Proyecto 3 – Verificador de sentimientos en mensajes de redes Sociales

Carnet 202010816 – César André Ramírez Dávila

Resumen

El verificador de sentimientos en mensajes es una aplicación desarrollada utilizando una API en Python con la librería Flask y consumiéndola mediante un frontend implementado con el framework Django. Esto para que la aplicación pueda ser consumida a través de internet y pueda ser subida a alguna nube para su uso remoto.

Palabras clave

Django

API

Frontend

Backend

A Abstract

The message sentiment checker is an application developed using a Python API with the Flask library and consuming it through a frontend implemented with the Django framework. This is so that the application can be consumed through the internet and can be uploaded to a cloud for remote use.

Keywords

Django

API

Frontend

Backend

Introducción

El proyecto tiene un enfoque en la programación web, haciendo uso de lo que se conoce como frontend y backend. Estos dos se desarrollaron de tal forma que el backend guardara los datos y el frontend los muestra, consumiendo el backend. En este proyecto también se utilizó el framework Django para poder desarrollar el frontend.

Desarrollo del tema

Para este proyecto número tres, se tuvo que hacer uso de lo que se le conoce como

a. API

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones.

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

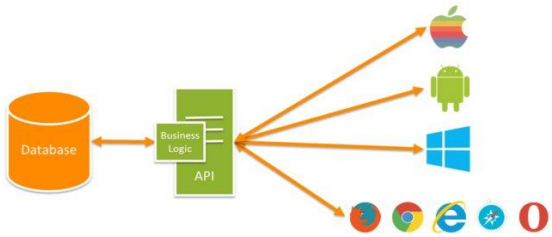

Figura 1. Representación del consumo de una API.

En este proyecto se utilizó una API Rest, la cuál recibe solicitudes del servidor y realiza diferentes funciones dependiendo de cuál se haya recibido. En este tipo de API's tenemos cuatro solicitudes principales.

a.1. Solicitud GET:

Este tipo de solicitud lee los datos en el host enviándolos para su posterior análisis.

a.2. Solicitud POST:

Este tipo de solicitud recibe información para luego crear datos pertinentes, y ser almacenados en el host.

a.3. Solicitud DELETE:

Este tipo de solicitud elimina datos que esté siendo almacenada en el host.

a.4. Solicitud PUT:

Este tipo de solicitud modifica datos que esté siendo almacenada en el host.

Donde se construyó la API Rest fue en Python, haciendo uso de un concepto llamado:

b. Backend

El backend es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no vemos como, por ejemplo, la comunicación con el servidor.

Dentro del desarrollo web, el backend se encarga de todos los procesos necesarios para que la web funcione de forma correcta. Estos procesos o funciones no son visibles, pero tienen mucha importancia en el buen funcionamiento de un sitio web. Algunas de estas acciones que controla el backend son la conexión con la base de datos o la comunicación con el servidor de hosting.

Una página web no solo tiene que ser visualmente atractiva, bien estructurada y con contenido de calidad. Otros aspectos son igual de importantes como la rapidez de carga, la seguridad o el acceso a las búsquedas, por lo que el desarrollo del backend es muy importante. Podemos destacar algunas de sus ventajas.

Las páginas web son entes dinámicos donde los usuarios pueden interactuar con ella. Todo este intercambio de información que se produce en una página web puede generar en errores si el backend no está bien desarrollado. A la hora de implementar el proyecto, es en el backend donde se recibe y verifica la información de los DTE’s que son recibidos de los archivos de entrada del Frontend.

c. Frontend

Frontend es la parte de un sitio web que interactúa con los usuarios, por eso decimos que está del lado del cliente. Backend es la parte que se conecta con la base de datos y el servidor que utiliza dicho sitio web, por eso decimos que el backend corre del lado del servidor. Estos dos conceptos explican a grandes rasgos cómo funciona una página web y son fundamentales para cualquier persona que trabaje en el mundo digital, ya sea en programación, marketing, diseño o emprendimiento.

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

d. Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista– controlador (MVC). Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt.

