

---

## PROYECTO 3, ARQUITECTURA MULTISERVICIOS UTILIZANDO APLICACIONES WEB EN PYTHON

---

202006629 – Wilber Steven Zúñiga Ruano

### Resumen

Un framework es una manera de programar usando un lenguaje base, en este caso el lenguaje base sobre el cual se usó este framework 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.

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 esta.

### 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 esta 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 diseñó la interfaz utilizando la funcionalidad de templates de django y además se manejó 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 fueel siguiente:

Una empresa de software le ha solicitado construir unsoftware 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 desarrolló el Back-end enPython 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

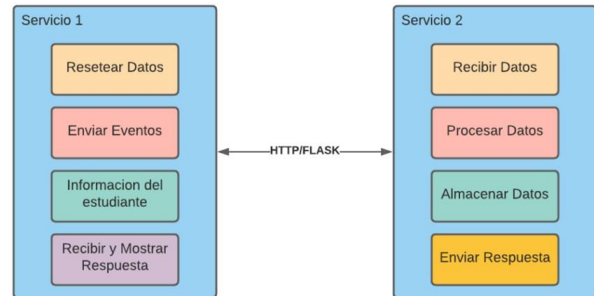


Figura 1: Arquitectura de la aplicación

Figura I: Diagrama de servicios

#### a. 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.

En nuestro caso fue creado utilizando el framework Django, utilizado su funcionalidad de templates se diseñó la interfaz como un servicio web, esta interfazpermite al usuario cargar archivos, que serán enviados al backend en django, para posteriormente ser procesados en la api Flask

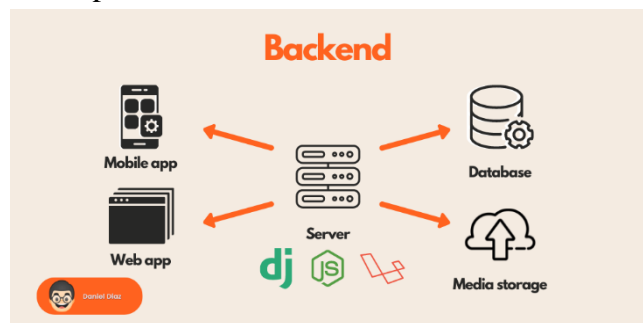


Figura I. Backend y conexiones.

Fuente: <https://geekflare.com>.

## b. Frontend:

El desarrollo web front-end es la práctica de utilizar HTML, CSS y JavaScript para crear aplicaciones web únicas. Aplicaciones como sitios web, sitios web móviles, pero también aplicaciones móviles y Aplicaciones web progresivas.

También puede pensarlo de esta manera, cada sitio web que navega, incluido este, ha sido creado de alguna manera por un desarrollador de front-end. Al menos del lado del cliente

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

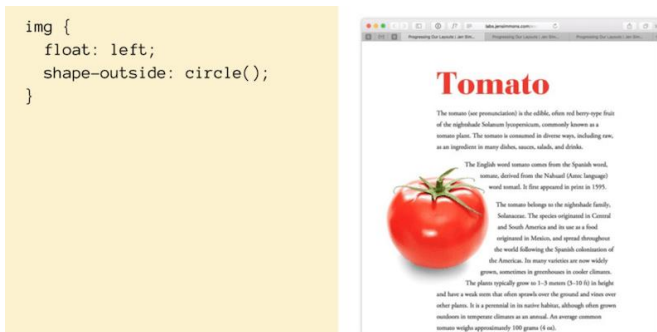


Figura 2. Frontend.

Fuente: <https://geekflare.com>.

El problema que nos presenta es el siguiente:

¿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 archivoXML: 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 creó 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.

Para la solución de este problema se usó el lenguaje Python, el cual importamos las librerías para leer el archivo, usamos la librería de “mini Dom” para la manipulación de archivos XML y poder pasarlas listas enlazadas que es el tema principal del proyecto. Para todo este proceso usamos programación orientado a objetos, el cual creamos varias clases. Esto nos ayudó a crear una mejor estructura de datos, las clases fueron las siguientes:

- app
- empresa
- mensaje
- servicio

La clase principal es app, donde se manejan todos los métodos, se inicia con un método get para montar tu servidor luego tenemos otros como lo es cargar el archivo, recibe el contenido de tu archivo XML posteriormente la pasaa una variable y usando la librería miniDom lo convierte en un archivo miniDom XML que es manipulable y de fácil acceso a todos sus atributos.

Los métodos más importantes dentro de la clase app:

1. GET: El cual envia información hacia mi frontend para poder mostrarla gráficamente.
2. POST: Este método es con el que recibo todos los datos desde mi frontend para poder manipularlos en el back, comúnmente con un request.

Tabla I.

*Variables principales*

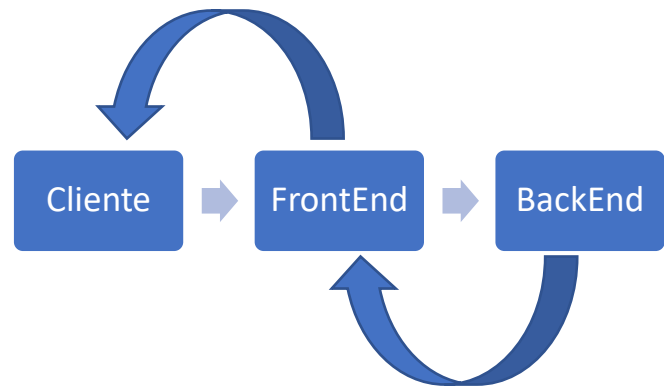
CATEGORÍA	CATEGORÍA
positivas	Lista de palabras positivas
negativas	Lista de palabras negativas
listaMensajes	Lista de mensajes

Fuente: elaboración propia.

## Referencias bibliográficas

- C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.
- Narciso Martí, Estructura de datos y métodos algorítmicos, Madrid, McGraw Hill, 2003
- Román Martínez/Elda Quiriga, Estructura de datos: referencias prácticas, México, Thomson Learning, 2001.

## Anexos



Modelo Conceptual del Proyecto.

Fuente: Elaboración propia.

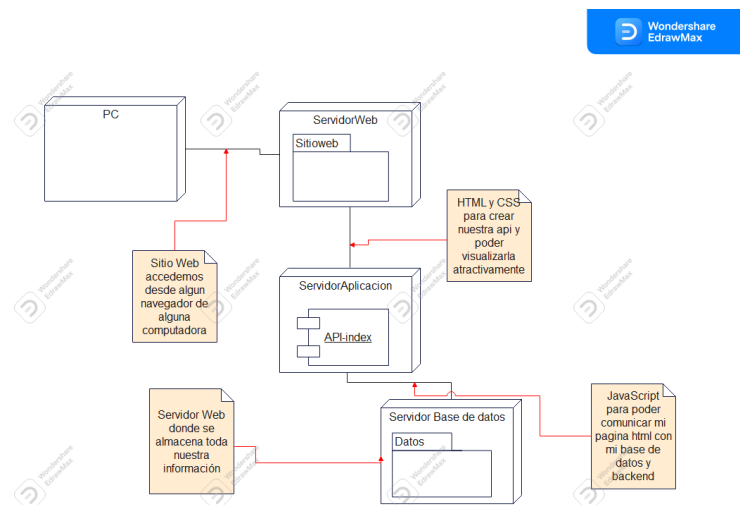


Diagrama de despliegue.

Fuente: Elaboración propia.