Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2022.

PROYECTO 3 IPC2 “ANALIZADOR DE MENSAJES”

201900042 – Rodrigo Alejandro Hernández de León

Resumen

El ensayo presenta la explicación del desarrollo de la aplicación web utilizando los frameworks de Django para la presentación de Frontend y de Flask en la parte lógica del Backend, siempre utilizando el lenguaje de programación Python para la lógica de ambos frameworks. En donde se utilizan para un analizador de mensajes donde mencionan a cada empresa y servicio calificándolo como un mensaje positivo, negativo o neutro que puede ser útil para la generación de reportes y estadísticas de cada empresa, viendo asimismo la fecha de publicación del mensaje al cual se esta analizando. Para poder realizar el análisis se necesita de un archivo XML donde será cargado y contendrá un diccionario de palabras positivas, palabras negativas y una lista de empresas a las cuales se le van a analizar sus mensajes y poder sacar sus estadísticas para poder generar una base de datos de los mensajes con un archivo XML en el backend.

Palabras clave

Frontend, Backend, Django, Flask, Aplicación web.

Abstract

The essay presents the explanation of the development of the web application using the Django frameworks for the Frontend presentation and Flask in the logical part of the Backend, always using the Python programming language for the logic of both frameworks. Where they are used for a message analyzer where they mention each company and service qualifying it as a positive, negative or neutral message that can be useful for the generation of reports and statistics of each company, also seeing the date of publication of the message which is being analyzed. In order to perform the analysis an XML file is needed where it will be loaded and will contain a dictionary of positive words, negative words and a list of companies whose messages are going to be analyzed and their statistics to be able to generate a database of messages with an XML file in the backend.

Keywords

Frontend, Backend, Django, Flask, Web Application.

Introducción

Se desarrolló una aplicación web el cual analiza mensajes de usuarios de distintas redes sociales donde se mencionan a empresas junto con sus servicios en el cual esas empresas entran dentro de una lista de empresas por analizar y un conjunto de palabras positivas y negativas para calificar y categorizar los mensajes por positivos, negativos y neutros; así mismo también categorizarlo por su fecha y la mención a su servicio. Al hacer su análisis se guarda en un archivo XML donde se encuentra una base de datos de todos los mensajes ingresados al sistema y también con un archivo XML donde se encuentra el resultado del ultimo archivo analizado. Luego de hacer su respectivo análisis el usuario puede ver las estadísticas categorizadas por fechas o dentro de un rango de fechas y también puede hacer pruebas con mensajes para ver las estadísticas del mismo.

Desarrollo del tema

La aplicación se conforma de dos servicios donde uno es el Frontend conformado por el framework Django y el usode Python y el otro servicio conformado por el framework Flask para poder procesar la parte lógica del Backend y el procesamiento de datos, siguiendo la siguiente arquitectura.

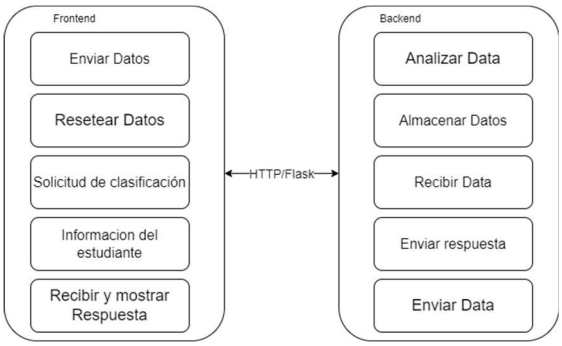

Figura 1. Arquitectura de la aplicación.

Fuente: Elaboración propia, 2022.

