
PROYECTO 3 (SOLUCIONES CHAPINAS)

Nombre del grupo

202001482 – DIEGO PAOLO MEDINA GARCIA (Coordinador)

Resumen

Se realizó una API con el framework Flask del lenguaje de programación Python para poder ser consumido por cualquier tecnología “frontend”, para el frontend se eligió el framework de Django.

Abstract

An API was created with the Flask framework of the Python programming language in order to be consumed by any frontend technology; the Django framework was chosen for the frontend.
El abstract y las keywords deben abarcar solamente esta columna.

Palabras clave

Framework

Flask

API

Django

Keywords

Framework

Flask

API

Django

Introducción

Tecnologías Chapinas, S.A. desarrollar una herramienta con la capacidad para elaborar facturas detalladas por los servicios de infraestructura en la nube proporcionados a sus clientes. Estructura de Tecnología Cloud por Tecnologías Chapinas, S.A. Incluye la creación de configuraciones de infraestructura para recolectar los recursos requeridos las empresas pueden crear arquitecturas de implementación de aplicaciones que Pedir. Como hay muchos recursos técnicos (cores1, memoria, almacenamiento secundario, sistemas operativos, bases de datos, etc.) y debe tener poder configurar según la carga de trabajo para la que se configurará la infraestructura, Chapinas Technologies, S.A. Ha creado categorías como: Máquinas virtuales para desarrollo/testing, máquinas virtuales baratas, etc. Cada una categoría tiene un nombre, una descripción y una definición de carga de trabajo las configuraciones en esta categoría serán personalizadas.

Desarrollo del tema

Pequeña descripción

¿Qué es un framework?

Actualmente en el desarrollo moderno de aplicaciones web se utilizan distintos Frameworks que son herramientas que nos dan un esquema de trabajo y una serie de utilidades y funciones que nos facilita y nos abstrae de la construcción de páginas web dinámicas.

En general los Frameworks están asociados a lenguajes de programación (Ruby on Rails (Ruby),

Symphony (PHP)), en el mundo de Python el más conocido es Django pero Flask es una opción que quizás no tenga una curva de aprendizaje tan elevada pero nos posibilita la creación de aplicaciones web igual de complejas de las que se pueden crear en Django.

Ventajas de usar un Framework

1. Proporciona una estructura de proyecto, es decir, todas las Apps que estén construidas con Flask van a tener el mismo elemento y los mismos ficheros.
2. Facilita la colaboración.
3. Es fácil encontrar bibliotecas adaptadas al Framework

¿Por qué usar Flask?

- a. Flask es un “micro” Framework: Para desarrollar una App básica o que se quiera desarrollar de una forma ágil y rápida Flask puede ser muy conveniente, para determinadas aplicaciones no se necesitan muchas extensiones y es suficiente.
- b. Incluye un servidor web de desarrollo: No se necesita una infraestructura con un servidor web para probar las aplicaciones sino de una manera sencilla se puede correr un servidor web para ir viendo los resultados que se van obteniendo.
- c. Tiene un depurador y soporte integrado para pruebas unitarias: Si tenemos algún error en el código que se está construyendo se puede depurar ese error y se puede ver los valores de las variables. Además está la posibilidad de integrar pruebas unitarias.

- d. Es compatible con Python3.
- e. Es compatible con wsgi: Wsgi es un protocolo que utiliza los servidores web para servir las páginas web escritas en Python.
- f. Buen manejo de rutas: Cuando se trabaja con Apps Web hechas en Python se tiene el controlador que recibe todas las peticiones que hacen los clientes y se tienen que determinar que ruta está accediendo el cliente para ejecutar el código necesario.
- g. Soporta de manera nativa el uso de cookies seguras.
- h. Se pueden usar sesiones.
- i. Flask no tiene ORMs: Pero se puede usar una extensión.
- j. Sirve para construir servicios web (como APIs REST) o aplicaciones de contenido estático.
- k. Flask es Open Source y está amparado bajo una licencia BSD.
- l. Buena documentación, código de GitHub y lista de correos.