En junio de 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.

Aunque Django está fuertemente inspirado en la filosofía de desarrollo Modelo Vista Controlador, sus desarrolladores declaran públicamente que no se sienten especialmente atados a observar

estrictamente ningún paradigma particular, y en cambio prefieren hacer "lo que les parece correcto". Como resultado, por ejemplo, lo que se llamaría "controlador" en un "verdadero" framework MVC se llama en Django "vista", y lo que se llamaría "vista" se llama "plantilla".

Gracias al poder de las capas mediator y foundation, Django permite que los desarrolladores se dediquen a construir los objetos Entity y la lógica de presentación y control para ellos.

Conclusiones

La utilización de Django facilita el desarrollo del Frontend a la hora de querer hacer y construir diferentes vistas las cuales pueden ser repetitivas, pero con Django se aceleran esos procesos, así como su fácil utilización al estar basado en Python.

El uso de un API Rest es fundamental en la buena construcción de aplicaciones web para poder manejar toda la información de manera ordenada y prolija.

Las aplicaciones web son el ahora de la gran mayoría del software que se desarrolla debido a la facilidad de uso remoto y acceso a datos. Si uno quiere estar en la jugada estos días, deberá estar siempre actualizado respecto a estos temas.

Referencias bibliográficas

Breatheco, (2019). *Construyendo APIs REST utilizando Flask.*

https://content.breatheco.de/es/lesson/building-apis-with-python-flask

Mozilla ORG, (2021). *Introducción a Django.*

https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction

Diagrama de Clases

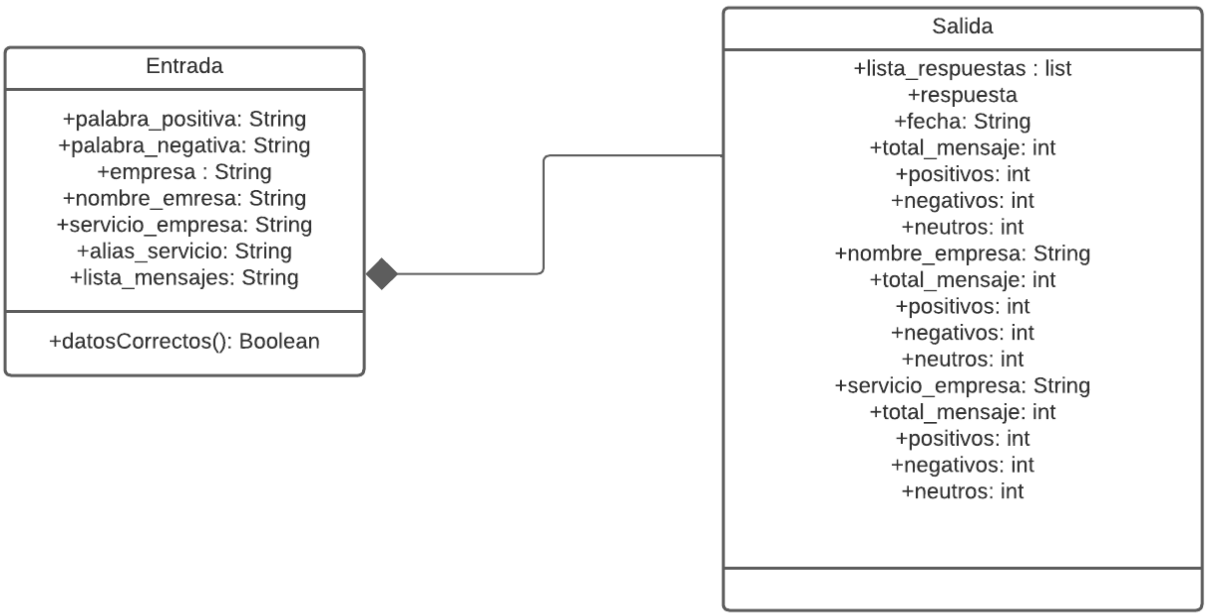

Figura 2. Diagrama de clases para la elaboración del proyecto