Para la elaboración de la solución se provee un archivo extensión y estructura XML, el cual provee información del diccionario de palabras, lista de empresas a analizar y la lista de mensajes a la cual se va a analizar.

```
<?xml version="1.0"?>
<solicitud_clasificacion>
  <diccionario>
    <sentimientos_positivos>
      <palabra> bueno </palabra>
    </sentimientos_positivos>
    <sentimientos_negativos>
      <palabra> malo </palabra>
    </sentimientos_negativos>
    <empresas_analizar>
      <empresa>
        <nombre> USAC </nombre>
        <servicio nombre="inscripción">
          <alias> inscribí </alias>
        </servicio>
      </empresa>
    </empresas_analizar>
  </diccionario>
  <lista_mensajes>
    <mensaje>
      Lugar y fecha: Guatemala, 01/04/2022 15:01 Usuario:
      map0001@usac.edu   Red social: Twitter
      El servicio en la USAC para inscripción fue muy bueno y me siento muy satisfecho.
    </mensaje>
  </lista_mensajes>
</solicitud_clasificacion>
```

Figura 2. Estructura del archivo XML de entrada.

Fuente: Elaboración propia, 2022.

donde:

solicitud_clasificacion = Contiene todo el contenido que se necesita para el análisis.

diccionario = Contiene todo el listado de palabras positivas y negativas.

sentimientos_positivos = listado de palabras positivas.

sentimientos_negativos = listado de palabras negativas.

empresas_analizar = listado de empresas a las que se quieren analizar

empresa = datos de la empresa y servicios de la misma.

nombre = nombre de la empresa a analizar

servicio=nombre del servicio a analizar

alias= alias del servicio a analizar

lista_mensajes = lista de mensajes a analizar

mensaje = contiene el mensaje al cual se va a analizar.

Este archivo se procesa y se obtienen los datos para poder hacer un nuevo archivo de salida con los mensajes analizados:

```
<?xml version="1.0"?>
<lista_respuestas>
  <respuesta>
    <fecha> 01/04/2022 </fecha>
    <mensajes>
      <total> 20 </total>
      <positivos> 8 </positivos>
      <negativos> 7 </negativos>
      <neutros> 5 </neutros>
    </mensajes>
    <analisis>
      <empresa nombre="USAC">
        <mensajes>
          <total> 3 </total>
          <positivos> 1 </positivos>
          <negativos> 1 </negativos>
          <neutros> 1 </neutros>
        </mensajes>
          <servicios>
            <servicio nombre="inscripción">
              <mensajes>
                <total> 3 </total>
                <positivos> 1 </positivos>
                <negativos> 1 </negativos>
                <neutros> 1 </neutros>
              </mensajes>
            </servicio>
            ...
          </servicios>
        </empresa>
      ...
    </analisis>
  </respuesta>
  ...
</lista_respuestas>
```

Figura 3. Estructura del archivo XML de salida.

Fuente: Elaboración propia, 2022.

donde:

lista_respuestas = Contiene todo el contenido del análisis hecho del archivo de entrada.

respuesta = Contiene todo el análisis de los mensajes por cada fecha.

fecha = contiene la fecha a la cual se analiza.

mensajes= muestra todo el contenido del análisis de los mensajes de esa fecha.

total= cantidad de mensajes en esa fecha.

positivos=cantidad de mensajes positivos en esa fecha.

negativos= cantidad de mensajes negativos en esa fecha.

neutros= cantidad de mensajes neutros en esa fecha.

analisis= muestra el análisis de las empresas en esa fecha.

empresa = nombre de la empresa analizada.

servicio = nombre del servicio analizado.

También en el proceso de hacer mensajes de prueba se necesita del siguiente texto de entrada:

```
<?xml version="1.0"?>
<mensaje>
Lugar y fecha: Guatemala,
 01/04/2022 15:20 Usuario: map0002@usac.edu Red social: Facebook Hoy
me inscribí en la USAC, no encontré parqueo e inicié molesto mi asignacion,
luego tuve que hacer cola y me indicaron que era la cola incorrecta,
esto me enojó mucho y mejor me fui.
</mensaje>
```

Figura 4. Estructura del mensaje XML de entrada.

Fuente: Elaboración propia, 2022.

