
ARQUITECTURA MULTISERVICIOS UTILIZANDO UNA ARQUITECTURA DE 2 CAPAS CON APLICACIONES WEB EN PYTHON

201801444 – José Haroldo Donis Alonso

Resumen

Un framework es una manera de hacer las cosas de una mejor manera usando un lenguaje base como en este caso es Python, En este ensayo se hablará un poco del framework y sobre el proyecto el cual solicito el cliente. Se hablará de forma técnica para que el cliente pueda comprender de una buena forma la solución planteada del proyecto final de un curso complejo de la carrera de Ingeniería en Ciencias y Sistemas.

Este Proyecto está desarrollado en 2 etapas, un backend que proporciona acceso a una API, la cual permite enviar y recibir documentos XML, los cuales son procesados por la API, luego está un frontend, desde el cual se accede al contenido almacenado en la API y permite interactuar con ella, permitiendo de este modo administrar la información de la misma.

Palabras clave

- XML
- API
- DJANGO
- FLASK
- BACKEND
- FRONTEND

Abstracta

A framework is a way of doing things in a better way using a base language such as python in this case. In this essay we will talk a little about the framework and about the project which the client requested. It will speak in a technical way so that the client can understand in a good way the solution proposed for the final project of a complex course in the Science and Systems Engineering career.

This Project is developed in 2 stages, a backend that provides access to an API, which allows sending and receiving XML documents, which are processed by the API, then there is a frontend, from which you can access the content stored in the API and allows you to interact with it, thus allowing you to manage its information.

Keywords

- XML
- API
- DJANGO
- FLASK
- BACKEND
- FRONTEND

Introducción

El problema que se nos pide resolver es en una app web, y para el desarrollo de la misma usamos el framework Django como la parte del frontend, es decir la pestaña del navegador donde el usuario interactúa con la aplicación. Este framework usa la programación MVT, ya que los templates (lo que es las vistas) (es decir lo que el cliente puede ver), y en Python se desarrolla la lógica (conocido como views) en Django.

Django es un marco web de Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático muy fácil de crear sitios web de manera dinámica usando un patrón MVT (MODELS-VIEWS-TEMPLATES)

Creado por personas experimentadas, se encarga de gran parte de la molestia del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto, no todo lo que es código abierto es gratis, pero en esta ocasión la licencia lo ofrece gratis.

Se utilizó el formato XML para la lectura de los datos, y después manipularlos en la API, la principal esencia de este proyecto fue la simulación de una base de datos donde el almacenamiento de datos fue en un archivo xml donde podríamos hacer consultas para saber que tipo de información y con ayuda de graficas saber los datos de estamos viendo en el frontend de igual manera se diseñó la interfaz utilizando la funcionalidad de templates de django y además se maneja los formularios de manera estándar desde el backend de Django hacia la api escrita en Flask

Desarrollo del tema

La Superintendencia de Administración Tributaria le ha solicitado construir un software que pueda ser consumido desde Internet como un servicio.

Este software recibirá un mensaje con los datos para solicitar la autorización de un Documento Tributario Electrónico (DTE) emitido por un contribuyente y como respuesta emitirá un número único de autorización, este número será un correlativo que iniciará con el valor 1 cada día y no deberá repetirse de nuevo en ese día.

La estructura del número de autorización es la siguiente:

yyyymmdd#####,

donde: yyyy corresponde al año en que se solicita la autorización mm corresponde al mes en que se solicita la autorización

dd corresponde al día en que se solicita la autorización

corresponde al correlativo precedido por la cantidad de 0's necesarios para mantener el formato:

Ej. Para el correlativo 1: 00000001.

Para esto se desarrolló una aplicación web, desarrollada en Django, se desarrolló el Back-end en Python y el Front-End con Html y Bootstrap (es una librería).

Tuvimos que implementar la programación MCV, ya que este framework implementa este tipo de programación.

Para la solución se creó dos servicios, la cual la arquitectura es la siguiente

Nota :

Este proyecto fue creado con la ARQUITECTURA CLIENTE-SERVIDOR donde

ARQUITECTURA

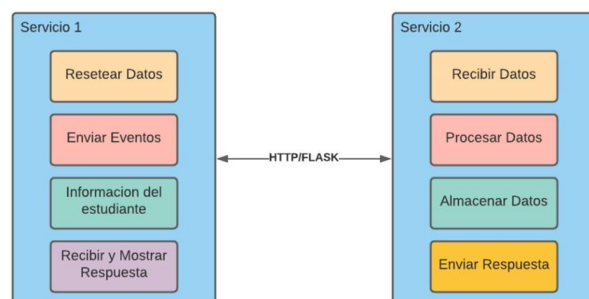


Figura 1: Arquitectura de la aplicación

Figura I: Diagrama de servicios
Fuente: Documentación del proyecto.

Servicio 1(CLIENTE):

Frontend(Desarrollado en Django) y se utilizo otros clientes para acceder a los servicios como Postman/Insomnia

El servicio 1 fue creado utilizando el framework Django, utilizando su funcionalidad de templates se diseño la interfaz como un servicio web, esta interfaz permite al usuario cargar archivos, que serán enviados al backend en django, para posteriormente ser procesados en la api Flask.

Servicio 2(SERVIDOR)

Backend(Desarrollado en Flask) y se utilizo para manipular toda la información necesaria para realizar peticiones por medio de los métodos request http POST y GET

NOTA:

Para diseñar el frontend se utilizo la librería CSS Bootstrap, con la cual se aplicaron los estilos usados por la aplicación en la vista del usuario

Los componentes los cuales se crearon fueron:

CARGAR ARCHIVO: este componente se realizó en el servicio 1, en el lado del front-end, el cual sirve para interactuar con el usuario, aquí servirá para que el usuario suba su archivo a la app web. Y a través del back-end pueda procesar el archivo.

PETICIONES: El programa lleva diferentes opciones las cuales pide el usuario, a continuación, se presentan las que solicitan en el enunciado:

- **Consultar Datos:** esta opción se manejará en el lado del servidor 2, la cual seguirá siendo el back-end, donde procesará los datos y hará un filtrado para ordenar los datos pedidos al usuario.

Proyecto Final-IPC2

Consultar Datos Peticiones Ayuda

Entrada Salida

Seleccionar archivo No se eligió archivo Enviar Cancelar

Filtrar información por fecha y nit que reporte: esta opción se realizó un filtro donde se organice la información específica con lo que pide el enunciado todos los movimientos de los nits

- **Filtrar por rango de fechas:**

En esta opción se filtro un rango de fechas el cual se selecciona 2 fechas y todas aquellas fechas que estén dentro de estas 2 fechas se visualizara el iva ingresado por cada fecha encontrada

AYUDA:

Para la realización de esta opción, se creo un navbar donde muestra las opciones que solicitaron en el enunciado, las cuales fueron:

- **Información del estudiante**
- **Documentación**

BOTONES:

- **ENVIAR:**
 - Este botón servirá para enviar el archivo xml para la manipulación, esta información se maneja en el back-end, ya que aquí se procesara los datos del archivo
- **RESET**
 - Este botón se creó para poder limpiar la información procesada desde el back-end

Bibliografía:

<https://www.djangoproject.com>

<https://docs.python.org/es/3/>

<https://flask.palletsprojects.com/en/2.0.x/>

