

Iniciar un proyecto en Django

- [Iniciar un proyecto en Django](#)
 - [Crear el entorno virtual](#)
 - [Activar el entorno virtual](#)
 - [Windows \(PowerShell\)](#)
 - [Windows \(Git Bash\)](#)
 - [Linux / macOS](#)
 - [Verificación](#)
 - [Desactivar el entorno virtual](#)
 - [Instalar Django](#)
 - [Inicializar el proyecto Django](#)
 - [Diferencias entre ambos métodos](#)
 - [django-admin startproject](#)
 - [python -m django startproject](#)
 - [Recomendación](#)
 - [Ejecutar el servidor de desarrollo](#)
 - [Ejecutar el servidor en un puerto personalizado](#)

Crear el entorno virtual

Desde el directorio del proyecto, crea un entorno virtual con:

```
python -m venv venv
```

Esto generará la carpeta `venv` con una instalación aislada de Python.

Activar el entorno virtual

Windows (PowerShell)

Es común encontrar el error:
“running scripts is disabled”

Para solucionarlo, ejecuta una sola vez (como usuario):

```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
```

Luego, activa el entorno virtual:

```
.\venv\Scripts\Activate.ps1
```

Windows (Git Bash)

```
source venv/Scripts/activate
```

Linux / macOS

```
source venv/bin/activate
```

Verificación

Si el entorno está activo, el prompt debería verse similar a:

```
(venv) C:\proyecto>
```

```
(venv) PS C:\proyecto>
```

(el formato exacto puede variar según la terminal)

Desactivar el entorno virtual

En cualquier sistema operativo:

```
deactivate
```

Instalar Django

Con el entorno virtual activo, instala Django:

```
pip install django
```

Inicializar el proyecto Django

Para crear un nuevo proyecto Django, puedes usar cualquiera de los siguientes comandos:

```
python -m django startproject miproyecto
```

o bien:

```
django-admin startproject miproyecto
```

Ambos comandos generan el directorio principal del proyecto con la estructura base de Django.

Diferencias entre ambos métodos

django-admin startproject

El comando `django-admin` ejecuta el script de Django que se encuentre primero en el **PATH del sistema**.

Implicaciones:

- Puede utilizar una instalación **global de Django** en lugar de la instalada en el entorno virtual.
- Funciona correctamente siempre que el entorno virtual esté **bien activado**.
- Es el método más habitual en tutoriales y documentación básica.

python -m django startproject

Este comando le indica explícitamente al **intérprete de Python activo** que ejecute el módulo `django`.

En la práctica significa:

"Usa el Django instalado en este Python".

Ventajas:

- Garantiza el uso del Django instalado en el **entorno virtual activo**.
- Evita problemas relacionados con el PATH del sistema.
- Es más **predecible y seguro** en entornos profesionales, CI/CD y contextos educativos.

● Recomendación

Aunque ambos métodos son válidos, se recomienda preferir:

```
python -m django startproject miproyecto
```

ya que reduce ambigüedades y asegura coherencia entre el entorno virtual y la versión de Django utilizada.

Ejecutar el servidor de desarrollo

Ingrésa al directorio creado y levanta el servidor:

```
cd miproyecto
```

```
python manage.py runserver
```

Por defecto, el servidor se ejecuta en el puerto 8000:

```
http://127.0.0.1:8000/
```

Ejecutar el servidor en un puerto personalizado

Para usar otro puerto, por ejemplo el 4444:

```
python manage.py runserver 4444
```

Accede desde el navegador en:

```
http://127.0.0.1:4444/
```