**Django Configuración de ambiente virtual en Windows**

Primero instalar python

$pip install virtualenv

Para instalar el decorador

$pip install virtualenvwrapper-win

mkvirtualenv <name>

Create a new virtualenv environment named <name>. The environment will be created in WORKON\_HOME.

lsvirtualenv

List all of the enviornments stored in WORKON\_HOME.

rmvirtualenv <name>

Remove the environment <name>. Uses folder\_delete.bat.

workon [<name>]

If <name> is specified, activate the environment named <name> (change the working virtualenv to <name>). If a project directory has been defined, we will change into it. If no argument is specified, list the available environments. One can pass additional option -c after virtualenv name to cd to virtualenv directory if no projectdir is set.

deactivate

Deactivate the working virtualenv and switch back to the default system Python.

add2virtualenv <full or relative path>

If a virtualenv environment is active, appends <path> to virtualenv\_path\_extensions.pth inside the environment’s site-packages, which effectively adds <path> to the environment’s PYTHONPATH. If a virtualenv environment is not active, appends <path> to virtualenv\_path\_extensions.pth inside the default Python’s site-packages. If <path> doesn’t exist, it will be created.

Si no se trabaja con el decorador es necesario

Una vez creado nuestro entorno virtual, necesitamos activarlo para poder usarlo. Para ello vamos a usar uno de los scripts que se han instalado por defecto al crearlo. El script necesario es activate, y se encuentra en el directorio bin que se ha creado dentro del de nuestro entrono virtual:

$source mis-virtualenvs/mi-entorno-virtual/bin/activate