

---

## Proyecto 3 IPC2 Primer semestre, ciclo 2021

---

201701059 – Luis Amilcar Morales Xón

### Resumen

Un proyecto desarrollado en Python que evalúa los conocimientos y la implementación de temas en específico, tales como: Desarrollo web con el framework Django, implementación de una API con Flask y manejo de archivos XML. En una época dada las circunstancias actuales, la migración de una empresa o negocio al ámbito web es vital y clave para la supervivencia o desarrollo de la empresa o negocio. Es por eso que el desarrollo web está teniendo su auge y todavía tiene mucho que recorrer todavía para llegar a la cúspide. Muestra de ello es la aparición e introducción de frameworks más a menudo o bien el desarrollo o refinamiento de los que ya existían. Una página web además de ofrecer mayor alcance en cuanto a ventas y varios aspectos de mercadotecnia, también ofrece un mejor control sobre la información y datos que la empresa maneja.

### Palabras clave

Framework  
API  
Rutas  
Frontend  
Backend

### Abstract

*A project developed in Python that assesses the knowledge and implementation of specific topics, such as: Web development with the Django framework, implementation of an API with Flask and handling of XML files.*

*In an age given the current circumstances, the migration of a company or business to the web field is vital and key to the survival or development of the company or business. That is why web development is having its heyday and still has a long way to go to get to the top. Proof of this is the appearance and introduction of frameworks more often or the development or refinement of those that already existed. A website, in addition to offering greater scope in terms of sales and various marketing aspects, also offers better control over the information and data that the company handles.*

### Keywords

Framework  
API  
Routes  
Frontend  
Backend

## Introducción

Un proyecto que implementa una API rest, el cual nos permite interactuar con la aplicación web desde distintos clientes (Computadora, teléfono, Tablet, etc.). La API rest es la encargada de proveer acceso a las distintas funciones de un determinado software y la forma en la que interactuamos con ella es por medio del frontend. La interfaz gráfica que utilizará el usuario para manejar el software viene siendo el frontend, que consta básicamente de HTML's. La forma en como podemos juntar la API rest (backend) con la interfaz grafica de usuario (frontend) lo hacemos por medio del Framework conocido como Django.

## Desarrollo del tema

EL proyecto se realizó en el lenguaje Python y consta con un entorno gráfico desarrollado en HTML con el que logramos la manipulación de los datos a partir de una fuente XML que es procesada por la API, las opciones que se manejan en el programa son los siguientes:

- **Cargar Archivo:** Se desplegará una pantalla para gestionar la carga de los archivos de entrada con extensión .xml con uno o varios eventos.
- **Peticiones:** En este apartado se tiene las siguientes opciones:
  - **Consultar Datos:** Al seleccionar esta opción se deben de consultar los datos almacenados en el archivo estadísticas.xml y se mostrarán los datos en el recuadro de texto de salida
  - **Filtrar información por fecha y usuario que reporta:** Al seleccionar esta opción se podrá elegir la fecha por la cual se requiere filtrar y se debe de mostrar gráficamente los usuarios que reportaron errores en esa fecha.
  - **Filtrar por fecha y código de error:** Al seleccionar esta opción se podrá ingresar un código de error, se deberá presentar

gráficamente el total de mensajes que contienen ese código por cada fecha en donde se hizo el reporte de dicho error

- **Ayuda:** Desplegará 2 opciones, una para visualizar información del estudiante y otra para visualizar la documentación del programa.
- **Botón Enviar:** Envió los eventos del recuadro de texto a la Api para su posterior procesamiento.
- **Botón Reset:** Este botón mandará la instrucción a la Api para devolver al estado inicial la Api, es decir sin datos.

## ¿Qué es una API Rest?

Antes de introducir el concepto de Rest API, es necesario explicar qué es una API (Interfaz de programación de aplicaciones). Es un conjunto de requisiciones que permite la comunicación de datos entre aplicaciones.

Para eso, la API utiliza requisiciones HTTP responsables de las operaciones básicas necesarias para la manipulación de datos.

Las principales solicitudes son:

- POST: crea datos en el servidor;
- GET: lectura de datos en el **host**;
- DELETE: borra la información;
- PUT: registro de actualizaciones.

Rest, que es la abreviación de Representational State Transfer, es un conjunto de restricciones que se utilizan para que **las solicitudes HTTP cumplan con las directrices definidas en la arquitectura.**

Básicamente, las restricciones determinadas por la arquitectura Rest son:

- **Cliente-servidor:** las aplicaciones existentes en el servidor y el cliente deben estar separadas.