Algunas extensiones que podemos usar en Flask son:

flask-script: Permite tener un comando de la línea de comando para manejar la aplicación.

flask-Bootstrap: Hojas de estilo para la página.

flask-WTF: Sirve para generar formularios de HTML con clases y objetos.

flask-Sqlalchemy: Sirve para poder generar el modelo de datos.

flask-login: Sirve para la autenticación de usuario y contraseña.

¿Qué es Django?

Django es un marco web avanzado escrito en Python que utiliza el patrón arquitectónico del controlador de vista de modelo (MVC). Django se creó en un entorno de sala de redacción de rápido movimiento y su objetivo clave es facilitar el desarrollo de sitios web complicados basados en bases de datos. Este marco web se desarrolló inicialmente para The World Company para administrar algunos de sus sitios orientados a las noticias. En julio de 2005, se lanzó al público bajo una licencia BSD.

Django está disponible como un marco web de código abierto y utiliza ampliamente Python para crear archivos, configuraciones y modelos de datos. Está diseñado para abordar dos desafíos principales: los rigurosos requisitos de los desarrolladores web altamente experimentados y los intensos plazos de una sala de redacción. Django se concentra más en la automatización siempre que sea posible y se apegue al principio de "no te repitas".

Django enfatiza lo siguiente:

- a. Conectividad y reutilización de componentes.
- b. Desarrollo rápido
- c. El principio de no repetición

Ventajas de Django

- a. Cuenta con un sistema de autenticación de usuarios.
- b. Manejo de versiones que permite una distribución simple de actualizaciones.
- c. Ofrece un gran rendimiento y flexibilidad, pudiendo escalar proyectos de forma sencilla.
- d. Trabajar bajo un patrón MVC (Modelo Vista Controlador), lo que permite un desarrollo ágil y reutilizable.

- e. Incorpora una amplia variedad de paquetes de librerías (más de 4000).
- f. Dispone de una inmensa comunidad de usuarios en internet.
- g. Incluye opciones de protección para las aplicaciones, por ejemplo, contra ataques de SQL injection o ataques XSS (cross site scripting).
- h. Proporciona una estructura de código autogenerado.
- i. Cuenta con panel de administración para bases de datos.

Desventajas de Django

Entre los inconvenientes que se pueden encontrar al utilizar Django para desarrollar aplicaciones web con Python podemos destacar su extensa documentación, que a veces puede llegar a ser confusa y dispersa. También, puede resultar difícil a la hora de realizar API Rest (estilo de arquitectura software para crear aplicaciones web respetando el protocolo HTTP) o resultar complejo cuando se trabaja con sockets.

Desventajas de Flask

Flask no contiene librerías integradas, por lo que es necesario descargarlas para poder utilizar distintas funciones. Se trata de un entorno que genera dificultades a la hora de realizar migraciones o pruebas unitarias. También es una desventaja tener que recurrir a un mapeo de objetos relacionales (ORM) externo para conectar con bases de datos. El sistema de autenticación de usuarios de Flask es muy básico, algo que se echa de menos en este entorno de desarrollo Python.

Ejemplos de los Usos de Flask

El uso de este framework de Python se ha visto incrementado durante los últimos años gracias a su sencillez y al control que proporciona sobre los proyectos desarrollados con este lenguaje de programación.

Podemos destacar entre las webs más reconocidas desarrollada con Python utilizando el entorno de desarrollo Flask a: Netflix, Red Hat, Reddit, Lyft o Airbnb.

