
PROYECTO 3 LABORATORIO IPC2

201906173 – Sharon Estefany Tagual Godoy

Resumen

El ensayo se presentan los componentes utilizados para la realización del tercer proyecto de IPC2 en el cual se realizó una aplicación web con estructura MVT, teniendo un frontend realizado con el framework Django y un backend que realiza las peticiones respectivas con Flask, como bien sabemos, estos frameworks trabajan con Python, es decir se aplicó también el lenguaje Python para poder manejarlos como lo indica su sintaxis.

Para almacenar estos datos se hizo uso de programación orientada objetos creando un arreglo de objetos en el cual se estarían guardando para luego poder hacer uso de ellos en diferentes procedimientos, estos datos fueron almacenados no sin antes haberse extraído de un archivo de entrada ejecutado por el cliente, desde el frontend, para su posterior análisis a través de diferentes estructuras, e implementando expresiones regulares; las cuales son utilizadas a través de librerías para hacer ciertas validaciones en la lógica de nuestro programa.

Todo esto al ser cargado, se mostró en un archivo de salida, con extensión xml para poder utilizarlo como una base de datos la cual almacena nuestros elementos ya validados.

Palabras clave

objetos, datos, interfaz, archivo, peticion.

The essay presents the components used for the realization of the third IPC2 project in which a web application with MVT structure was made, having a frontend made with the Django framework and a backend that performs the respective requests with Flask, as we know, these frameworks work with Python, that is, the Python language was also applied to handle them as indicated by its syntax.

To store this data, we made use of object-oriented programming creating an array of objects in which they would be stored to then be able to make use of them in different procedures, these data were stored not before having been extracted from an input file executed by the client, from the frontend, for further analysis through different structures, and implementing regular expressions; which are used through libraries to make certain validations in the logic of our program.

All this when loaded, was shown in an output file, with xml extension to be used as a database which stores our validated elements.

Keywords

objects, data, frontend, file, request.

Introducción

A continuación se presentan ciertos temas implementados en el desarrollo del proyecto realizado, con el objetivo de fomentar las ideas plasmadas en dicho proyecto, como lo son la aplicación de los frameworks Django y Flask, que fueron utilizados para poder darle la funcionalidad respectiva al proyecto realizado, además se hace mención de librerías utilizadas como lo son “re” para poder realizar el análisis de las expresiones regulares que fueron utilizadas para la lectura de los datos, y de otras que fueron requeridas para su respectivo uso.

Desarrollo del tema

La aplicación consiste en crear una estructura de Cliente-Servidor, con los frameworks Django y Flask, la implementación de Django se realizó en el Frontend de la aplicación el cual representa la parte visual, es decir la interfaz con la que el usuario interactúa y realiza las respectivas acciones.

La utilización de Flask se implementó en el backend, para poder realizar la lógica que realice peticiones haciendo la ejecución del programa. Cada uno fue levantado en un puerto de la computadora propia, en este caso el backend fue levantado en el puerto 5000, y el frontend en el puerto 8000. Esto para que los dos se pudieran gestionar en paralelo y así que la aplicación no generara problemas al ejecutarla.

Frameworks:

Marco o entorno de trabajo que está sintetizado por herramientas, librerías y buenas prácticas utilizado para crear aplicaciones.

Dicho de otra manera un framework para desarrollo web es un esquema conceptual con módulos de

software específicos, el cual pasa a ser tomado como base para el desarrollo de diferentes proyectos informáticos.

Modelo Vista Template:

Este modelo lleva cierta estructura, para realizar nuestro manejo de datos en una aplicación web, como se muestra en la figura I.

- ☐ Model Template View (MTV)
- ☐ Vistas->Template
- ☐ Controlled->View
- ☐ Model->Model

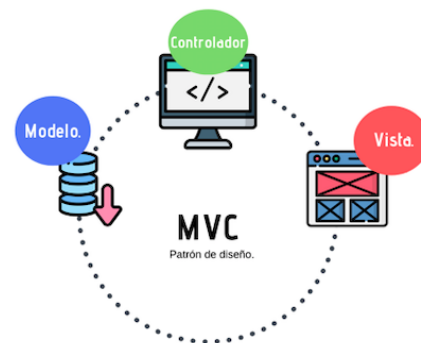


Figura I.

Representación de la estructura MVT

Frontend:

Este es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas (RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros

elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario.

El frontend es el encargado de la experiencia del usuario, es decir, en el momento en el que este entra a una página web, debe ser capaz de navegar por ella, por lo que el usuario verá una interface sencilla de usar, atractiva y funcional.

Como bien sabemos la interfaz que maneje el usuario para hacer sus respectivas acciones en cualquier aplicación web, debe de ser intuitiva y amigable, para ello se aplicó HTML y CSS para poder dar estilo a la misma, y así brindar una mejor experiencia al usuario

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.

Para mandar los datos del archivo que carga el usuario, se selecciona el botón respectivo desde el frontend, para poder mandar el archivo que queramos, siempre y cuando este contenga con la estructura correcta. De lo contrario este no cargará el archivo y se mostrará como error.

Los archivos cargados son enviados al backend para su correcto análisis y validación, como el objetivo de la aplicación es realizar solicitudes necesarias para la validación de las peticiones, estas son analizadas y validadas por medio de expresiones regulares las cuales verifican que los datos contengan la estructura correcta y la sintaxis correcta, según se indica en el proyecto.

Expresiones regulares:

Una expresión regular es una cadena de caracteres que es utilizada para describir o encontrar patrones dentro de otros strings, en base al uso de delimitadores y ciertas reglas de sintaxis. La mayoría de los programadores no se dan el tiempo de aprender a aplicar las expresiones regulares, lo cual es una lástima ya que son de gran utilidad. Cuando aprendas a aplicar las expresiones regulares, te darás cuenta de lo poderosas que son, la gran cantidad de problemas que pueden resolver, y lo mucho que aumentará tu productividad al programar.

En el proyecto fue indispensable la aplicación de las expresiones regulares para la lectura y análisis del formato exigido por la misma, algunas de ellas fueron:

```
re.search(r"(\d{1,2}[-/] \d{1,2}[-/] \d{4})", arreglo[p].getTiempo()).group(0)
```

Como bien observamos en la imagen anterior, se hace uso de la librería re, la cual nos ayuda a implementar expresiones regulares en el lenguaje de programación python, y así verificar su formato correcto en este caso se analiza la escritura correcta de la fecha obtenida.

GET:

el método GET añade los datos a la URI definida en el atributo de acción del formulario.

POST:

el método POST adjunta datos al organismo solicitado.

Conclusiones

Los frameworks son muy útiles e importantes en el desarrollo web, ya que facilitan el uso al utilizarse como marcos de trabajo, además de que brindan diversas opciones que pueden ser aprovechadas para la realización de lo que se requiere.

La programación orientada a objetos es bastante útil cuando queremos hacer búsqueda y referencia de ciertos elementos con muchos atributos, en este caso el manejo de peticiones, para el análisis correspondiente.

El modelo MVT sirve de gran utilidad para llevar el respectivo orden al manejar los datos, haciendo así una forma más eficiente y ordenada al momento de trabajar con desarrollo web.

Referencias bibliográficas

Uniwebsidad, Eugenia Bahit POO en Python.

(2006-2021). Uniwebsidad, Fundamentos de Git.

(2005-2021). Web Docs:

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