

---

## ARQUITECTURA MULTISERVICIOS UTILIZANDO APLICACIONES WEB EN PYTHON

---

201900462 – Xhunik Nikol Miguel Mutzut

### Resumen

Un framework es una manera de programar usando un lenguaje base, en este caso el lenguaje base sobre el cual se uso este framework es Python. En este ensayo se hablara 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.

Este Proyecto esta 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 esta 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

### Abstract

*A framework is a way of programming using a base language, in this case the base language on which this framework was used is Python. In this essay we will talk a little bit about the framework and about the project requested by the client. We will talk about it in a technical way so that the client can understand in a good way the solution of the project. 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 interact with it, thus allowing to manage the information of the same.*

### Keywords

- XML
- API
- DJANGO
- 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. este framework usa la programación MVC, ya que los templates(lo que es las vistas), y en Python se desarrolla la logica.

Django es un marco web de Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Creado por desarrolladores experimentados, 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 utilizo el formato XML para la lectura de los datos, y después manipularlos en la API, se diseño la interfaz utilizando la funcionalidad de templates de django y además se manejo los formularios de manera estándar desde el backend de dkango hacia la api escrita en Flask

## Desarrollo del tema

El planteamiento del problema que se nos entregó fue el siguiente:

Una empresa de software le ha solicitado construir un software que pueda ser consumido desde Internet como un servicio. Este software recibirá un mensaje de la bitácora del software principal de la compañía y producirá una serie de información estadística relacionada

Para esto se desarrolló una aplicación web, desarrollada en Django, se desarrollo 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:

### ARQUITECTURA

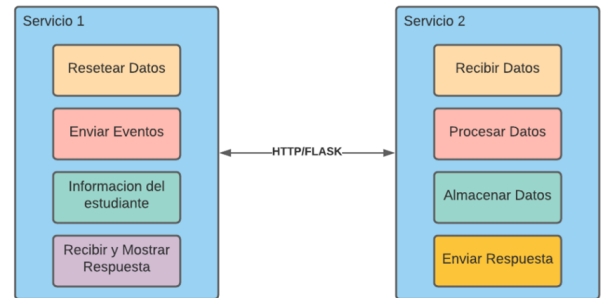


Figura 1: Arquitectura de la aplicación

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

El servicio 1 fue creado utilizando el framework Django, utilizado 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.

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.



Figura II. Prototipo de interfaz

Fuente: Elaboración propia.

¿Por qué se desarrollo lo en Django?

En Django se utiliza la programación MVC, esto nos permite a tener una buena conexión con Flask, el cual se uso para crear la api, lo que hablaremos mas adelante,

En Django se crearon las rutas, las vistas y la parte de la logica lo cual fue:

Lectura del archivo XML:  
el xml los leímos como un archivo, para la manipulación de los archivos tuvimos que recurrir a la api

Manipulación del archivo XML:

En la manipulación de la data que contiene el xml, se creó una función para que reconociera cada etiqueta y así poder colar la información y presentarla ordenada como lo pide el usuario en el enunciado.

Para esto se creo una api, la cual iba hacer la comunicación entre el servicio 1 y el servicio 2, esto ayuda a que podamos tener una comunicación con cada servicio.

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.

- **Filtrar información por fecha y usuario que reporte:** esta opción se realizó un filtro donde se organice la información específica con lo que pide el enunciado

- **Filtrar por fecha y código de error:** esta opción se realizó un filtro donde se filtra la información por fecha y código de error, donde solo detectara los errores dentro del archivo.

Las ultimas do peticiones mostrar la gráfica ordenada donde se mostrar por el tipo de filtro

### AYUDA:

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

- **Información del estudiante**
- **Documentacion**

### BOTONES:

Se agregaron dos botones los cuales harán una comunicación entre el back-end y el front-end las cuales son los siguientes:

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



*Figura III. RESET*

Fuente: Documentación del proyecto.

## Conclusiones

Este proyecto se concluyó que la empezar a usar un framework, debemos aprender la base del mismo, el cual es un lenguaje, en este caso fue el Framework Django, la base de este lenguaje es Python, los proyectos anteriores nos ayudo a poder comprender o si no nos hubiera sido difícil comprender el manejo del Framework, ya que se necesita una base para poder dominarlo.

## Referencias bibliográficas

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

<https://developer.mozilla.org/es/>

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>