirá la respuesta de la solicitud y un mensaje de éxito.



De no haber seleccionado un archivo con el formato correcto o si el texto enviado está vacío o tampoco es del formato .xml se desplegará un mensaje de error.



2. Reset Database

Al dar click al botón reset database se ejecutara un delete de todos los datos almacenados en la base de datos se desplegará un mensaje de éxito.



3. Ayuda

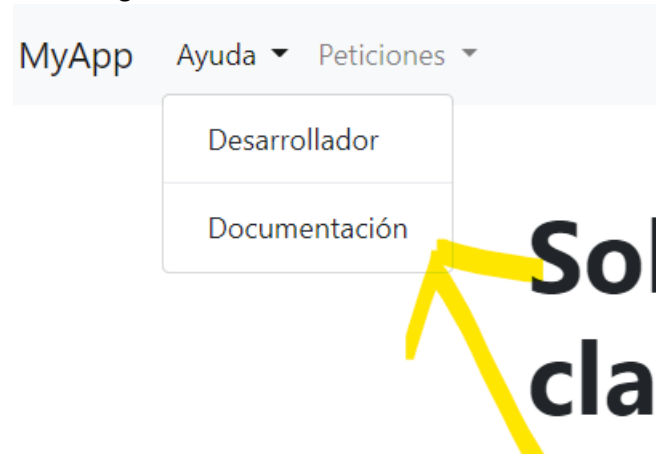
Al dar click en la parte de la barra de navegación en el botón ayuda se desplegarán dos opciones.

a. Desarrollador

Nos redirige a una página con la información del desarrollador de la aplicación.

b. Documentación

No redirige a un archivo PDF de este documento.



4. Peticiones



4.1. Reportes

Nos redirige a una página donde se muestran los repotes y los resultados de todas las solicitudes agrupados por fechas o por empresas.



4.2. Reporte PDF

Nos muestra un reporte en un documento pdf de todos los datos almacenados.

4.3. Prueba Mensaje



Nos dirige a una página con un cuadro de texto para analizar un único mensaje.

Referencias bibliográficas

Charles Severance (2015). *Python para Informaticos*. CreateSpace Independent Publishing Platform.

ANEXOS EN EL FOLDER