

---

## Proyecto 3

---

### Nombre del grupo / estudiante

201905750 – Rony Omar Miguel López

### Resumen

Hoy en día el dominio de temas web es muy útil ya que casi cualquier empresa desea poseer un lugar en internet donde pueda mostrarse y vender su producto, nosotros como programadores debemos ser capaces de adaptarnos a las necesidades del cliente y tener un previo conocimiento de servidores y paginas para poder brindar un buen servicio

### Abstract

*Nowadays the control of web topics is very useful this because almost any company wants to own a place on internet where they can show themselves and sell their product, we as programmers should be able to adapt ourselves to the client's needs and have a previous knowledge about servers and web pages to be able to bring a good service.*

### Palabras clave

frontend  
backend  
Protocolo Http

### Keywords

Frontend  
Backend  
Http Protocol

## Introducción

A continuación se da a conocer una breve explicación de las conexiones entre el FrontEnd y BackEnd de una aplicación esto mediante el uso de dos frameworks llamados Django y Flask al igual que el protocolo Http para comunicarse entre ellos.

## Desarrollo del tema

Para este proyecto se utilizaron dos Frameworks los cuales son Django y Flask, cada uno de ellos lleva un rol muy importante dentro de nuestra aplicación ya que sin ellos no podríamos ser capaces de procesar o mostrar ninguna información.

Empezando por explicar el uso del Framework Django, este se utilizó para poder recrear la parte visual de nuestra aplicación es una de las partes mas importantes de nuestro proyecto ya que por medio de esta nuestro programa interactuará con el usuario haciéndolo de forma clara y amigable para que el usuario no tenga ninguna dificultad al utilizarlo.

El framework django nos permite crear las url que el navegador consumirá para poder realizar las diferentes funciones que nuestro programa sea capaz de ejecutar así mismo nos permite crear y agregar las interfaces que se utilizaran para cada una de esas url's y darles funcionalidad a cada uno de los elementos, además nos permite por medio del protocolo Http por medio de requests y responses (peticiones y respuestas) comunicarnos con nuestro servidor el cual es el que almacenará todos los datos y procesarlos.

El framework Flask nos permite crear un servidor con unas cuantas líneas de código, Flask al igual que Django nos permite crear las url's que serán consumidas por django para así conectar las url's que el cliente ve con las que el servidor usa, este por medio del protocolo Http, este al igual nos permite tener una función para cada url creada diferenciando el método que esta consuma como lo puede ser get, post, put, delete entre otros.

Partes Importantes:

Método Get:

Este método nos permite realizar requests (peticiones) a nuestro servidor o explicándolo de una forma más sencilla realizar desde el frontend una petición a nuestro servidor de algún dato en específico y esperar a que el servidor responda a esta petición con la información esperada.

Método Post:

Este método nos permite realizar envíos de datos a nuestro servidor, este es un poco parecido al método get ya que este también puede esperar que se devuelva información pero esto puede no suceder, este método es muy útil cuando se requiere que alguna información de nuestro cliente sea enviada al servidor para su procesamiento o almacenamiento.

Templates:

Una parte importante de nuestra aplicación es la interfaz que interactuará con el usuario, estas plantillas o templates se almacenan en una carpeta dentro de nuestra aplicación en Django para luego cuando se haga un request de la Url en la que se

encuentra nuestra pagina web esta pueda ser renderizada y mostrada al cliente.

#### Envío de Datos:

El envío de datos es una de las funcionalidades que se espera tener al trabajar con un pagina web para esto como programadores se debe tener una estructura y validación de datos para poder llevar así un orden logístico para procesar los datos y para almacenarlos, en este caso se recibe una estructura xml y al ser enviada al servidor esta es procesada y transformada a formato JSON esto para facilitar aún mas el acceso de los datos y agilizar las validaciones necesarias.

#### Recepción de Datos:

Otra parte muy importante de una página web es la recepción de información la cual es requerida por el cliente y debe ser mostrada en nuestra aplicación, debemos tener cuidado con el formato usado para enviar los datos ya que debemos tener un formato especifico para que así nuestra aplicación pueda leerla tal como es y no haya ninguna confusión con nuestro cliente.

#### Procesamiento de Datos:

Para procesar los datos recibidos en formato xml enviados por el cliente simplemente se decodifican desde nuestro servidor al formato deseado siendo el más usado UTF-8, luego de esto se utilizó una librería llamada xmldict la cual nos permite recibir una cadena de texto con formato xml y transformarla a un formato de diccionario, luego simplemente ese diccionario se transforma a un objeto de tipo Json para así empezar a obtener la información de este

archivo accediendo a cada una de sus claves para obtener la información requerida.

#### Obtención de fecha:

Para obtener la fecha de su etiqueta se utiliza una expresión regular esto es debido a que el texto puede contener más palabras y dentro de ellas puede encontrarse la fecha, la expresión regular nos sirve para encontrar una coincidencia con la forma dada para así asegurarse que la fecha siempre será la única información que extraigamos de esta etiqueta.

Expresión regular: `(([0][1-9]|[1-2][0-9]|3[0-1]))/([0][1-9]|1[1-2])/([0][1-9]|1[0-9]|1[1-9]|1[1-9]|1[2][0]|1[1-9]|2[0][2][0-1]))`

#### Análisis de datos:

Luego de haber obtenido todos los datos de nuestro archivo estos deben ser procesados, se utilizan diferentes funciones para realizar las validaciones una de las que mas resaltan es la validación del Nit esta se hace utilizando una fórmula especifica haciendo unas cuantas sumas y restas y aplicando el modulo de 11 a estas para completar la validación.

#### Lógica de procesos:

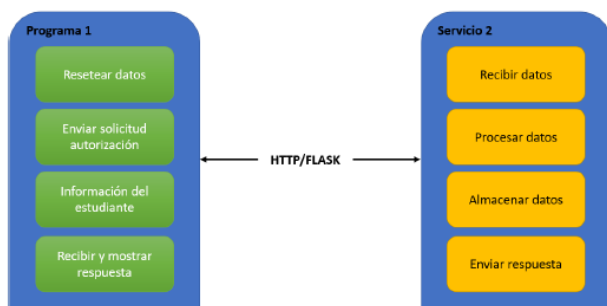
Una parte muy importante de nuestro servidor es la lógica que maneja, como devolver algún error cuando la pagina solicitada no existe o los datos requeridos no fueron enviados, así como la notificación de que todos fue realizado con éxito, esto se puede realizar mediante la utilización de códigos de status enviados al navegador para hacerle saber qué tipo de respuesta se generó, cuál fue la causa y cuál fue la respuesta.

## Conclusiones

Tener una organización y estructura con un plan de ordenamiento es útil en gran manera para permitir el correcto funcionamiento de nuestra arquitectura cliente-servidor, así como también el uso de frameworks facilita mucho más el avance y entendimiento de las diferentes funciones de nuestra aplicación.

## Anexos

### Arquitectura Cliente-Servidor



Envío y recepción de datos.