El mensaje se procesa y genera la siguiente respuesta de salida:

```
<respuesta>
    <fecha> 01/04/2022 </fecha>
    <red_social> Facebook </red_social>
    <usuario> map0002@usac.edu </usuario>
    <empresas>
        <empresa nombre="USAC">
            <servicio> inscripcion</servicio>
            <servicio> asignacion</servicio>
        </empresa>
    </empresas>
    <palabras_positivas> 0</palabras_positivas>
    <palabras_negativas>2</palabras_negativas>
    <sentimiento_positivo> 0% </sentimiento_positivo>
    <sentimiento_negativo> 100% </sentimiento_negativo>
    <sentimiento_analizado> negativo </sentimiento_analizado>
</respuesta>
```

Figura 5. Estructura del análisis XML de respuesta.

Fuente: Elaboración propia, 2022.

Para poder realizar los procesos de análisis de los archivos de entrada se utilizaron los siguientes algoritmos:

a. Analizar Datos: Aquí se utiliza la petición HTTP: “POST” para obtener el archivo XML, leerlo y mandar su contenido a una clase llamada Analizador donde comienza utilizando una librería llamada Element Tree para leer el contenido XML donde la clase crea listas globales para guardar la información de los diccionarios y empresas a analizar. También se crean listas temporales para obtener un archivo de salida del último archivo analizado. Luego al obtener las listas de palabras positivas, negativas y de empresas entonces, se obtenía el mensaje y se analizaba por aparte con un método creado para el análisis del mensaje. Para el método se tuvo

que utilizar un Split en el mensaje y quitar las comas, puntos y tildes, y poner todas las palabras minúsculas para tener un análisis exitoso del mensaje. Luego de hacer un split generando el arreglo de palabras del mensaje se utilizo un ciclo While para poder obtener los siguientes datos: Fecha y lugar donde en este mismo se utilizo una Expresion Regular para obtener la fecha y crear un objeto de tipo mensajes, Usuario donde por medio de una expresión regular se obtenia el username y la Red Social donde también se obtiene por medio de la verificación de una expresión regular para que tenga los primeros datos del mensaje. Luego se separa su análisis con lo demás del contenido del mensaje donde palabra por palabra comprueba si esa palabra se encuentra dentro de la lista de palabras positivas e incrementarlo con un contador de palabras positivas o de lo contrario si encuentra dentro del listado de palabras negativas incremente un contador de palabras negativas. Luego verifica si el mensaje menciona a una empresa o un servicio de la empresa y lo guarda dentro de un diccionario que cuenta cuantas veces ha sido mencionado. Luego verifica si el contador de palabras positivas tiene la misma cantidad de palabras negativas entonces dentro del atributo de mensajes neutros del objeto mensaje aumenta su contador a 1. Si el contador de palabras positivas es mayor al contador de palabras negativas entonces dentro del atributo de mensajes positivos del objeto mensaje aumenta su contador a 1. De lo contrario el contador de mensajes negativos aumenta 1. Luego crea un objeto de tipo empresa y crea sus servicios junto con la fecha analizada anteriormente y hace el mismo proceso de ver si es mensaje positivo, negativo o neutro.

b. Crear XML de salida: Obtiene de las listas globales los datos para generarla estructura en XML de los datos de las listas globales y genera un archivo XML llamado Respuestas.XML donde guarda los análisis de todos los archivos de entrada y poder retornar el XML de salida.
c. Analizar Archivo Temporal: Realiza el mismo análisis del algoritmo de Analizar Datos pero con la diferencia de que usa listas temporales para poder generar un XML de salida del ultimo archivo de entrada analizado y guardándolo como Respuesta.XML para retornarlo en el Frontend y mostrarlo al consultar datos.
d. Analizar Mensaje de Prueba: Realiza el mismo análisis que en el algoritmo de mensaje de prueba con la diferencia de no guarda en la lista global del analizador de mensajes pero haciendo el análisis de palabras positivas o negativas con el uso de la lista global y generando un XML con el nombre de Respuesta.XML para poder retornar el contenido del mismo como salida del análisis del mensaje de prueba.

