

INTRODUCCIÓN A PYTHON FLASK

JOSÉ DOMINGO MUÑOZ

IES GONZALO NAZARENO

MARZO 2022



ENTORNOS VIRTUALES EN PYTHON



- El **Python Package Index** o **PyPI**, es el repositorio de paquetes de software oficial de paquetes Python.
- **pip**: Sistema de gestión de paquetes utilizado para instalar y administrar paquetes de PyPI.



- Utilizar el que este empaquetado en la distribución que estés usando. ¿Y si necesitamos una versión determinada?

```
$ apt-cache show python3-requests
...
Version: 2.25.1+dfsg-2
```

- Instalar **pip** en nuestro equipo, y como root instalar el paquete python que nos interesa. **!!!**Podemos romper dependencias de los paquetes del sistema.

```
$ pip_search requests
requests      2.27.1
```

- **Utilizar entornos virtuales**



Es un mecanismo que me permite gestionar programas y paquetes python **sin tener permisos de administración**, es decir, **cualquier usuario sin privilegios** puede tener uno o más “**espacios aislados**” (ya veremos más adelante que los entornos virtuales se guardan en **directorios**) donde poder instalar distintas versiones de programas y paquetes python. Para crear los entornos virtuales vamos a usar el módulo **venv**.



CREANDO ENTORNOS VIRTUALES PYTHON

■ Instalamos el módulo **venv**:

```
$ apt install python3-venv  
...  
Version: 2.25.1+dfsg-2
```

■ Como un **usuario sin privilegios** creamos el entorno (se va a crear un directorio que podemos guardar en un directorio **venv**):

```
$ python3 -m venv entorno_prueba
```

■ Para activar y desactivar el entorno virtual:

```
$ source entorno_prueba/bin/activate  
(entorno_prueba)$ deactivate
```



INSTALANDO PAQUETES EN UN ENTORNO VIRTUAL

- En un entorno virtual activo, puedo instalar un paquete:

```
(entorno_prueba)$ pip install requests
```

- Si queremos ver los paquetes instalados:

```
(entorno_prueba)$ pip list
```

- Puedo guardar los paquetes instalados en un fichero...

```
(entorno_prueba)$ pip freeze > requirements.txt
```

- ... para instalar los mismos paquetes en otro ordenador:

```
(otro_entorno)$ pip install -r requirements.txt
```



¿POR QUÉ USAMOS ENTORNOS VIRTUALES?

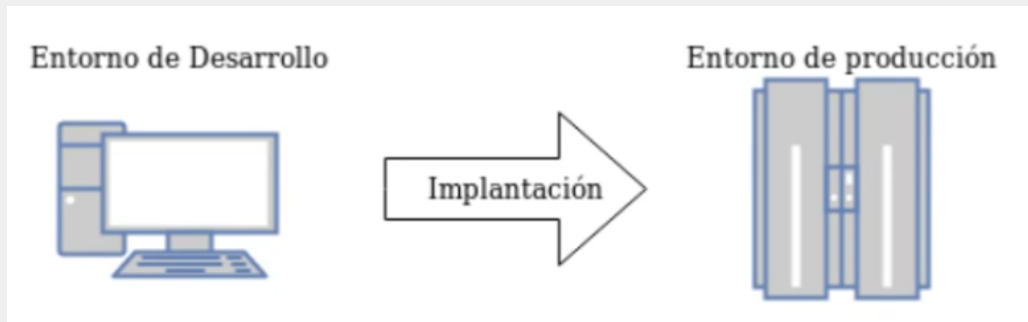


Figura 1: Entornos

- Necesito las mismas librerías y versiones en **desarrollo y producción**.
- Los entornos suelen tener **distintos SO**.
- Solución: **ENTORNOS VIRTUALES**.



FLASK



¿QUÉ ES FLASK?

Flask es un **framework** python que nos permite construir **páginas web dinámicas**.

- Incluye un **servidor web de desarrollo** para que puedas probar tus aplicaciones sin tener que instalar un servidor web.
- Buen manejo de **rutras**: Con el uso de un decorador python podemos hacer que nuestra aplicación con URL simples y limpias.
- Flask se apoya en el motor de plantillas **Jinja2**.
- Flask es **Open Source** y está amparado bajo una **licencia BSD**.



INSTALACIÓN DE FLASK

Creo un entorno virtual, lo activo e instalo el paquete flask:

```
$ python3 -m venv flask
$ source flask/bin/activate
(flask)$ pip install flask
(flask)$ flask --version
Python 3.9.2
Flask 2.0.3
Werkzeug 2.0.3
```

Cuando terminamos de trabajar podemos desactivar el entorno:

```
(flask)$ deactivate
```



¿CÓMO VAMOS A TRABAJAR CON FLASK? - URLS

- En PHP las **url corresponden a ficheros** en el servidor:

`https://dit.gonzalonazareno.org/moodle/course/view.php?id=25`

- En python flask la **URL son virtuales**, no corresponden a ficheros en el servidor.

`https://dit.gonzalonazareno.org/gestiona/grupos/asir1`

- Flask, examina la URL del navegador, comprueba que hemos definido la ruta, y ejecuta un programa que suele terminar mostrando una página web dinámica.



¿CÓMO VAMOS A TRABAJAR CON FLASK? - RUTAS Y VISTAS

- **Rutas:** Vamos a declarar las **URLS** con las que podemos acceder a la aplicación.
- Vista:** Si la URL que ponemos en el navegador corresponder con alguna de las que hemos declarado se ejecuta una función, que llamamos vista.

```
@app.route('/')  
def inicio():  
    ...  
  
@app.route('/articulos')  
def articulos():  
    ...  
  
@app.route('/acercade')  
def acercade():  
    ...
```



¿CÓMO VAMOS A TRABAJAR CON FLASK? - ACCIONES EN LAS VISTAS

- En la vista **se puede ejecutar algún código**:
 - ▶ Buscar información en una BD.
 - ▶ Buscar información en un servicio web
 - ▶ Gestionar la información enviada desde un formulario
 - ▶ Gestionar enviada en la URL
 - ▶ Cualquier operación adicional
- De estas operaciones podemos obtener distintas **variables** con **información**.



- Después de ejecutar el código que hemos puesto en la vista, se nos devolverá un resultado:
 - ▶ Generar una página web dinámica a partir de una **plantilla** a la que podemos enviar la información que hemos generado. Las **plantillas** son parecidas a las páginas html, pero tienen lógica (for, if, ...)
 - ▶ Generar una **redirección** que nos lleve a otra URL.
 - ▶ Generar una **respuesta http de error** (por ejemplo **404**).



EJEMPLOS PARA APRENDER FLASK



EJEMPLO 1: MI PRIMER PROGRAMA EN FLASK

