**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 **proyectoinicial** dentro de **django**

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

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

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

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

1. 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**

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

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.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

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.

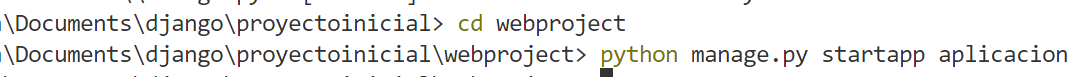
1. El siguiente paso será crear nuestra app dentro del proyecto recién creado.

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**



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

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

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

Diagrama

El contenido generado por IA puede ser incorrecto.

1. 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.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

**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**

Texto

El contenido generado por IA puede ser incorrecto.

Vemos que nos indica que podemos visitarlo abriendo el navegador a través de la URL:[**http://127.0.0.1:8000/**](http://127.0.0.1:8000/)

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

Interfaz de usuario gráfica, Texto, Sitio web

El contenido generado por IA puede ser incorrecto.