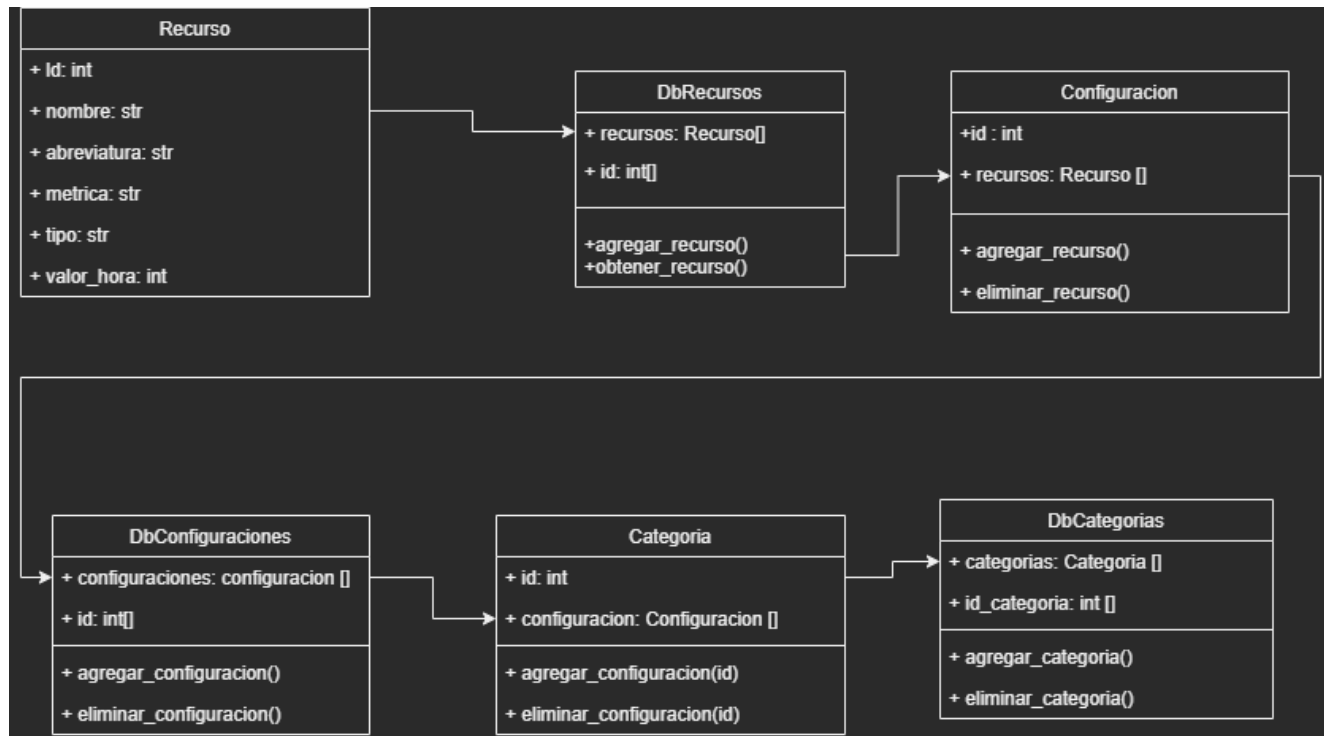
Al realizar esta comparativa entre Flask y Django hemos visto cuáles son los beneficios e inconvenientes de utilizar ambos frameworks a la hora de afrontar proyectos con el lenguaje de programación Python.

Ambos entornos de desarrollo son utilizados por proyectos web importantes que soportan mucho tráfico web. A la hora de decantarse por uno u otro hay que analizar sus pros y contras, y así decidir cuál de ellos se adapta mejor a las necesidades de cada proyecto web.

Conclusiones

Luego de aprender un lenguaje de programación, es importante conocer los diferentes frameworks y librerías que nos ofrece, ver cual se adapta mejor a nuestras necesidades y aprenderlo. Ya que esta es una gran ventaja al momento de realizar nuestros trabajos.

ANEXOS



Referencias bibliográficas

Aguilar, Q. (2020, julio 30). Comparativa de Flask vs Django. Ilimit.com. <https://www.ilimit.com/blog/flask-vs-django/>

Techopedia, W. is D.-D. (2012, agosto 30). Django. Techopedia.com; What is Django? - Definition from Techopedia.<https://www.techopedia.com/definition/28227/django>

Qué es Flask. (2017, noviembre 17). Openwebinars.net. <https://openwebinars.net/blog/que-es-flask/>