
PROYECTO NO.3 - CHAPINCHAT

201901103 – Josué Daniel Rojché García

Resumen

La aplicación consiste en el funcionamiento de mensajes que realicen los usuarios, los cuales servirán para realizar cálculos que produzcan el resultado de actividades cargadas en el sistema, estos resultados estarán basados en cálculos de acuerdo con los datos del mensaje postado por el usuario, de esta manera tendrá la posibilidad de saber los porcentajes de ser un perfil en específico de acuerdo con sus actividades.

La red social es de suma importancia en el ámbito nacional e internacional ya que permite a los usuarios interactuar con otros a través de mensajes, además puede obtener reportes en formato pdf que permiten al usuario descargar sus datos, de tal manera que puede ser de gran utilidad para compartir sus resultados. También tiene la posibilidad de eliminar los datos que ya ha sido cargados en el sistema de forma previa. Algunos de los resultados obtenidos, el usuario los puede visualizar en tablas, o a través de formato XML.

Palabras clave

Social, Frontend, Backend, listas, probabilidad.

Abstract

The application consists of the operation of messages made by users, which will be used to perform calculations that produce the result of activities loaded in the system, these results will be based on calculations according to the data of the message posted by the user, in this way in this way, you will have the possibility of knowing the percentages of being a specific profile according to your activities.

The social network is extremely important nationally and internationally since it allows users to interact with others through messages and can also obtain reports in pdf format that allow the user to download their data, in such a way that it can be of great utility to share your results. You also have the possibility of deleting data that has already been previously loaded into the system. Some of the results obtained can be viewed by the user in tables, or through XML format.

Keywords

Social, Frontend, Backend, Lists, Probability.

Introducción

Para darle solución al programa se deben tener conocimientos básicos en el manejo de tecnologías WEB, algunos de estos pueden ser JavaScript, HTML, CSS, etc. Sin embargo, para darle solución a este proyecto, se utiliza el framework django, Python, HTML, CSS, y Flask.

Es de suma importancia comprender que rol ocupa cada una de estas tecnologías, por ejemplo, Python servirá como el lenguaje de programación, y para la parte de Backend que junto a Flask permiten el desarrollo de la lógica con la cual se realizaran consultas, por lo que también es importante conocer HTTP para poder obtener, eliminar o enviar información.

También es importante el manejo de bases de datos, en este caso se hace uso de XML, la cual almacena cada parte de la información necesaria para las cargas de datos al sistema, así también para la obtención de respuestas, luego de haber procesado la información.

Desarrollo del tema

Como primer punto se maneja la base de la información, la cual será manejada por el Backend. Otra parte fundamental del software es el Frontend, el cual permite la interacción con el usuario.

a. Backend

Es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios. Además, contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma en la que el navegador hace solicitudes.

Los desarrolladores backend se encargan de hacer que la lógica del sitio funcione correctamente, la

información se trasmita de manera segura y el desempeño de la aplicación no entorpezca la experiencia del usuario.



Figura 1. Backend.

Fuente: elaboración propia.

b. 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.

Los desarrolladores frontend son aquellos que crean los componentes visuales de un sitio web. Dan formato a las imágenes, la animación y la interactividad de un sitio web.



Figura 2. Frontend.

Fuente: elaboración propia.

c. XML

Es una estructura de archivo que tiene en orden jerárquico los datos y permite que los programas que se comunican entre sí puedan mantener la información y no perderla durante el manejo de estos.

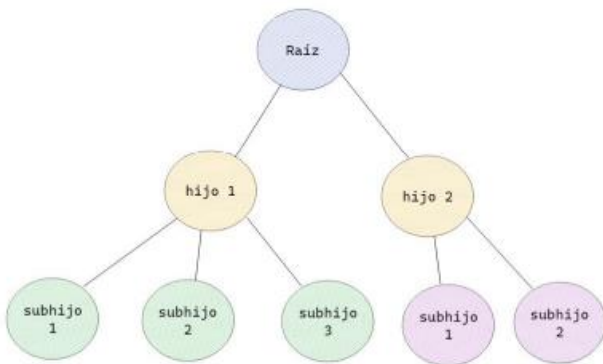


Figura 3. Estructura de árbol de un archivo XML.

