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 2023.

Proyecto 3 | Introducción a la programación y computación 2

202103105 – Luis Castellanos

Resumen

El siguiente proyecto consiste en la creación de una herramienta que sea capaz de administrar los mensaje de una red social llamada ChapinChat, la cual también se encarga de crear una base de datos en XML.

También es capaz de ver cada mensaje y ver en qué perfil, los cuales pueden ser establecidos, encaja el mensaje de dicho usuario

Palabras clave

HTTP

Red Social

Bases de datos

XML

Abstract

This project consists in the creation of a tool capable of handling messages of a social network called ChapinChat. This social network will use a database using XML

This tool is also capable of finding out in which profile a user fits, given a message by said user

Keywords

HTTPS

Social Networ

Databases

XML

Introducción

La siguiente herramienta se desarrolló para el análisis de mensajes de la red social ChapinChat. Es capaz de almacenar en bases de datos generadas en archivos xml los perfiles en los cuales un usuario puede encajar y también de almacenar los mensajes, ordenados por ubicación, fecha, usuario y el cuerpo del mensaje

Es posible solicitar a la herramienta los mensajes de cualquier usuario, además de la fecha en que se escribieron. También se pueden almacenar nuevos perfiles cuando se desee

Desarrollo del tema

Para el desarrollo de la aplicación, en específico del backend, se usó la herramienta Flask, la cual cuenta con una amplia gama de extensiones disponibles, lo que facilita la incorporación de características adicionales a la aplicación. Estas extensiones abarcan desde la autenticación y la gestión de bases de datos hasta la integración con servicios en la nube, lo que permite a los desarrolladores ampliar las capacidades de sus aplicaciones de manera rápida y sencilla.

También se usaron varios algoritmos de ordenamiento y ciertas expresiones regulares, las cuales se utilizaron para el desarrollo del calculo de probabilidad de perfil.

Estos números se usan en la aplicación para obtener el peso de un perfil en cualquier usuario

Para el modulo de flask, se utilizaron los siguientes endpoints

```
@app.route("/")
def hello_world():···

@app.route("/DatosEstudiante",methods=['GET'])
def DatosEstudiante():···

@app.route("/CrearPerfiles",methods=['POST'])
def CrearPerfiles():···

@app.route("/CrearMensajes",methods=['POST'])
def CrearMensajes():···

@app.route("/ResetearDatos",methods=['GET'])
def ResetearDatos():···

@app.route("/MensajesPorUsuario",methods=['GET'])
def MensajesPorUsuario():···
```

Figura 1. Endpoints usados para el desarrollo.

Fuente: elaboración propia

Los endpoints que aparecen en la figura uno son los que se encargan de almacenar los datos en los archivos XML

Estructura de los archivos XML

Para almacenar la información que se mandan a través de peticiones HTTP, se utilizaron 3 archivos los cuales son

Mensajes.xml

En este apartado se encuentran los mensajes que se procesan por la herramienta. Con expresiones regulares se obtiene la fecha, el usuario, la red social que usa, y el cuerpo del mensaje

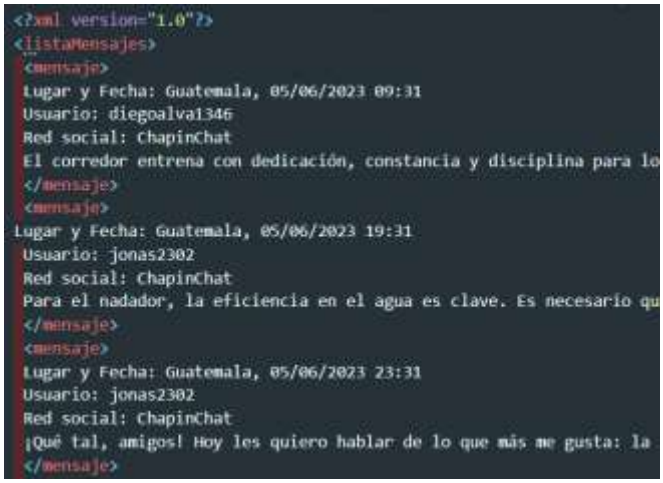

*Figura 2.*estructura del archivo XML.

