Entornos virtuales.

Un virtualenv o entorno virtual de Python es un ambiente creado con el objetivo de aislar recursos como librerías y entornos de ejecución.

En cada entorno se puede utilizar diferentes versiones de un mismo paquete.

En otras palabras, en la misma computadora, cada entorno puede tener instaladas diferentes versiones de una misma librería sin crear ningún tipo de conflicto.

Se pueden hacer diferentes desarrollos, con versiones específicas de una misma librería.

# Instalación de ‘virtualenv’.

Para crear entornos es necesario tener instalada la aplicación **virtualenv**.

Si no está ya instalada, se la instala con:

* pip install virtualenv

# Creación de un entorno virtual.

Un entorno virtual es una subcarpeta.

Para crear un entorno se deberá posicionarse en la carpeta donde se quiere crear el mismo.

Desde la línea de comando deberá ingresarse:

* virtualenv NOMBRE\_DE\_CARPETA\_ENTORNO

# Creación de un entorno virtual con un intérprete Python específico.

* virtualenv -p **ruta\_a\_python.exe\_específica** NOMBRE\_DE\_CARPETA\_ENTORNO

Ejemplo:

* virtualenv -p "C:\Python38-32\python.exe" env

# Activar el entorno virtual en Windows.

1. En la línea de comandos, posicionarse en la carpeta que contiene el entorno.
2. Luego ingresar:
   * NOMBRE\_DE\_CARPETA\_ENTORNO\Scripts\activate

# Trabajar dentro del entorno virtual.

Para estas opciones, se debe tener el entorno activado.

## Instalar paquete/librería.

* pip install nombre-librería

## Instalar una versión concreta de un paquete/librería.

* pip install nombre-librería==1.0.1 (*número de versión*).

## Actualizar la versión de un paquete/librería.

* pip install nombre-librería -u

## Desinstalar una librería.

* pip uninstall nombre-librería

## Listar todas las librerías.

* pip list

## Listar todos los paquetes/librerías en formato requirements.txt.

* pip freeze

## Crear/Actualizar el fichero requirements.txt.

* pip freeze > requirements.txt

## Mostrar la información de un paquete/librería.

* pip show nombre-librería

# Salir del entorno virtual.

* Deactivate

# Eliminar entorno virtual.

Simplemente se elimina la carpeta del entorno virtual.

Usar entornos virtuales desde Visual Studio Code.

Simplemente usar todos los comandos anteriores, desde una sesión de Terminal.

Como VSCode utiliza PowerShell, es necesario (por única vez) habilitar en el mismo la ejecución de Scripts.

# Habilitar ejecución de Scripts en PowerShell.

**Paso 1**: iniciar PowerShell como administrador, e ingresar el siguiente comando:

* Set-ExecutionPolicy RemoteSigned -Force

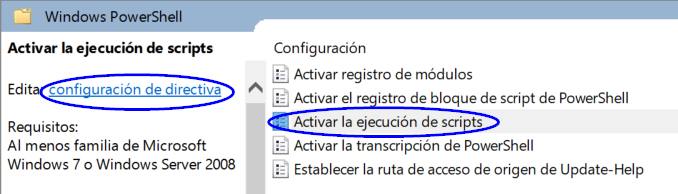
**Paso 2:** Abrir el ‘Editor de directivas de grupo local’. Con Windows+R, ejecutar:

* gpedit.msc

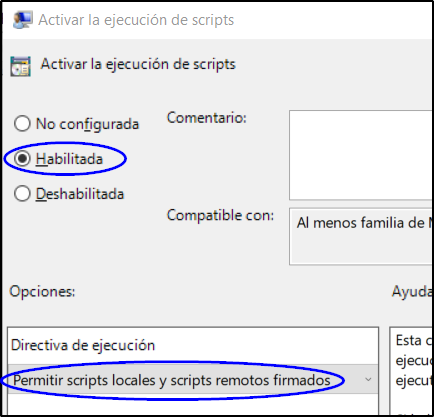
**Paso 3:** Dentro del Editor de directivas de grupo local, seguir la siguiente ruta:

* Configuración del equipo
  + Plantillas administrativas
    - Componentes de Windows
      * Windows PowerShell

**Paso 4:** ya dentro de la opción anterior…



**Paso 5:** En la ventana resultante, establecer los siguientes valores:



Instalación y desinstalación de paquetes.

# Desinstalación de todos paquetes instalados.

**Paso 1:** armado del archivo **requeriments.txt** con todos los paquetes instalados.

* pip freeze > requirements.txt

**Paso 2:** desinstalación a parti de dicho archivo.

* pip uninstall -r requirements.txt -y

# Instalación de paquete de entornos virtuales.

* pip install virtualenv

# Instalación de Django.

* pip install django