Fuente: elaboración propia.

d. Django

Es un marco web Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Creado por desarrolladores experimentados, se ocupa de gran parte de las molestias del desarrollo web, por lo que puede concentrarse en escribir la aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto.



Figura 4. Django

Fuente: Django, 2023.

e. Flask

Es un framework micro web escrito en Python. Se clasifica como un microframework porque no requiere herramientas o bibliotecas particulares. No tiene capa de abstracción de base de datos, validación de formularios ni ningún otro componente en el que las bibliotecas de terceros preexistentes proporcionen funciones comunes. Sin embargo, Flask admite extensiones que pueden agregar funciones de aplicación como si estuvieran implementadas en Flask. Existen extensiones para mapeadores relacionales de objetos, validación de formularios, manejo de carga, varias tecnologías de autenticación abierta y varias herramientas relacionadas con el marco común.



Figura 5. Django

Fuente: Flask, 2023.

f. Interfaz

La interfaz contiene varias opciones que el usuario puede realizar según sea el caso. Al iniciar la aplicación lo primero que se visualiza es un mensaje de bienvenida a la aplicación.

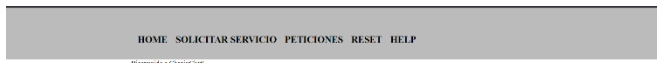


Figura 6. Página Principal.

Fuente: elaboración propia.

Cuando presione el botón de Home, siempre observará el mensaje de bienvenida.

g. Solicitar servicio.

Si presiona en el botón de solicitar servicio, entonces tendrá espacio para cargar el archivo XML, y tendrá también otro espacio en el cual recibirá la respuesta de la petición en formato XML.



Figura 7. Opción Solicitar servicio.

Fuente: elaboración propia.

h. Peticiones.

En el caso de la opción Peticiones, se tendrán varias opciones, en las cuales podrá consultar por medio de fecha y usuarios los datos procesados por las entradas en las opciones anteriores, por lo cual se mostrará unas tablas que contendrán la información de porcentajes, además la otra opción obtendrá los datos de los pesos y también serán presentadas en tablas, estos se pueden realizar búsquedas solamente por medio de los usuarios.



Figura 8. Opción peticiones.

Fuente: elaboración propia.

i. Reset.

La opción de reset permite al usuario el eliminar todos los datos que ya han sido cargados en el sistema, de esta manera puede realizar nuevamente las veces que quiera, el ingreso de los datos.

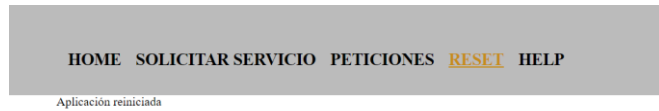


Figura 9. Opción reset.

Fuente: elaboración propia.

j. Help.

La opción de ayuda permite obtener los datos del desarrollador, así como la opción para visualizar este documento.



Figura 10. Opción help.

Fuente: elaboración propia.

Para realizar el cálculo de los porcentajes de probabilidad por perfil en un mensaje.

$$r = \left(\frac{p}{t}\right) * 100 \quad (1)$$

donde:

r = resultado en %.

p = cantidad de palabras en el mensaje correspondientes a un perfil.

t = total de palabras, sin incluir signos, números, y palabras a excluir.

Para realizar el cálculo de los pesos para cada perfil, correspondiente a un usuario.

$$r = \left(\frac{\sum p}{t}\right) \quad (2)$$

donde:

r = resultado.

$\sum p$ = Sumatoria del porcentaje de probabilidad de un perfil por usuario.

t = cantidad de mensajes con porcentajes diferentes de cero.

Conclusiones

El manejo correcto de los datos a través de la separación entre Backend y Frontend, permiten tener un control adecuado de los datos, sin perder la seguridad del manejo de estos, por lo cual es importante tener buenas practicas para no perder información de los usuarios.

El control de la aplicación debe ser fácil en su uso, por lo que es de suma importancia la experiencia de usuario, ya que, si la aplicación no es lo suficientemente amigable con el usuario, entonces, estos no harán uso habitual de la misma.