- **Sin estado:** las requisiciones se realizan de forma independiente, es decir, cada una ejecuta solo una determinada acción.
- **Caché:** la API debe utilizar la caché para evitar llamadas recurrentes al servidor.
- **Interfaz uniforme:** agrupa otros cuatro conceptos en los que se determina que los recursos deben ser identificados, la manipulación de los recursos debe ser a través de la representación, con mensajes autodescriptivos y utilizando enlaces para navegar por la aplicación.

Luego, cuando se habla de Rest API, significa utilizar una API para acceder a aplicaciones back-end, de manera que **esa comunicación se realice con los estándares definidos por el estilo de arquitectura Rest.**

## ¿Qué tipos de API existen?

Básicamente, existen tres tipos de APIs:

- **Privados:** que se utilizan internamente entre las aplicaciones de una empresa, es decir, localmente.
- **Socios:** que se utilizan entre socios comerciales o para permitir la integración entre diferentes softwares.
- **Público:** que se puede utilizar libremente. A menudo, la empresa los pone a disposición para que los desarrolladores puedan integrarse con otras aplicaciones.

## ¿Para qué sirve API Rest?

Existe una gran variación en las formas en que se utilizan los API. Las redes sociales, por ejemplo, proporcionan un API que se puede utilizar en otros sitios para recuperar información de una página.

Existen varios plugins de WordPress que acceden a las redes sociales a través de ellos y transforman el resultado de esta interacción en pequeñas visualizaciones del estado actual de la página correspondiente.

Así, si un usuario quiere darle like a la página, por ejemplo, no es necesario que abandone el sitio original para esta acción.

Al hacer clic en el botón Me Gusta, hay una llamada al API para completar esta operación. Para que eso sea posible, las redes sociales proporcionan un token con la debida autorización para que API tenga acceso a la información.

Al igual que las redes sociales, los APIs también se utilizan en los sitios de ecommerce para acceder a los intermediarios de pago y completar las transacciones de compra.

Por lo tanto, **el API sirve para comunicarse entre aplicaciones para intercambiar informaciones de forma rápida y segura.**

## ¿Qué importancia tiene la API Rest para el sitio web de la empresa?

Uno de los factores más importantes al utilizar un sitio web con Rest API es la facilidad de comunicación con otras aplicaciones. Las interfaces permiten agregar funcionalidades o información al sitio web de una manera simple, rápida y segura.

De esa forma, es posible crear sitios web o aplicaciones en la nube para acceder a datos de sistemas más robustos, como ERPs, sea cual sea la plataforma desde la que se originan los datos.

Esa función **permite a las empresas acceder a informaciones actualizadas mucho más rápidamente.**

## ¿Cuáles son las ventajas de utilizar una

### API Rest?

Las características de la arquitectura Rest API brindan una serie de beneficios a las aplicaciones que utilizan este concepto.

Ve algunas a continuación.

### Separación entre cliente y servidor

Una de las ventajas de utilizar el modelo API Rest es la separación entre las aplicaciones de front-end y back-end.

Esto es importante para proteger el almacenamiento de datos, ya que no existe un tratamiento de las reglas comerciales, es decir, **solo se intercambia informaciones** sea para recuperar datos, o para insertar o eliminar nuevos registros.

### Más visibilidad, confiabilidad y

#### escalabilidad

Debido a la separación cliente / servidor, hay mucha más facilidad durante el desarrollo de la aplicación. Esto se debe a que se puede escalar fácilmente, ya que no hay dificultad para vincular recursos.

Como cada solicitud se realiza de forma única e independiente, es posible cambiar una solicitud a otro DNS, sin interferir con la aplicación.

En otras palabras, **una API Rest permite que la aplicación acceda a bases de datos desde diferentes servidores**, lo que a menudo es importante para el desarrollo en aplicaciones grandes. Por lo tanto, su uso

garantiza una mayor visibilidad y credibilidad a la hora de utilizar estos recursos.

### Multiplataforma

Las requisiciones HTTP realizadas en API Rest devuelven datos en formato JSON. Cabe mencionar que existen otros posibles formatos de retorno, como XML, sin embargo, JSON es el más utilizado. Por lo tanto, la mayoría de los sitios que trabajan con este modelo reciben este formato de datos.

