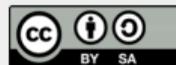


# IMPLANTACIÓN DE APLICACIONES WEB PYTHON

JOSÉ DOMINGO MUÑOZ

IES GONZALO NAZARENO

NOVIEMBRE 2022



# INTRODUCCIÓN A PYTHON



Python es un lenguaje:

- Interpretado
- Alto nivel
- Multiparadigma, ya que soporta **orientación a objetos, programación imperativa y programación funcional.**
- Multiplataforma
- Libre



**Entorno virtual:** “espacios aislados” (directorio) donde poder instalar distintas versiones de programas y paquetes python. Nos permiten:

- Utilizar versiones de paquetes python que no son las que vienen empaquetadas oficialmente en nuestra distribución linux
- Acercar los entornos de desarrollo, prueba y producción.
- Tener en producción varias aplicaciones con versiones de librerías pythons distintas.



```
# apt-get install python3-venv
$ python3 -m venv entorno
$ source entorno/bin/activate
(entorno)$ pip install django
(entorno)$ pip install -r requirements.txt
(entorno)$ deactivate
$
```

- [VÍDEO: Entornos de desarrollo virtuales con Python3](#)



## ¿Cómo podemos hacer que un servidor web como apache2 sea capaz de servir una aplicación escrita en python?

- Para ello se utiliza un protocolo que nos permite comunicar al servidor web con la aplicación web: [WSGI \(Web Server Gateway Interface\)](#).
- WSGI define las reglas para que el servidor web se comuniquen con la aplicación web.



Para que un servidor web o un servidor de aplicaciones pueda ejecutar el código python:

- El servidor accede a un único fichero (fichero de entrada). Este fichero se llama **fichero WSGI**.
- La aplicación web python con la que se comunica el servidor web utilizando el protocolo WSGI se debe llamar `application`. Por lo tanto el fichero WSGI entre otras cosas debe nombrar a la aplicación de esta manera.



Para una aplicación flask podemos tener un fichero `wsgi.py` con el siguiente contenido:

```
from programa import app as application
```

- **programa** corresponde con el nombre del módulo, es decir del fichero del programa, en nuestro caso se llama `programa.py`.
- **app** corresponde a la aplicación flask creada en `programa.py`: `app = Flask(__name__)`.
- Importamos la aplicación flask, pero la llamamos **application**, necesario para que el servidor web pueda enviarle peticiones.



- Si tenemos apache2 podemos usar el módulo wsgi:  
`libapache2-mod-wsgi-py3`.
- Con apache2 y nginx podemos usar un servidor de aplicación python: [Lista](#).  
Vamos a usar **uwsgi**.
- En este caso el **contenido estático lo devuelve el servidor web** y la **ejecución del código python lo hace uwsgi**. El servidor web hará de **proxy inverso** para que podamos acceder a la aplicación.



Un framework es una aplicación, que nos ayuda en el desarrollo de aplicaciones web. Ejemplos de frameworks python: flask, django,...

- [WebFrameworks de Python](#)



[Django](#) es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC).

El modelo MVC es un patrón de diseño software que separa los datos de la aplicación, la lógica del programa y la representación de la información:

- **El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información.
- **El Controlador:** Es la parte del programa donde se implementa la lógica y las funciones de la aplicación.
- **La Vista:** Representa la información ofrecida por la aplicación en un formato adecuada.

[Lista de cms desarrollados en Django](#)