Referencias bibliográficas

Graphviz. (28 de 02 de 2023). *graphviz*. Obtenido de graphviz: <https://graphviz.org/>

Luis Joyanes Aguilar, I. Z. (2008). *Estructuras de datos en Java*. España: Mc Graw Hill.

Nicole. (2018). *Platzi*. Obtenido de platzi: <https://platzi.com/blog/que-es-frontend-y-backend/>.

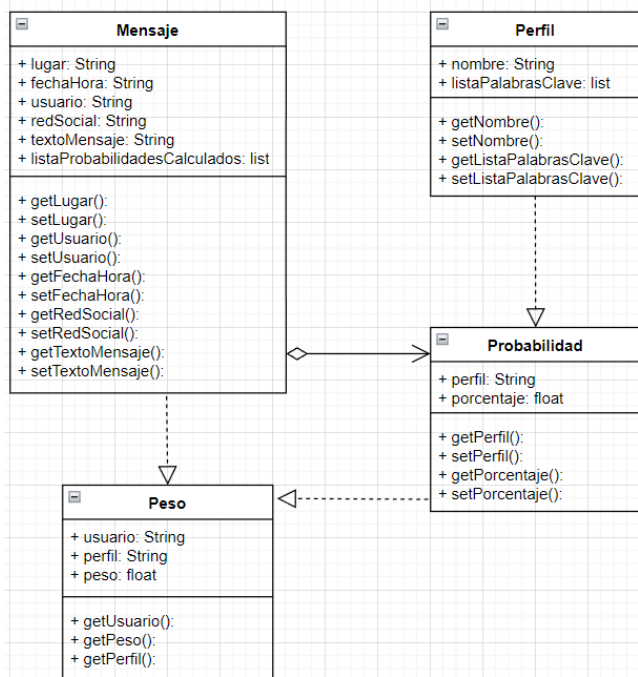
North, E. R. (2015). *Drawing graphs with dot*.

w3schools. (25 de 02 de 2023). *w3schools*. Obtenido de w3schools: <https://www.w3schools.com/xml/default.asp>

Anexos

Imagen I.

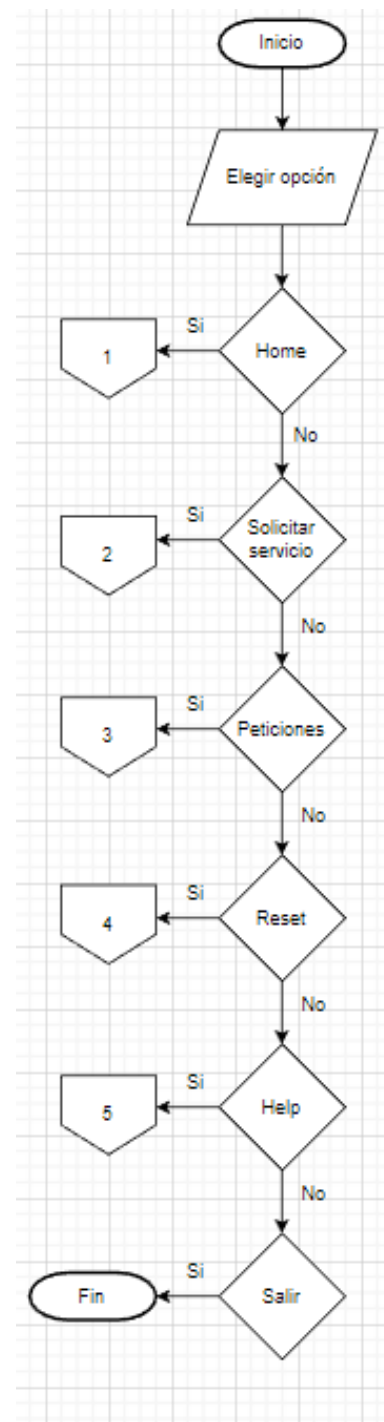
Diagrama de Clases.



Fuente: elaboración propia.

Imagen II.

