

### Pasos para la puesta en marcha de Django

1.Comprobar las versiones de python, pues deben ser superiores a las 3.8

```
python --versión
```

También en un .py, con el siguiente código:

```
#codigo para obtener la versión
import sys
print("Versión de Python:", sys.version)
```

2.Pasamos a instalar django. Es interesante la web, dónde nos indican también los pasos.

([Descargar Django | Django](#))

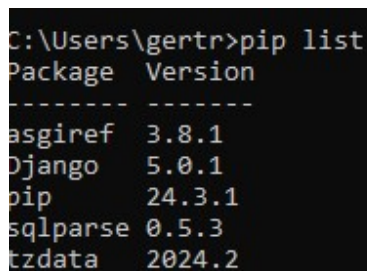
La primera instrucción para instalar sería la siguiente. Se instalará django entre los paquetes de python que tenemos instalado en nuestro sistema. También se podría instalar django para un entorno virtual creado para un proyecto específico. Lo veremos más adelante.

```
pip install Django
```

```
pip install Django==5.1.4 #Se le puede indicar la versión
```

Podemos ver las versiones que tenemos de django depues de la instalación con la instrucción:

```
pip list
```



```
C:\Users\gertru>pip list
Package Version
-----
asgiref 3.8.1
Django  5.0.1
pip      24.3.1
sqlparse 0.5.3
tzdata   2024.2
```

También si simplemente queremos saber la versión de Django:

```
python -m django --version
```

3.Creación de un proyecto de Django. Lo suyo sería crearlo dentro de un espacio de trabajo workspace.

Las instrucciones serían:

`django-admin startproject NombreProyecto`

```
PS C:\Users\gertr\workspace> django-admin startproject UnoDjango
PS C:\Users\gertr\workspace> cd unodjango
PS C:\Users\gertr\workspace\unodjango> code .
```

Con las instrucción Code . dentro de nuestro proyecto , se abre VScode con el proyecto que como puede observarse contiene un carpeta con el mismo nombre del proyecto dónde se encuentran los archivos de configuración del proyecto.

Se nos abre VScode con una estructura de paquetes dentro de nuestro proyecto creado. A groso modo, las funciones que tienen cada uno de ellos serían:

Manage.py : permite ejecutar aplicaciones así como comandos.

Dentro del paquete que tiene el mismo nombre que nuestro proyecto estarían:

asgi.py : paquete para desplegar nuestras aplicaciones django

setting.py: están las configuraciones de nuestra aplicación. Ej: modo debug, host, aplicaciones instaladas, base de datos, etc.

urls.py: gestionar las rutas de django o de aplicaciones

.py: relativo al despliegue de una aplicación en django

Hay que diferenciar entre la instrucción **startproject**, que crea una aplicación dentro del proyecto con el mismo nombre que el proyecto y la instrucción:

`python manage.py startapp nombre_aplicación`

Esta última crea otra aplicación con otra lógica dentro del mismo proyecto. La usaremos más adelante.

4.Levantar servidor de Django, dentro de nuestra aplicación. Lo ejecutamos dentro de dónde se encuentra el archivo manage.py, en este caso, dentro de nuestro proyecto django creado.

`Python manage.py runserver`

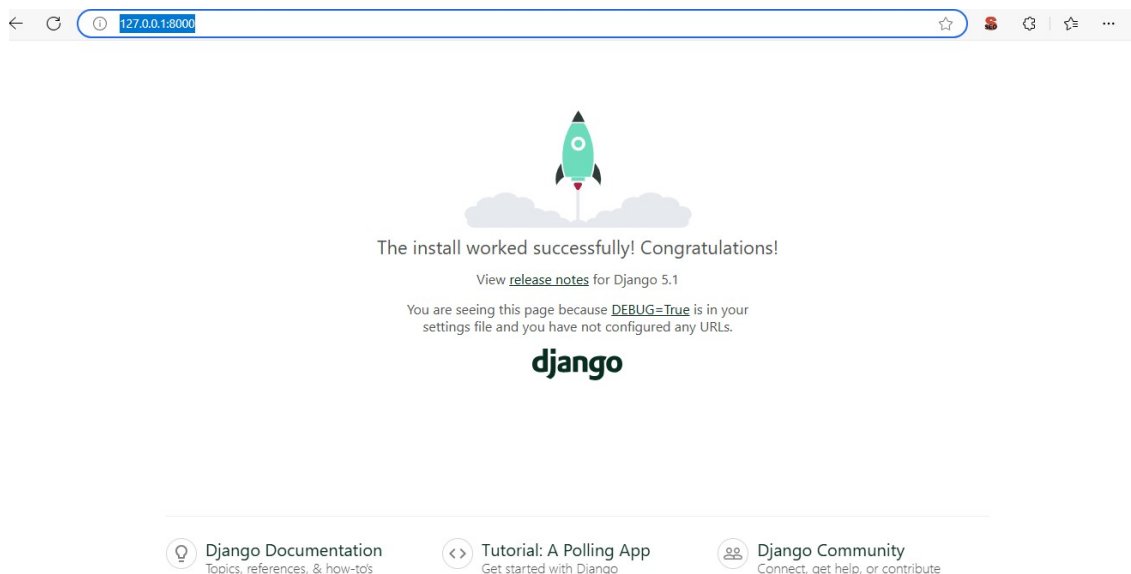
```
PS C:\Users\gertr\workspace> django-admin startproject UnoDjango
PS C:\Users\gertr\workspace> cd unodjango
PS C:\Users\gertr\workspace\unodjango> code .
PS C:\Users\gertr\workspace\unodjango> Python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

Nos avisa de que hay 18 aplicaciones que falta migrar, no pasa nada, más adelante las ejecutaremos

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 02, 2025 - 11:01:21
Django version 5.0.1, using settings 'UnoDjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

```
[02/Jan/2025 11:04:00] "GET / HTTP/1.1" 200 10629
Not Found: /favicon.ico
[02/Jan/2025 11:04:01] "GET /favicon.ico HTTP/1.1" 404 2113
```

Metemos en el navegador el <http://127.0.0.1:8000/> , tal como dice la información de los mensajes y ya aparece la página de inicio de Django



Para cerrar el servidor de Django, en la consola haríamos ctrl+B. Lógicamente ya no se vería la página de inicio.

Siempre tenemos opción de volverlo a iniciar, con la misma instrucción:

```
python manage.py runserver
```