# **Instalación de Django**

**Nota** Si ya has realizado los pasos de instalación, esto ya lo has hecho. ¡Puedes avanzar directamente al siguiente capítulo!

Parte de esta sección está basada en tutoriales de Geek Girls Carrots (<https://github.com/ggcarrots/django-carrots>).

Parte de esta sección se basada en el [django-marcador tutorial](http://django-marcador.keimlink.de/) bajo licencia Creative Commons Attribution-ShareAlike 4.0 internacional. El tutorial de django-marcador tiene derechos de autor de Markus Zapke-Gündemann et al.

## **Entorno virtual**

Antes de instalar Django, instalaremos una herramienta extremadamente útil que ayudará a mantener tu entorno de desarrollo ordenado en tu computadora. Es posible saltarse este paso, pero es altamente recomendable. ¡Empezar con la mejor configuración posible te ahorrará muchos problemas en el futuro!

Así que, vamos a crear un **entorno virtual** (también llamado un *virtualenv*). Virtualenv aísla tu configuración de Python/Django por cada proyecto. Esto quiere decir que cualquier cambio que hagas en un sitio web no afectará a ningún otro que estés desarrollando. Genial, ¿no?

Todo lo que necesitas hacer es encontrar un directorio en el que quieras crear el virtualenv; tu directorio home, por ejemplo. En Windows puede verse como C:\Users\Name (donde Name es el nombre de tu usuario).

**NOTA:** En Windows, asegúrate que este directorio no contenga acentos ni caracteres especiales; si tu nombre contiene acentos, utiliza un directorio diferente, por ejemplo C:\djangogirls.

Para este tutorial usaremos un nuevo directorio djangogirls en tu directorio home:

Terminal

$ mkdir djangogirls  
$ cd djangogirls

Haremos un virtualenv llamado myvenv. El comando general estará en el formato:

$ python3 -m venv myvenv

**Virtual environment: Windows**

Para crear un nuevo virtualenv, debes abrir la consola y ejecutar python -m venv myvenv. Se verá así:

C:\Users\Name\djangogirls> python -m venv myvenv

en donde myvenv es el nombre de tu virtualenv. Puedes utilizar cualquier otro nombre, pero asegúrate de usar minúsculas y no usar espacios, acentos o caracteres especiales. También es una buena idea mantener el nombre corto. ¡Vas a referirte a él mucho!

**Virtual environment: Linux y OS X**

Crear un virtualenv en Linux y OS X es tan simple como ejecutar python3 -m venv myvenv. Se verá así:

$ python3 -m venv myvenv

myvenv es el nombre de tu virtualenv. Puedes usar cualquier otro nombre, pero sólo utiliza minúsculas y no incluyas espacios. También es una buena idea mantener el nombre corto. ¡Vas a referirte muchas veces a él!

**NOTA:** En algunas versiones de Debian/Ubuntu quizás recibas el siguiente error:

The virtual environment was not created successfully because ensurepip is not available. On Debian/Ubuntu systems, you need to install the python3-venv package using the following command.  
 apt-get install python3-venv  
You may need to use sudo with that command. After installing the python3-venv package, recreate your virtual environment.

En este caso, sigue las instrucciones mencionadas arriba e instala el paquete python3-venv así:

$ sudo apt-get install python3-venv

**NOTA:** En algunas versiones de Debian/Ubuntu inicializar el entorno virtual puede producir el siguiente error:

Error: Command '['/home/eddie/Slask/tmp/venv/bin/python3', '-Im', 'ensurepip', '--upgrade', '--default-pip']' returned non-zero exit status 1

Para solucionar esto, usa el comando virtualenv en cambio.

$ sudo apt-get install python-virtualenv  
$ virtualenv --python=python3.6 myvenv

**NOTA:** Si obtienes un error como este

E: Unable to locate package python3-venv

intenta ejecutar el siguiente comando en cambio:

sudo apt install python3.6-venv

## 

## **Trabajar con virtualenv**

El comando anterior creará un directorio llamado myvenv (o cualquier nombre que hayas elegido) que contiene nuestro entorno virtual (básicamente un montón de archivos y carpetas).

**Virtual environment: Windows**

Inicia el entorno virtual ejecutando:

C:\Users\Name\djangogirls> myvenv\Scripts\activate

**NOTA:** en Windows 10 quizás obtengas un error en la consola Window PowerShell diciendo execution of scripts is disabled on this system. En ese caso, abre otra ventana Windows PowerShell con la opción "Run as Administrator". Luego intenta escribir lo siguiente antes de iniciar tu entorno virtual:

C:\WINDOWS\system32> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned  
 Execution Policy Change  
 The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about\_Execution\_Policies help topic at http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy? [Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A

**Virtual environment: Linux y OS X**

Inicia el entorno virtual ejecutando:

$ source myvenv/bin/activate

¡Recuerda reemplazar myvenv con tu nombre de virtualenv que hayas elegido!

**NOTA:** a veces source podría no estar disponible. En ese caso trata hacerlo de esta forma:

$ . myvenv/bin/activate

Sabrás que tienes virtualenv iniciado cuando veas que la línea de comando tiene este prefijo (myvenv).

Cuando trabajes en un entorno virtual, python automáticamente se referirá a la versión correcta, de modo que puedes utilizar python en vez de python3.

OK, tenemos todas las dependencias importantes en su lugar. ¡Finalmente podemos instalar Django!

## **Instalar Django**

Ahora que tienes tu virtualenv iniciado, puedes instalar Django.

Antes de hacer eso, deberíamos asegurarnos de que tenemos la última versión de pip, el programa que usamos para instalar Django:

(myvenv) ~$ pip install --upgrade pip

Luego ejecuta pip install django~=1.11.0 (nota que usamos la tilde seguido de un signo igual: ~=) para instalar Django.

(myvenv) ~$ pip install django~=1.11.0  
Collecting django~=1.11.0  
 Downloading Django-1.11.3-py2.py3-none-any.whl (6.8MB)  
Installing collected packages: django  
Successfully installed django-1.11.3

**Virtual environment: Windows**

Si obtienes un error al ejecutar pip en Windows comprueba si la ruta de tu proyecto contiene espacios, acentos o caracteres especiales (por ejemplo, C:\Users\User Name\djangogirls). Si lo tiene, por favor considera moverla a otro lugar sin espacios, acentos o caracteres especiales (sugerencia: C:\djangogirls). Crea un nuevo entorno virtual en ese directorio, luego elimina el viejo y vuelve a probar el comando anterior (mover el directorio del virtualenv no funcionará debido a que virtualenv usa rutas absolutas)

**Virtual environment: Windows 8 y Windows 10**

Tu línea de comandos quizás se congele luego de que intentas instalar Django. Si esto sucede, intenta este comando en cambio:

C:\Users\Name\djangogirls> python -m pip install django~=1.11.0

**Virtual environment: Linux**

Si obtienes un error al ejecutar pip en Ubuntu 12.04 ejecuta python -m pip install -U ---force-resintall pip para arreglar la instalación de pip en el virtualenv.

¡Eso es todo! Ahora estás listo (por fin) para crear una aplicación Django!