

INSTALACIÓN Y CONFIGURACIÓN DJANGO

- Django es un framework de desarrollo web de código abierto, escrito en Python.
- Django sigue el **patrón MVC**.
- Debido a que el controlador es manejado por el propio framework, podremos escuchar que es un framework **MTV** en lugar de MVC.

Modelo: Capa de acceso a datos.

Template: Capa de presentación (Sería como la vista en otros framework MVC).

Vista: Capa de la lógica de negocios (Sería como el controlador en otros framework MVC).

Instalar Django con el comando pip desde la Terminal:

pip install django

Vamos a seguir los pasos para poder instalar y configurar Django en nuestro editor **Visual Studio Code**.

Comenzamos creando una carpeta en nuestro Equipo para alojar los diferentes proyectos de Django.

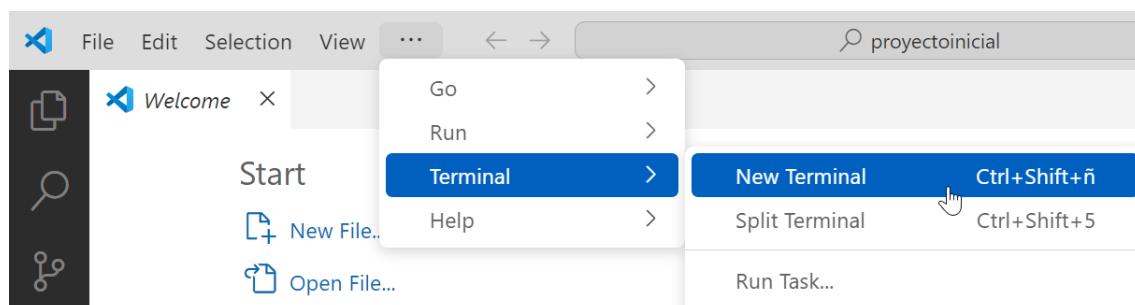
En **Documents**, creamos una carpeta llamada **django**

Dentro de dicha carpeta iremos creando los diferentes proyectos que pudiéramos tener.

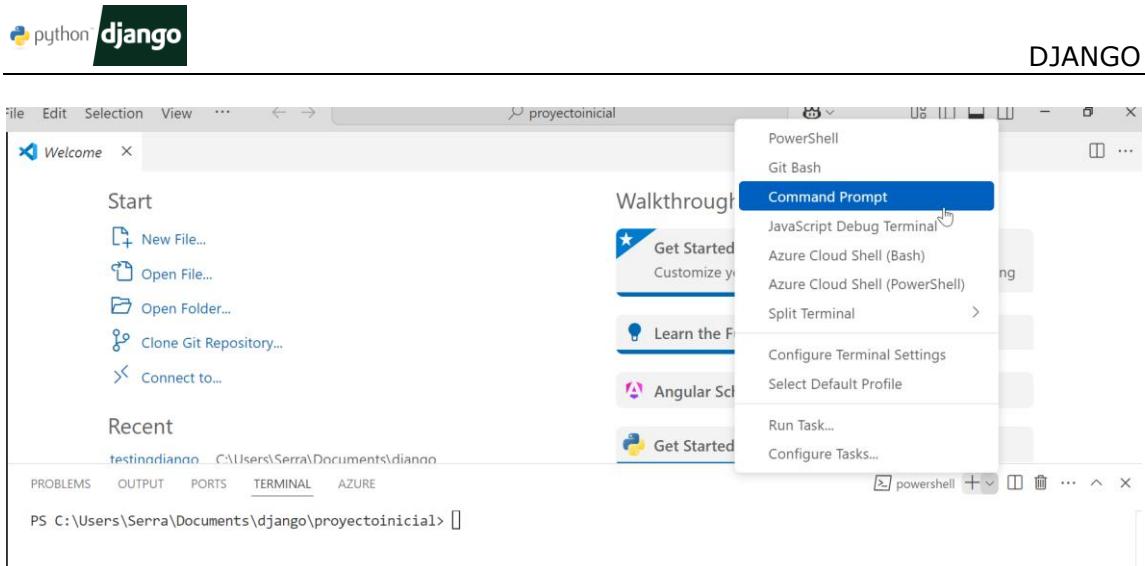
Cada carpeta será una aplicación de django.

Comenzamos creando una nueva carpeta llamada **proyecto inicial** dentro de **django**

1. Abrir VS Code y abrir la carpeta **proyecto inicial**.
2. Abrir la pestaña para acceder a la consola (terminal) y poder ejecutar desde la línea de comandos.



3. En la terminal seleccionamos **Command Prompt**



4. Dentro de Django tenemos el concepto de **proyecto**, dentro de un proyecto podríamos tener muchas **aplicaciones**.

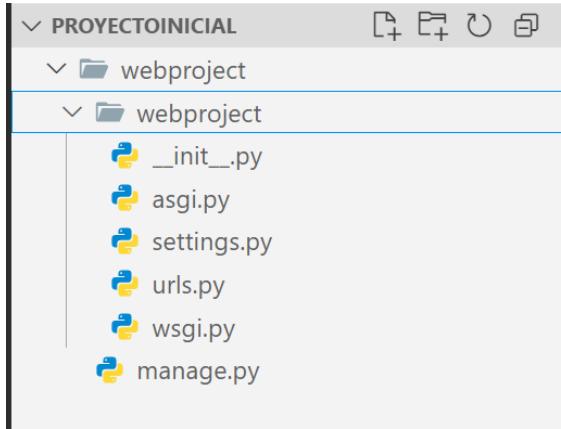
Vamos a crear un proyecto y dentro nuestra primera aplicación.

