

---

## Chapinas S.A

---

**202010770 – Luis Mariano Moreira García**

### Resumen

El proyecto hecho para la importante empresa de chapinas, con quien acordamos el realizar una estructura de tipo cliente-servidor el cual nos permite saber el estado de reacción el cual los usuarios tuvieron ante ciertas publicaciones.

El software en cuestión esta realizado en el mundialmente conocido lenguaje de programación de Python, el cual con sus útiles librerías permitieron un desarrollo bastante más eficiente.

Para su utilización se utilizo el framework de flask para poder trabajar el lado de la rest api, y por parte del cliente se utilizo el potente framework de django.

La transmisión de datos fue hecha directamente desde XML, el cual sirve para comunicarnos con el servidor.

El usuario será capaz de realizar llamadas para poder ejecutar correctamente el software que esta en la web.

### Palabras clave

XML, framework, django, flask, rest API.

### Abstract

The project made for the important chapinas company, with whom we agreed to make a client-server type structure which allows us to know the state of reaction which users had before certain publications.

The software in question is made in the worldwide known Python programming language, which with its useful libraries allowed a much more efficient development.

For its use, the flask framework was used to be able to work on the rest api side, and on the client side, the powerful django framework was used.

The data transmission was done directly from XML, which is used to communicate with the server.

The user will be able to make calls to be able to correctly execute the software that is on the web.

### Keywords

XML, framework, django, flask, rest API.

## Introducción

El poder realizar solicitudes mediante endpoints ha sido de vital importancia para poder desarrollar el proyecto ya que permite el tener un mejor flujo de las funciones a utilizar y así ser capaces de mostrar la información mediante jsons.

Json o también conocidos como Javascript oriented notation, es una manera sencilla de poder comunicar entre el cliente y el servidor, en este caso gracias a flask es posible lograr esta interacción de una manera bastante sencilla y eficiente.

## Desarrollo del tema

La manera en la que se envían los datos es mediante un archivo de tipo XML, con este se comunica con la base de datos local (que es un XML) y así poder desplegar la información de la manera más sencilla posible.

La estructura del archivo XML, debe de obedecer los siguientes parámetros para ello visualizar la imagen 1 de los apéndices.

Algoritmo para mostrar la información deseada

1. Cargar los datos.
2. Se guardan los datos en un objeto.
3. El objeto se guarda en una lista.
4. Se analiza la información.
5. Devuelve la información.

A continuación, se enumeran las listas que se utilizaron para poder realizar el programa

- a. Lista de Mensajes
- b. Lista de mensajes positivos
- c. Lista de mensajes negativos
- d. Lista de mensajes

La manera en la que el usuario puede interactuar directamente con el servidor, obedece a la siguiente página que tiene como endpoint: <http://127.0.0.1:8000/>, se puede cargar una vez se haya iniciado el servidor.

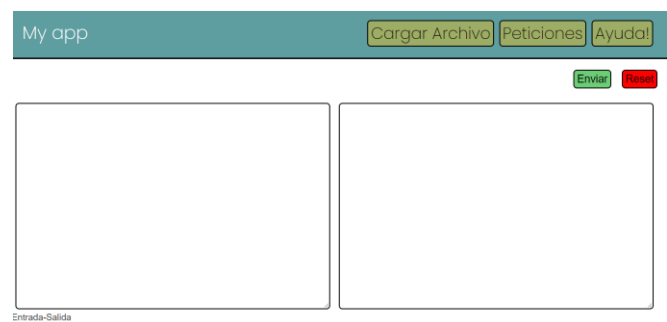


Figura 1. Estructura del frontend

Fuente: elaboración propia, 2022

## Conclusiones

El uso de endpoints y de herramientas de desarrollo web permite hacer un trabajo más eficiente a la hora de poder llevar a cabo un pagina web.

Flask provee una gama de comandos para poder realizar los endpoints que creamos necesarios para poder dar la estructura a nuestra rest api.

Django es un framework que permite unir tanto cliente como servidor de una manera rápida y eficiente.

## Referencias bibliográficas

1. Calzadilla, J. C. F., Herrera, A. N., & Delfino, E. La enseñanza de los arrays estáticos, dinámicos y listas enlazadas¿ cuál usar? Análisis de códigos The teaching of static arrays, dynamics and linked lists. What to use? Code analysis.
2. Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, 20(2), 1-13.
3. Fernández, A. (2013). *Python 3 al descubierto*. Alfaomega Grupo Editor.
4. Legarreta, I. V. (1995). *Diseño de un array bidimensional dinámico implementado mediante listas enlazadas y árboles AVL* (Doctoral dissertation, Universidad de Deusto).
5. Van Rossum, G. (2007, June). Python Programming language. In *USENIX annual technical conference* (Vol. 41, No. 1, pp. 1-36).

## ANEXOS

### Estructura del XML:

```
FECHA: dd/mm/yyyy
Cantidad total de mensajes recibidos: Valor entero que representa el total de mensajes recibidos
Cantidad total de mensajes positivos: Valor entero que representa el total de mensajes positivos recibidos
Cantidad total de mensajes negativos: Valor entero que representa el total de mensajes negativos recibidos
Cantidad total de mensajes neutros: Valor entero que representa el total de mensajes neutros recibidos
Empresa:
  Número total de mensajes que mencionan a Empresa: Valor entero
  Mensajes positivos: Valor entero
  Mensajes negativos: Valor entero
  Mensajes neutros: Valor entero
Servicio #1
  Número total de mensajes que mencionan al servicio: Valor entero
  Mensajes positivos: Valor entero
  Mensajes negativos: Valor entero
  Mensajes neutros: Valor entero
...
Empresa #n:
  Número total de mensajes que mencionan a Empresa o servicio #n: Valor entero
  Mensajes positivos: Valor entero
  Mensajes negativos: Valor entero
  Mensajes neutros: Valor entero
Servicio #1
  Número total de mensajes que mencionan al servicio: Valor entero
  Mensajes positivos: Valor entero
  Mensajes negativos: Valor entero
  Mensajes neutros: Valor entero
...
```