**Esta característica es fundamental para el desarrollo de aplicaciones multiplataforma.** Eso se debe a que, al recibir los datos en este formato, la camada front-end de la aplicación es capaz de realizar el tratamiento adecuado para mostrar los resultados según el tipo de dispositivo utilizado.

El uso de **Rest API es importante para agregar varias funciones al sitio**. Sus características permiten la integración con diferentes aplicaciones; entre ellos, redes sociales y sistemas de pago.

Por eso, es una tecnología que garantiza una mayor fiabilidad y escalabilidad, además de facilitar el desarrollo de aplicaciones multiplataforma.

## ¿Qué es Django?

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

Django te ayuda a escribir software que es:

## Completo

Django sigue la filosofía "Baterías incluidas" y provee casi todo lo que los desarrolladores quisieran que tenga "de fábrica". Porque todo lo que necesitas es parte de un único "producto", todo funciona a la perfección, sigue principios de diseño consistentes y tiene una amplia y actualizada documentación.

## Versátil

Django puede ser (y ha sido) usado para construir casi cualquier tipo de sitio web — desde sistemas manejadores de contenidos y wikis, hasta redes sociales y sitios de noticias. Puede funcionar con cualquier framework en el lado del cliente, y puede devolver contenido en casi cualquier formato (incluyendo HTML, RSS feeds, JSON, XML, etc.). ¡El sitio que estás leyendo actualmente está basado en Django!

Internamente, mientras ofrece opciones para casi cualquier funcionalidad que desees (distintos motores de base de datos, motores de plantillas, etc.), también puede ser extendido para usar otros componentes si es necesario.

## Seguro

Django ayuda a los desarrolladores evitar varios errores comunes de seguridad al proveer un framework que ha sido diseñado para "hacer lo correcto" para proteger el sitio web automáticamente. Por ejemplo, Django, proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar informaciones de sesión en cookies donde es vulnerable (en lugar de eso las cookies solo contienen una clave y los datos se almacenan en la base de datos) o se almacenan directamente las contraseñas en un hash de contraseñas.

*Un hash de contraseña es un valor de longitud fija creado al enviar la contraseña a una cryptographic hash function. Django puede validar si la contraseña ingresada es correcta enviándola a través de una función hash y comparando la salida con el valor hash almacenado. Sin embargo debido a la naturaleza "unidireccional" de la función, incluso si un valor hash almacenado se ve comprometido es difícil para un atacante resolver*

*la contraseña original.*

Django permite protección contra algunas vulnerabilidades de forma predeterminada, incluida la inyección SQL, scripts entre sitios, falsificación de solicitudes entre sitios y clickjacking (consulte Seguridad de sitios web para obtener más detalles sobre dichos ataques).

## Escalable

Django usa un componente basado en la arquitectura "shared-nothing" (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario). Teniendo en cuenta una clara separación entre las diferentes partes significa que puede escalar para aumentar el tráfico al agregar hardware en cualquier nivel: servidores de cache, servidores de bases de datos o servidores de aplicación. Algunos de los sitios más concurridos han escalado a Django para satisfacer sus demandas (por ejemplo, Instagram y Disqus, por nombrar solo dos).

## Mantenible

El código de Django está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. En particular, utiliza el principio No te repitas "Don't Repeat Yourself" (DRY) para que no exista una duplicación innecesaria, reduciendo la cantidad de código. Django también promueve la agrupación de la funcionalidad relacionada en "aplicaciones" reutilizables y en un nivel más bajo, agrupa código relacionado en módulos (siguiendo el patrón Model View Controller (MVC)).

## Portable

Django está escrito en Python, el cual se ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows y Mac OS X. Además, Django cuenta con el respaldo de muchos proveedores de alojamiento web, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios de Django.

## Conclusiones

Es imprescindible entender el comportamiento de una estructura abstracta para poder manipularlo y principalmente poder representarlo dentro de un lenguaje de programación ya que esto permitirá poder modelar cualquier información que se presente en el mundo real y convertirlo en datos que la computadora sea capaz de entender, así como el programador también es capaz de hacerlo

## Referencias bibliográficas

de Souza, I., 2020. *API Rest: ¿qué es y cómo funciona ese recurso? [con ejemplos]*. [online] Rock Content - ES. Disponible en: <<https://rockcontent.com/es/blog/api-rest/>> [Recuperado el 4 de mayo de 2021].

## Anexos

Diagrama de una aplicación web con una API rest y base de datos