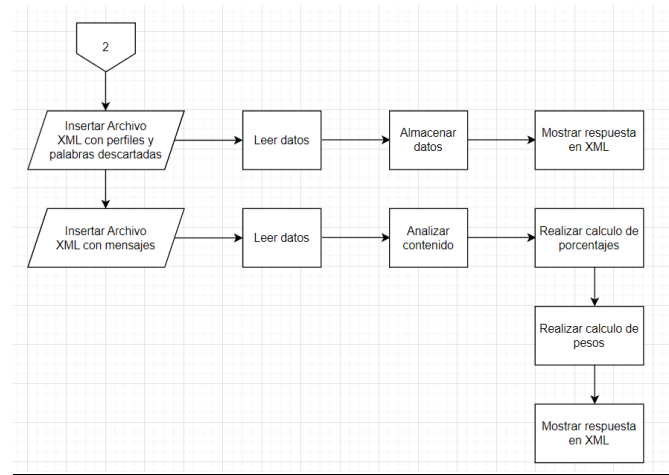
Diagrama de Flujo, Inicio.



Fuente: elaboración propia.

Imagen III.

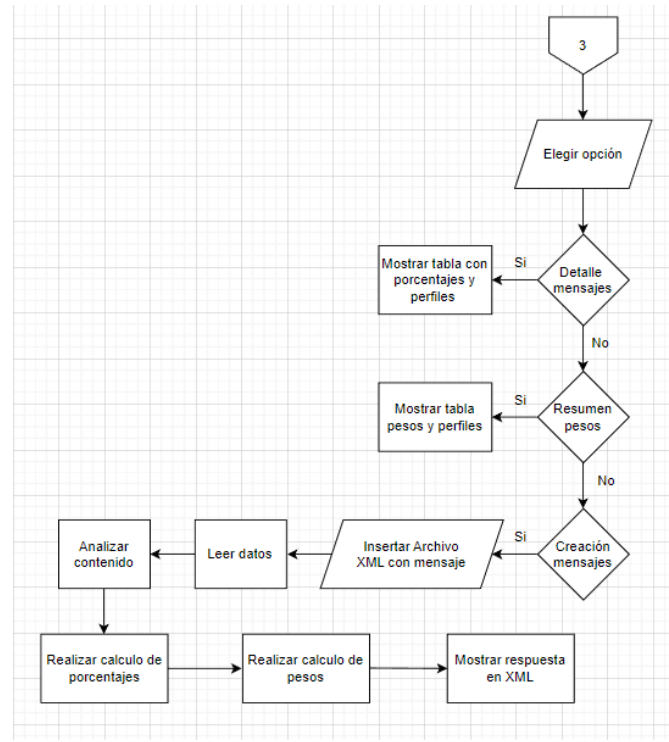
Diagrama de Flujo, solicitar servicio.



Fuente: elaboración propia.

Imagen IV.

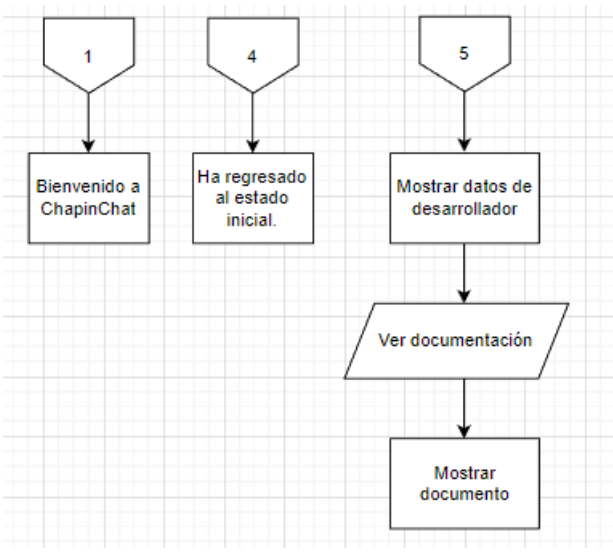
Diagrama de Flujo, peticiones.



Fuente: elaboración propia.

Imagen V.

Diagrama de Flujo, otras opciones.



Fuente: elaboración propia.