Fuente: elaboración propia

Perfiles.xml

En este apartado se almacenan los perfiles. Aquí se incluyen las palabras que serán analizadas con el analizador léxico y expresiones regulares; y el nombre del perfil al cual corresponden las palabras

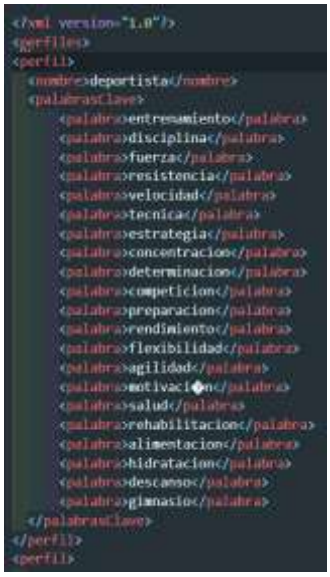

Figura 3. Estructura del archivo de perfiles

Fuente: elaboración propia

Descartadas.xml

Aquí es donde se incluyen las palabras que procederán a ser ignoradas por el analizador léxico. Estas palabras no se tomaran en cuenta durante el análisis

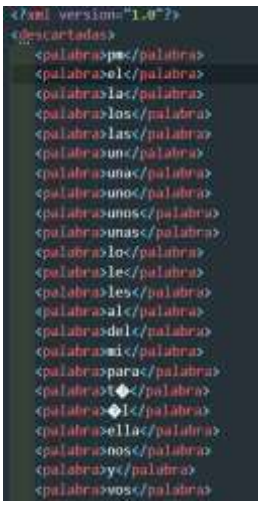

Figura 4. Estructura del archivo de palabras descartadas

Fuente: elaboración propia

Funciones.py

En este modulo es donde se encuentra toda la lógica principal para analizar las palabras y obtener la probabilidad de que un usuario encaje en un perfil, el almacenamiento de mensajes, perfiles, y palabras descartadas en sus respectivos archivos XML, y de la generación de las tablas que se muestran a continuación.

Para el manejo de archivos XML, se utilizó el módulo minidom, el cual es conveniente y accesible para trabajar con documentos XML en Python, ofreciendo una interfaz fácil de usar, compatibilidad multiplataforma y funcionalidades esenciales para el procesamiento de XML.

```
tablaUsuario(Usuario : str)
```

En este método es donde se crean las tablas las cuales analizan cada mensaje de un usuario dado, y verifica cual es la probabilidad de que encaje en un perfil de los que se encuentran en el archivo Perfiles.XML

Este método hace uso de otros dos métodos, los cuales se describen a continuación

```
obtenerProbabilidad(listaPerfil, mensaje)
```

En este método es en donde se hace uso exhaustivo del analizador léxico, el cual realiza las siguientes acciones:

- Elimina los caracteres innecesarios, como las tildes, las comas, los puntos etc.
- Transforma todas las letras a minúsculas
- Encuentra las coincidencias en la lista que se proveyó en los parámetros descritos anteriormente
- Divide el numero de coincidencias dentro del total de palabras

```
leerXmlPerfiles(xml)
```

```
leerXmlMensaje(xml)
```

Estos dos métodos son los encargados de almacenar los datos a los archivos XML, aquí es donde se usan los métodos de minidom

Respuestas a peticiones HTTP

Para la solicitud de creación de bases de datos se utilizaron las peticiones GET en donde los parámetros y datos se envían a través de la URL como parte de la cadena de consulta.

Las peticiones POST se usaron ya que los datos se envían en el cuerpo de la petición y no son visibles en la URL

Las funciones anteriores mencionadas generan respuestas para indicar que se ha realizado exitosamente el almacenamiento de los datos.

La estructura de la respuesta del primer método es la siguiente

```
<?xml version="1.0"?>
<respuesta>
    <perfilesNuevos> Se han añadido 5 perfiles </perfilesNuevos>
    <descartadas> Se han creado 67 nuevas palabras a descartar </descartadas>
</respuesta>
```

Figura 5. Estructura de la respuesta

Fuente: elaboración propia

Y la estructura de respuesta del método leerXmlMensaje es la siguiente

Figura 4. Estructura del archivo de palabras descartadas

Fuente: elaboración propia

Respuestas a peticiones HTTP

Si se desea se pueden restaurar los datos de la aplicación utilizando */ResetearDatos.*

Este método devuelve a su estado original a los archivos Descartadas.xml, Mensajes.xml y Perfiles.xml

Referencias bibliográficas

Open Flask Project. (s.f.). Flask - A Python Microframework. Recuperado el 6 de mayo de 2023, de https://flask.palletsprojects.com/

Python Software Foundation. (2021). minidom — Minimal DOM implementation. Recuperado el 6 de mayo de 2023, de https://docs.python.org/3/library/xml.dom.minidom.html