
SEGUNDO PROYECTO

201700490 – Marvin Daniel Castellanos Castillo

Resumen

Django y Flask son frameworks de Python, que permiten el desarrollo de aplicaciones web. Con Django es posible crear una aplicación web y desarrollar tanto el Frontend como el Backend, lo que la convierte en una poderosa herramienta para los desarrolladores.

Flask, es un framework fácil de comprender y sencillo de levantar aplicaciones para procesos y cálculos tanto del lado del servidor(Backend) como del cliente(Frontend), lo que lo convierte en un poderosa herramienta para creación de aplicaciones relativamente pequeñas.

Dado que Django y Flask, son uno de los frameworks más usados de Python, es importante el comprender la estructura de ambas herramientas, es por ello por lo que el siguiente proyecto buscaba la creación de una aplicación web que integrara ambas tecnologías, que, a grandes rasgos, procesa, traduce y muestra datos de forma dinámica al usuario.

En el siguiente ensayo, se presenta la forma en la que el proyecto fue desarrollado para integrar estas dos tecnologías.

Palabras clave

Backend, Frontend, Django, Flask

Abstract

Django and Flask are Python's frameworks that allows develop web apps. With Django is possibly create a web app and develop as frontend as backend, which converts it in a powerful tool to developers.

Flask is an easy understand framework and setting up applications to process and calcs by the server side(backend), which converts it in a powerful tool to create applications relatively small.

Given that Django and Flask are one of the most used frameworks form Python, is important understanding the structure of both tools. By that, the next project looks for creation of a web app that integrates both technologies, that process, translate, and shows data in a dynamic way to the user.

In the next essay, its presented the way in the project was developed to integrate this two technologies.

Keywords

Backend, Frontend, Django, Flask.

Introducción

En la actualidad, los servicios web dominan, en su mayoría, el mercado de la información digital. Para desplegar una aplicación para que sea accedida como un servicio en la web hay que tener en cuenta varios factores, tales como el hospedamiento, protocolos de comunicación, programación. Es por ello que existen distintas formas para poder asentar la estructura de la aplicación, tanto física como virtual.

Django es un framework en Python el cual permite el desarrollo de aplicaciones web tanto del lado del frontend como del backend; con una arquitectura "Model View Template (MVT)" la cual nos permite crear, procesar y administrar la información de manera dinámica y forma sencilla. Es por ello que su uso para creación de páginas web ha aumentado en los últimos años.

Flask es un framework en Python el cual permite el desarrollo de aplicaciones del lado del servidor de una aplicación web. Su sencillez y limpieza lo convierten en uno de los idóneos para la creación de páginas web de escala relativamente pequeña.

Tanto Django como Flask es posible utilizarlos para poder crear una aplicación web. Es por ello que se desarrolló una aplicación web con estos dos frameworks, Django y Flask, para el Frontend y Backend, respectivamente.

Desarrollo del tema

a. Definición del proyecto

Una de las estructuras más utilizadas para el desarrollo de aplicaciones es el modelo cliente-servidor, el cual provee la capacidad de procesar

los datos en un servidor y no en la máquina de forma local, lo que permite un mejor desempeño de la aplicación en momento de ejecución y así el usuario únicamente interactuar con información mostrada de manera dinámica.

Existen distintas plataformas y frameworks que permiten el desarrollo de estructuras cliente-servidor, como: Angular, nodeJS, Django, Flask, Laravel, etc. y escoger el adecuado para el proyecto queda a discreción del desarrollador.

“Django es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python.”. Con Django es posible la creación de aplicaciones web y configurar tanto el frontend como el backend, ya que posee un modelo MVT.

