

---

## PROYECTO 3

---

201900157 – KEVIN ALEXIS LOPEZ CONTRERAS

### Resumen

El proceso de realización de la api esta constituido por diferentes métodos http donde estos son get, put, post, delete, estos métodos conforman la api.

cuando se crea una transformación de consumidor de servicio web REST, debe seleccionar el método HTTP que utiliza el servicio de integración de datos en el mensaje de solicitud. No se puede cambiar el método HTTP después de crear la transformación.

Cada petición individual se envía a un servidor, el cuál la gestiona y responde. Entre cada petición y respuesta, hay varios intermediarios, normalmente denominados proxies (en-US), los cuales realizan distintas funciones, como: gateways o caches.

### Palabras clave

**HTTP:** es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor

**API:** Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones

**DJANGO:** Django es un marco web gratuito y de código abierto basado en Python que sigue el patrón arquitectónico modelo-plantilla-vistas. Lo mantiene la Django Software Foundation, una organización independiente estadounidense establecida como una organización

**DATOS:** información que representan algo en el ámbito dado

**Response:** Cuando se solicita una página, Django crea un `HttpRequest` objeto que contiene metadatos sobre la solicitud. Luego, Django carga la vista apropiada, pasando `HttpRequest` como primer argumento a la función de vista. Cada vista es responsable de devolver un `HttpResponse` objeto.

### **Abstract**

The process of making the api is made up of different http methods where these are get, put, post, delete, these methods make up the api.

When you create a REST Web Service Consumer transformation, you must select the HTTP method that the Data Integration Service uses in the request message. You cannot change the HTTP method after creating the transformation.

Each individual request is sent to a server, which manages and responds to it. Between each request and response, there are several intermediaries, usually called proxies (en-US), which perform different functions, such as: gateways or caches.

### **Keywords**

**HTTP:** it is the name of a protocol which allows us to make a request for data and resources, such as HTML documents. It is the basis for any data exchange on the Web, and a client-server structure protocol

**API:** An API is a set of definitions and protocols used to develop and integrate application software. API stands for Application Programming Interface

**DJANGO:** Django is a free and open source Python-based web framework that follows the model-template-views architectural pattern. Maintained by the Django Software Foundation, an independent American organization established as an organization

**DATA:** information that represents something in the given scope

Response: When a page is requested, Django creates an `HttpRequest` object that contains metadata about the request. Django then loads the appropriate view, passing `HttpRequest` as the first argument to the view function. Each view is responsible for returning an `HttpResponse` object.

## Introducción

La api se realizo con Flask es un marco de micro web escrito en Python. Se clasifica como un microframework porque no requiere herramientas o bibliotecas particulares. No tiene una capa de abstracción de base de datos, validación de formularios ni ningún otro componente donde las bibliotecas de terceros preexistentes y al utilizar Django se hizo mas intuitiva la interface ya que se realizo una cierta cantidad de vista para que el usuario lo entendiera de una forma adecuada, utilizando los protocolos http ya que con estos se puede trasferir información del lado del servidor al cliente y viceversa

## Desarrollo del tema

es el protocolo de transmisión de información de la World Wide Web, es decir, el código que se establece para que el computador solicitante y el que contiene la información solicitada puedan “hablar” un mismo idioma a la hora de transmitir información por la red.

Con el http se establecen criterios de sintaxis y semántica informática (forma y significado) para el establecimiento de la comunicación entre los diferentes elementos que constituyen la arquitectura web: servidores, clientes, proxies

como se ha dicho, es un lenguaje que media entre las peticiones del cliente y las respuestas del servidor en la Internet, para permitir una comunicación fluida y en un mismo “lenguaje”. Este protocolo establece las pautas a seguir, los métodos de petición (llamados “verbos”) y cuenta con cierta flexibilidad para incorporar nuevas peticiones y funcionalidades, en especial a medida que se avanza en sus versiones.

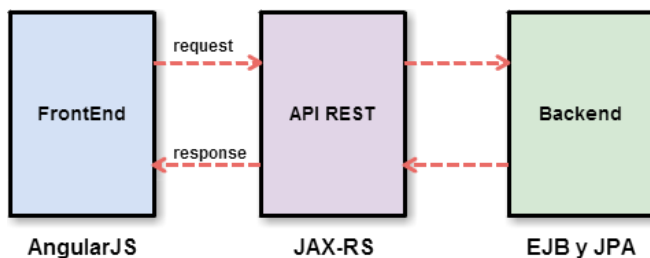
Considerando que la Internet es poco más que una compleja red de intercambio de información entre computadores a distancia, este tipo de herramientas digitales son clave en establecer las bases para ordenar y facilitar la transmisión de la información.

el proceso de transferencia es simple se utilizan dos tipos de procesos que estos se llaman backend y frontend estos procesos se hacen peticiones y el servidor responde con los datos solicitados los filtros creados en la api se analizaron detenidamente para que no se filtre un dato no deseado y obtener los datos correctos, al carga el archivo se analizó dato por dato creado un autómata determinista para poder analizar de una forma correcta los datos.

Una api Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

Así pues, podemos hablar de una API como una especificación formal que establece cómo un módulo de un software se comunica o interactúa con otro para cumplir una o muchas funciones. Todo dependiendo de las aplicaciones que las vayan a utilizar, y de los permisos que les dé el propietario.

*Proceso de la api*



Fuente: elaboración propia 2021.

El agente del usuario, es cualquier herramienta que actúe en representación del usuario. Esta función es realizada en la mayor parte de los casos por un navegador Web. Hay excepciones, como el caso de programas específicamente usados por desarrolladores para desarrollar y depurar sus aplicaciones.

El navegador es **siempre** el que inicia una comunicación (petición), y el servidor nunca la comienza (hay algunos mecanismos que permiten esto, pero no son muy habituales).

Para poder mostrar una página Web, el navegador envía una petición de documento HTML al servidor. Entonces procesa este documento, y envía más peticiones para solicitar scripts, hojas de estilo (CSS), y otros datos que necesite (normalmente vídeos y/o imágenes). El navegador, une todos estos documentos y datos, y compone el resultado final: la página Web. Los scripts, los ejecuta también el navegador, y también pueden generar más peticiones de datos en el tiempo, y el navegador, gestionará y actualizará la página Web en consecuencia.

## Conclusiones

Se implementó un api por medio del lenguaje Python utilizando como framework Flask y django.

Se utilizó el paradigma orientado a objetos para resolver toda la lógica de la api

Se utilizo xml como archivo de carga principal y poder procesarlo de una forma correcta

Se utilizaron expresiones regulares para analizar el archivo xml y tener un resultado exitoso

### **Referencias bibliográficas**

Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.