Para poder hacer las peticiones a la API se crearon los siguientes end-points:

a. ‘/ConsultarDatos’[POST]: Ingresa en el body el archivo de entrada y retorna un archivo de salida de la base de datos.
b. ‘/ConsultarDatos’[GET]:Obtiene el contenido del archivo de salida del análisis del ultimo archivo de entrada.

c. ‘/ProcesarMensaje’[POST]: Se envia un mensaje de prueba en el body y recibe el contenido del archivo de salida de respuesta.
d. ‘/Fechas’[GET]: Se obtienen todas las fechas almacenadas en el sistema.
e. ‘/Empresas’[GET]: Se obtienen todas las empresas almacenadas en el sistema.
f. ‘/ConsultaFecha’[POST]:Obtiene en el body una fecha y empresa o únicamente una fecha para que retorne el nombre de la empresa y las estadísticas de los mensajes.
g. ‘/reset’[DELETE]: Borra todo de los archivos de las bases de datos y borra todo el contenido de las listas de la clase Analizador.
h. ‘/ConsulaRangoFechas’[POST]: Obtiene en el body una fecha de inicio, una fecha final y como opcional el nombre de una empresa para obtener la cantidad de mensajes entre ese rango.
i. ‘/Totales’[POST]: Obtiene el rango de las fechas y retorna la cantidad de mensajes totales en esa fecha.
j. ‘/Fecha’[POST]: Obtiene la cantidad de mensajes en esa fecha.

Conclusiones

El desarrollo de esta aplicación da como ejemplo claro de como funciona un analizador de mensajes y puede dar como respuesta el sentimiento que transite cada mensaje y puede beneficiar o no a la empresa a la cual se esta analizando.

Se aprendió a utilizar frameworks como Django y Flask para poder comunicarse y poder hacer más sencilla la comunicación de sus peticiones y el uso de MVT para poder mostrar al usuario de forma amigable el resultado de las peticiones hechas.

Referencias bibliográficas

USAC, T.d.(7 de Abril de 2022). *Proyecto 3, UEDi.* Obtenido de *https://uedi.ingenieria.usac.edu.gt/campus/pluginfile.php/294136/mod_resource/content/0/%5BIPC2%5DProyeto%203.pdf?redirect=1*

Anexos

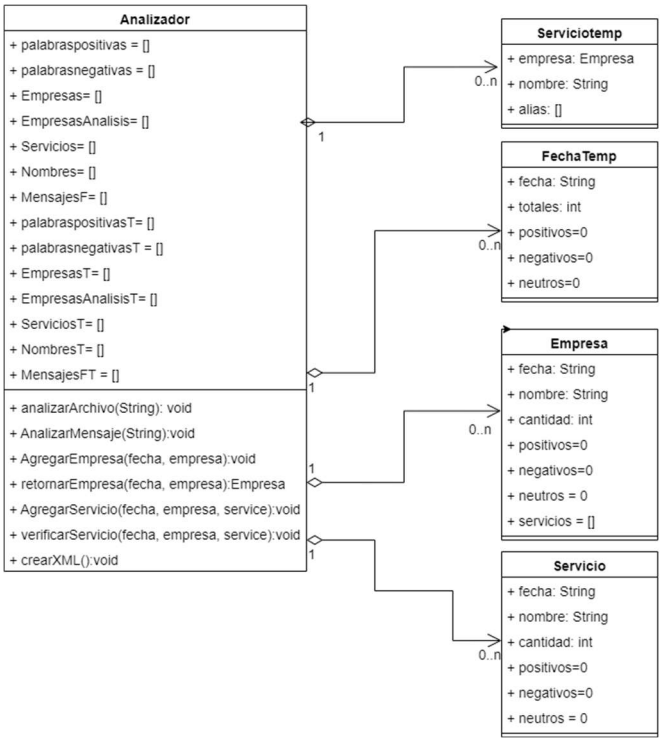

Figura 6. Diagrama de Clases del Analizador.

Fuente: Elaboración propia, 2022.