Creamos un proyecto nuevo con el comando: `Django-admin startproject "nombre de proyecto"`

django-admin startproject webproject

A screenshot of the Visual Studio Code interface. The title bar says 'PROBLEMS', 'OUTPUT', 'PORTS', 'TERMINAL', and 'AZURE'. The terminal tab is active, showing a PowerShell prompt: 'PS C:\Users\Serra\Documents\django\proyecto inicial> django-admin startproject webproject'. The output line below shows the creation of a new directory: 'PS C:\Users\Serra\Documents\django\proyecto inicial> |'. The status bar at the bottom right shows 'powershell + - X'.

Al ejecutar ese comando veremos que la consola **no** muestra mensajes, pero en el explorador de proyectos comprobaremos que nos ha creado una carpeta nueva.



Nos crea una serie de archivos, aunque cara al desarrollo los más importantes son `settings.py` y `urls.py`, iremos viendo para que sirven cada uno de ellos.

5. El siguiente paso será crear nuestra app dentro del proyecto recién creado.

El comando a utilizar será:

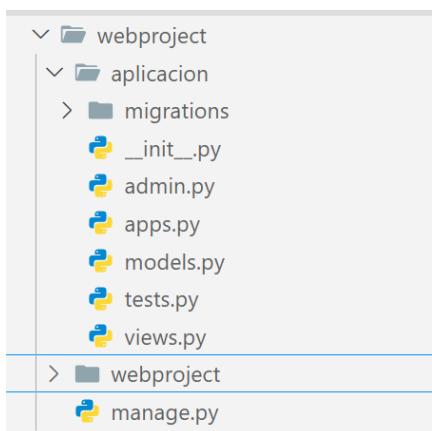
python manage.py startapp nombre_app

Primero nos metemos dentro de la carpeta proyecto1 (**entramos en el proyecto**), ahí crearemos la aplicación.

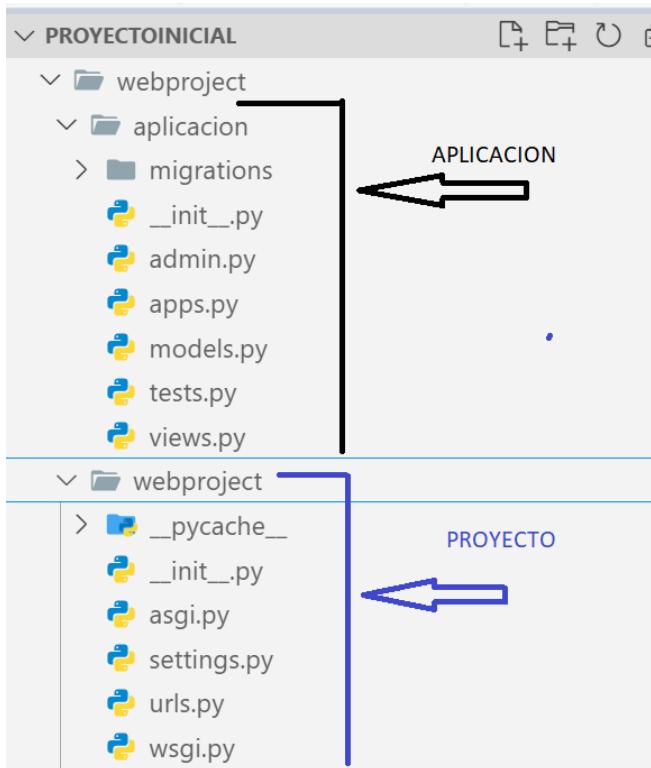
cd webproject

```
\Documents\django\proyecto inicial> cd webproject  
\Documents\django\proyecto inicial\webproject> python manage.py startapp aplicacion
```

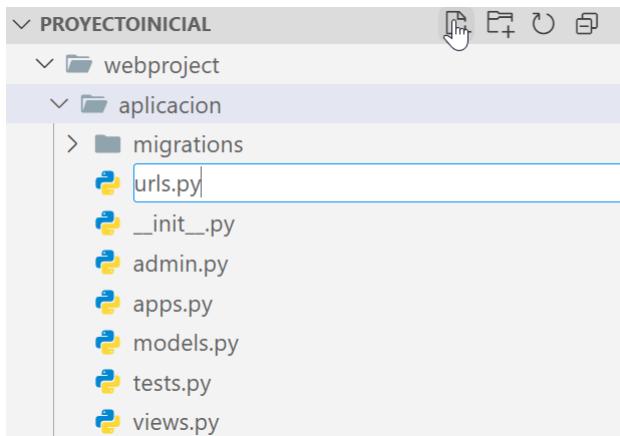
Si desplegamos el explorador de proyectos veremos que nos ha creado nuestra aplicación.



Ahora mismo tenemos dos carpetas donde podemos diferenciar el proyecto de la app.



6. Dentro de la carpeta **aplicacion** vamos a crear un fichero llamado **urls.py**, no viene creado por defecto y en nuestro ejercicio si será necesario.



EJECUTAR LA APLICACIÓN

Antes de comenzar a realizar nuestras primeras vistas, vamos a comprobar que nuestra aplicación está montada perfectamente.

Ejecutamos en el terminal la aplicación con el comando:

python manage.py runserver

System check identified no issues (0 silenced).

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.

February 15, 2025 - 10:51:35

Django version 5.1.6, using settings 'settings'

Starting development server at <http://127.0.0.1:8000/>

Quit the server with CTRL-BREAK.

Vemos que nos indica que podemos visitarlo abriendo el navegador a través de la URL: <http://127.0.0.1:8000/>

Cargamos esa dirección en el navegador y vemos que nos da respuesta.

