

# Contenidos teóricos

Sitio:

[Agencia de Aprendizaje a lo largo de la Vida](#)

Curso:

PYTHON - BACK END 2022

Libro:

Contenidos teóricos

Imprimido por:

Tomas Agustin Friz

Día:

lunes, 27 de junio de 2022, 19:14

# Tabla de contenidos

- 1. ¿Qué es Flask?
- 2. ¿Qué es un Framework?
- 3. ¿Por qué usar Flask?
- 4. Extensiones de Flask

# 1. ¿Qué es Flask?

## Qué es Flask

En la actualidad existen muchas opciones para crear páginas web y muchos lenguajes, y en este caso Flask nos permite crear de una manera muy sencilla aplicaciones web con Python.

Flask es un “*micro*” Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC.

La palabra “*micro*” no designa a que sea un proyecto pequeño o que nos permita hacer páginas web pequeñas sino que al instalar Flask tenemos las herramientas necesarias para crear una aplicación web funcional pero si se necesita en algún momento una nueva funcionalidad hay un conjunto muy grande extensiones (plugins) que se pueden instalar con Flask que le van dotando de funcionalidad.

De principio en la instalación no se tienen todas las funcionalidades que se pueden necesitar pero de una manera muy sencilla se pueden extender el proyecto con nuevas funcionalidades por medio de plugins.

El patrón MVC es una manera o una forma de trabajar que permite diferenciar y separar lo que es el modelo de datos (los datos que van a tener la App que normalmente están guardados en BD), la vista (página HTML) y el controlador (donde se gestiona las peticiones de la app web).

## Flask, el microframework con el que podemos hacer grandes proyectos

Según palabras de su propio autor, Flask es un microframework para Python basado en Werkzeug que permite crear aplicaciones web de todo tipo rápidamente.

Muchas son las comparaciones que se pueden encontrar sobre distintos frameworks web basados en Python. Normalmente se dice que Django es mejor para proyectos de cierta envergadura, mientras que Flask se debe usar para prototipos, crear APIs de forma ágil o pequeñas aplicaciones.

En realidad siempre depende del proyecto por lo que como un primer acercamiento al uso de Frameworks para el desarrollo de un sistema integral utilizaremos Flask. En la próxima unidad podrás adentrarte a otro Framework muy utilizado actualmente en el ámbito laboral como puede ser Django.

Flask es un framework completo que permite desarrollar todo tipo de aplicaciones, complejas y sencillas.

Flask es muy versátil y sí, es cierto que puedes hacer prototipos y APIs muy, muy rápido (en función de tus conocimientos de Python) también es verdad que se pueden desarrollar proyectos complejos haciendo uso del mismo y los resultados son igual de satisfactorios.

Fuente:

<https://openwebinars.net/blog/que-es-flask/>

<https://j2logo.com/tutorial-flask-espanol/>

<https://riptutorial.com/Download/flask.pdf>

<https://wiki.python.org/moin/WebFrameworks>

## 2. ¿Qué es un Framework?

### ¿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. Por ejemplo, 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 y además nos posibilita la creación de aplicaciones web igual de complejas de las que se pueden crear en Django.

### Ventajas de usar un Framework

Proporciona una estructura del proyecto, es decir, todas las Apps que estén construidas con Flask van a tener los mismos elementos y los mismos ficheros.

Facilita la colaboración.

Es fácil encontrar bibliotecas adaptadas al Framework.

### Frameworks en Python

En la Wiki de Python se muestra un listado con los distintos Frameworks compatibles: <https://wiki.python.org/moin/WebFrameworks>

Fuente:

<https://openwebinars.net/blog/que-es-flask/>

<https://j2logo.com/tutorial-flask-espanol/>

<https://riptutorial.com/Download/flask.pdf>

<https://wiki.python.org/moin/WebFrameworks>

## 3. ¿Por qué usar Flask?

### ¿Por qué usar Flask?

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 en donde no se necesitan muchas extensiones.

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.

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 pueden ver los valores de las variables. Además está la posibilidad de integrar pruebas unitarias.

Es compatible con Python 3.x.

Es compatible con wsgi: Wsig es un protocolo que utiliza los servidores web para servir las páginas web escritas en Python.

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.

Soporta de manera nativa el uso de cookies seguras.

Se pueden usar sesiones.

Flask no tiene ORMs: Pero se puede usar una extensión para resolver esto.

Sirve para construir servicios web (como una API REST) o aplicaciones de contenido estático.

Flask es Open Source y está amparado bajo una licencia BSD.

Buena documentación, código de GitHub y lista de correos.

Fuente:

<https://openwebinars.net/blog/que-es-flask/>

<https://j2logo.com/tutorial-flask-espanol/>

## 4. Extensiones de Flask

### Extensiones de Flask

Aquí se mencionaran las más usadas con Flask:

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.

Flask es una alternativa para la construcción de Apps Web con Python, tiene una buena curva de aprendizaje y se puede aprender muy rápido.

Fuente:

<https://openwebinars.net/blog/que-es-flask/>

<https://j2logo.com/tutorial-flask-espanol/>