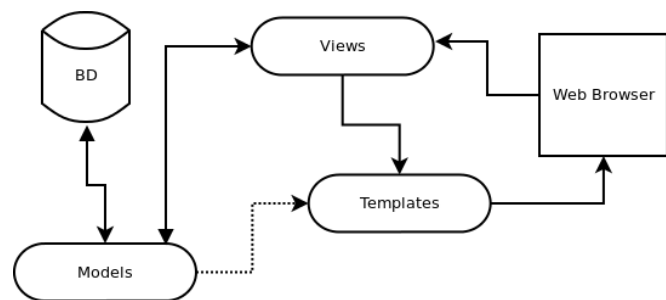
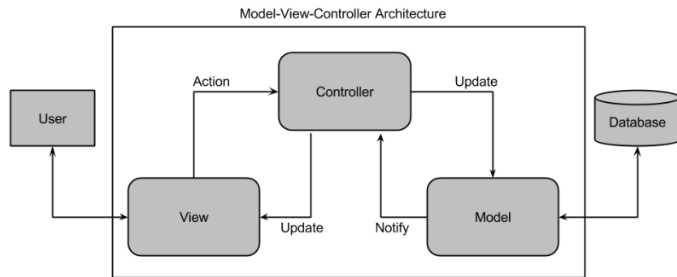


Grafico 1. Modelo, Vista Plantilla(model, view, template)

Fuente: <https://docs.hektorprofe.net/django/web-personal/patron-mvt-modelo-vista-template/>

“Flask es un “micro” Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC.”. Es por ello por lo que es sencillo la creación de páginas de manera rápida.



Grafica 2. Modelo, Vista, Controlador (Model, View, Controler)

Fuente: <https://www.patricksoftwareblog.com/tag/mvc/>

Para aplicar la arquitectura cliente-servidor, se utilizó Django como frontend y flask como backend.

El proyecto constaba de dos lados los cuales proveían las funcionalidades necesarias para la realización de lo que se requiere. En el frontend se contaba con una página web la que contenía las opciones de la aplicación, la cual permitía seleccionar un archivo XML del cual se procesaría su información para el posterior análisis de estos. El archivo XML contenía la información de los datos de solicitudes de aprobación de procesos fiscales, conteniendo los datos de NIT, fechas y valores. Al cargar el archivo XML se podía visualizar la estructura del archivo en la ventana principal, de la cual era posible realizar modificaciones previo al envío al Backend para su procesamiento. Luego de su procesamiento en el Backend se obtenía una respuesta la cual contenía los datos necesarios para desplegar de manera gráfica los datos solicitados. De igual forma era posible la

Del lado del backend, se recibía el XML cargado en el frontend para posteriormente interpretar los datos, analizarlos y reformular un nuevo XML de respuesta con los datos generados.

b. Estructura del Frontend

La página principal de la aplicación, la cual es con la que interactúa el usuario, tiene una vista simple de las funcionalidades del proyecto, dado que

únicamente cuenta con una barra Header la cual tiene el título del proyecto y las opciones de la aplicación de las cuales se presenta como popups mostrando los datos de estudiante y formularios de búsqueda, según sea el caso.

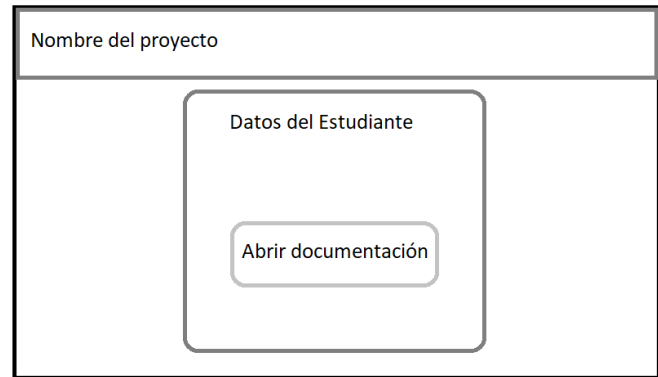


Grafico 3. Presentación de despliegue de datos de estudiante.

Fuente: Elaboración propia.

En el cuerpo de la página se muestra en un recuadro el XML cargado previo a enviarlo al Backend, dicho texto producido puede ser alterado, agregando etiquetas con información para luego enviarlo para su debido procesamiento.

