

---

## Proyecto 3

---

201900810 – Jorge Antonio Pérez Ordóñez

### Resumen

Durante el desarrollo de este proyecto se creó un API a la cual se le pueden hacer peticiones http y cada método realiza un procedimiento distinto para llevar la cuenta de las estadísticas en un archivo XML. Para llevar a cabo las peticiones al API se realizó un proyecto en Django en el que se pudiera cargar archivos de entrada con extensión XML para que luego mandarlas por el API. También se puede verificar las estadísticas filtrados por fecha o por error gráficamente utilizando el paquete de Python matplotlib. Al mismo tiempo el API puede ser usado por cualquier otro cliente, por ejemplo, la aplicación Postman, la cual sirve para probar APIs.

### Abstract

*Throughout the development of this project, an API was created to which http requests can be made and each method performs a different procedure to keep track of the statistics in an XML file. To carry out the requests to the API, a project was carried out in Django in which input files with an XML extension could be loaded so that they could then be sent through the API. You can also check statistics filtered by date or by error graphically using the Python package matplotlib. At the same time the API can be used by any other client, for example, the Postman application, which is used to test APIs.*

### Palabras clave

API

Peticiones

Django

Cliente

### Keywords

API

Requests

Django

Client

## Introducción

Cuando una página web necesita acceder a datos de otras aplicaciones, antes se conectaban por métodos inseguros debido a que el público podía tener acceso a estas aplicaciones sin el consentimiento de nadie. Un API se encarga de eliminar esta inseguridad ya que es un intermedio entre aplicaciones que recibe y responde información relevante a la cual solo se puede tener acceso si se tiene las credenciales correctas. Un framework como Django que sirve para crear páginas web, puede hacer consultas a un API si lo desea para mandar y obtener información sobre todo sobre una base de datos, pero también se puede hacer consultas para comunicar con otro tipo de aplicaciones.

## Desarrollo del tema

### API

API son las siglas de Application Programming Interface (Interfaz de programación de aplicaciones). En términos básicos, las API son un conjunto de funciones y procedimientos que permiten la creación de aplicaciones que acceden a datos y características de otras aplicaciones, servicios o sistemas operativos.

Las API facilitan el desarrollo de un programa informático al proporcionar todos los componentes básicos, que luego el programador ensambla. Las interfaces de programación de aplicaciones constan de dos componentes:

Especificación técnica que describe las opciones de intercambio de datos entre soluciones con la especificación realizada en forma de una solicitud de protocolos de procesamiento y entrega de datos

Interfaz de software escrita según la especificación que la representa

El software que necesita acceder a información o funcionalidad (es decir, una ruta desde el punto A al punto B en un mapa basado en la ubicación de un usuario) desde otro software, llama a su API mientras especifica los requisitos de cómo se deben proporcionar los datos y funcionalidad. El otro software devuelve datos y funcionalidad solicitados por la aplicación anterior.

### REST Server

REST, o REpresentational State Transfer (transferencia de estado representacional), es un estilo arquitectónico para proporcionar estándares entre sistemas informáticos en la web, lo que facilita que los sistemas se comuniquen entre sí. Los sistemas que cumplen con REST, a menudo llamados sistemas RESTful, se caracterizan por la forma en que no tienen estado y separan las preocupaciones del cliente y del servidor. Analizaremos qué significan estos términos y por qué son características beneficiosas para los servicios en la Web.

En el estilo arquitectónico REST, la implementación del cliente y la implementación del servidor se pueden realizar de forma independiente sin que cada una se sepa. Esto significa que el código del lado del cliente se puede cambiar en cualquier momento sin afectar el funcionamiento del servidor, y el código del lado del servidor se puede cambiar sin afectar el funcionamiento del cliente.

Siempre que cada lado sepa qué formato de mensajes enviar al otro, se pueden mantener modulares y separados. Al separar las preocupaciones de la interfaz de usuario de las preocupaciones sobre el almacenamiento de datos, mejoramos la flexibilidad de la interfaz en todas las plataformas y mejoramos la escalabilidad al simplificar los componentes del servidor. Además, la separación permite que cada

componente tenga la capacidad de evolucionar de forma independiente.

Al utilizar una interfaz REST, diferentes clientes acceden a los mismos puntos finales REST, realizan las mismas acciones y reciben las mismas respuestas.

### Django

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.

Django puede ser 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).

Internamente, mientras ofrece opciones para casi cualquier funcionalidad que desees como bases de datos o plantillas, también puede ser extendido para usar otros componentes si es necesario.

Dentro de los proyectos de Django se trabaja con los siguientes componentes de tal manera que el proyecto funcione:

- **URLs:** Aunque es posible procesar peticiones de cada URL individual vía una función individual, es mucho más sostenible escribir una función de visualización separada para cada recurso. Se usa un mapeador URL para redirigir las peticiones HTTP a la vista apropiada basándose en la URL de la petición. El mapeador URL se usa para redirigir las

peticiones HTTP a la vista apropiada basándose en la URL de la petición. El mapeador URL puede también emparejar patrones de cadenas o dígitos específicos que aparecen en una URL y los pasan a la función de visualización como datos.

- **Vistas:** Una vista es una función de gestión de peticiones que recibe peticiones HTTP y devuelve respuestas HTTP. Las vistas acceden a los datos que necesitan para satisfacer las peticiones por medio de modelos, y delegan el formateo de la respuesta a las plantillas ("templates").
- **Modelos:** Los Modelos son objetos de Python que definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar (añadir, modificar y borrar) y consultar registros en la base de datos.
- **Plantillas:** una plantilla (template) es un fichero de texto que define la estructura o diagrama de otro fichero (tal como una página HTML), con marcadores de posición que se utilizan para representar el contenido real. Una vista puede crear dinámicamente una página usando una plantilla, rellenandola con datos de un modelo. Una plantilla se puede usar para definir la estructura de cualquier tipo de fichero; ¡no tiene porqué ser HTML!

### Conclusiones

Las API son aplicaciones que actúan como intermediario entre aplicaciones para mandar y obtener información importante necesaria para el funcionamiento un sistema.

Un REST Server es un estilo de programa organizado de tal manera que un cliente pueda realizar peticiones

y dependiendo de lo que quiera el cliente, el servidor le responderá con la información adecuada.

Django es un framework programado con Python para hacer sitios web completos, el cual usa un sistema de modelo vista controlador y puede actuar como cliente al usar un API.

## Referencias Bibliográficas

- BBVA API Market. (23 de Marzo de 2016). bbvaapimarket.com. Obtenido de bbvaapimarket.com:  
<https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- Korsun, J. (03 de Marzo de 2021). djangostars.com. Obtenido de djangostars.com:  
<https://djangostars.com/blog/why-we-use-django-framework/>
- Singtel, P. (18 de Junio de 2019). altexsoft.com. Obtenido de altexsoft.com:  
<https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/#>