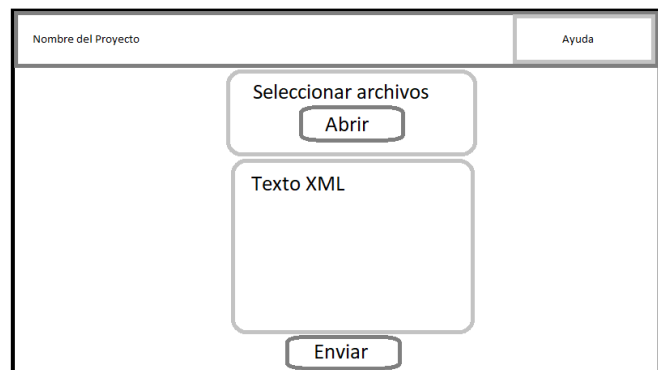
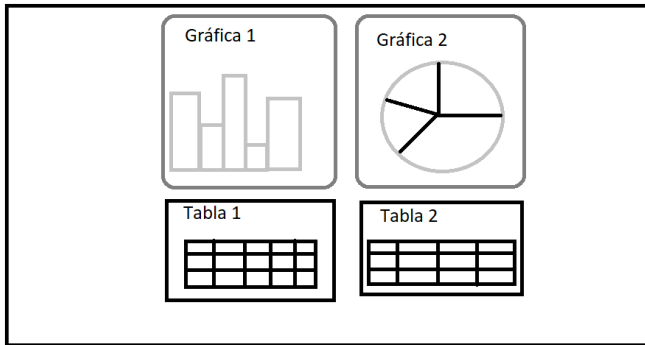


Grafico 4. Presentación página principal de usuario

Fuente: Elaboración propia

Luego de recibir el XML reestructurado del Backend, los datos obtenidos son utilizados para desplegar

gráficas y tablas que muestran la información de manera dinámica, en otra página llamada “/graficas”



Grafica 5. presentación de pagina de graficas.

Fuente: Elaboración propia

c. Flujo de despliegue del Forntend

Cuando el usuario presiona el botón de abrir en la pestaña principal de la aplicación, se abre un navegador del sistema el cual permite seleccionar el archivo a leer que residan en el ordenador, para su posterior lectura. Luego de seleccionado el archivo, automáticamente se despliega en el recuadro de texto, para su posible modificacion Los problemas que busca evitar son:

El XML se muestra automáticamente en la pantalla en su recuadro correspondiente, el cual permite la modificación, ya sea eliminación o adición de etiquetas para luego enviarlos al backend para su respectivo procesamiento.

Luego de validados los datos y modificado los datos, si se necesita, se procede a enviar el XML el cual, si no se modifica, será el que se enviara al Backend tal y como fue cargado.

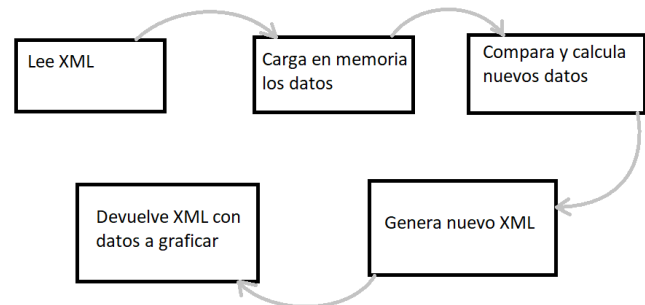
d. Flujo de despliegue del Frontend

Luego de recibir el XML con los datos, se leen y cargan los datos a memoria para su posterior procesamiento.

Una vez cargados en memoria, se comparan y suman los datos necesarios para evitar duplicidad de la información.

Posteriormente se genera un nuevo XML el cual se devuelve al Frontend para su posterior graficación de los datos.

Dado que el backend únicamente realiza cálculos y comparaciones, no se cuenta con una interfaz gráfica para que el usuario interactúa con ella.



Grafica 7. Flujo de trabajo del Backend

Fuente: Elaboración propia

Conclusiones

La arquitectura Cliente-servidor es una poderosa herramienta la cual es ampliamente utilizada hoy en día.

Existen varios freameworks, los cuales permiten el desarrollo de aplicaciones web las cuales estén basadas en la arquitectura cliente-servidor.

Django es una fuerte herramienta para la creacion de paginas web dinámicas.

Flask es una fuerte herramienta para la creaicion de paginas web dinámicas.

Es de vital importancia, como desarrollador, el estar a la vanguardia con las nuevas tecnologias que son creadas para solución de problemas, tales como la creación de aplicaciones web.

Referencias bibliográficas

¿Que es Django?

<https://tutorial.djangogirls.org/es/django/>

03/011/2021 06:35 PM

¿Que es Flask?

<https://openwebinars.net/blog/que-es-flask/>

03/011/2021 7:30 PM