



<u>Indice</u>

Introduccion
Creación de un primer proyecto
Base de Datos



Introducción

Django es un framework escrito en Python, es decir, un marco o entorno de trabajo formado por un conjunto de herramientas, librerías y buenas prácticas.

Django permite crear sitios web complejos (aplicaciones web) de forma rápida y sencilla.

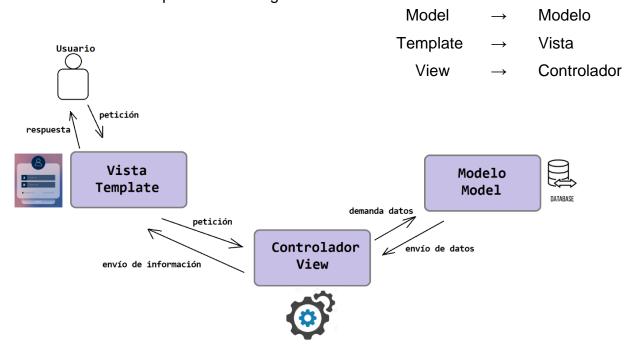
Sitio web	Aplicación web
Conjunto de páginas estáticas que	Plataformas principalmente
entregan información.	interactivas que se centran en que
	los usuarios realicen acciones.

Modelo Vista Controlador (MVC)

Patrón que consiste en dividir cualquier aplicación en tres grandes módulos.

- **Modelo**: encargado de gestionar los datos, en general con una Base de Datos.
- Vista: encargado de mostrar la información al usuario, la interfaz gráfica.
- Controlador: encargado de gestionar las comunicaciones entre la vista y el modelo.

En Django, se utiliza el modelo MTV, el cual es fundamentalmente igual que el modelo MVC pero con sus siglas diferentes:





Instalación local de Django

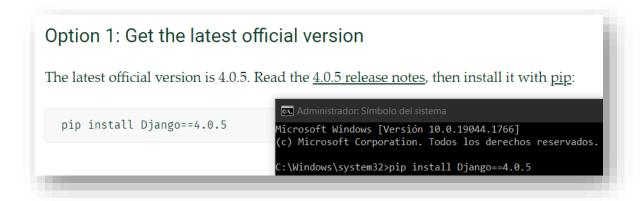
Requiere tener Python instalado.

En Linux ya viene preinstalado por defecto.

Dirigirse a <u>www.djangoproject.com</u>

Download latest release: 4.0.5

Presionar el botón de descargar y observar el comando que debe ejecutarse desde el **símbolo del sistema** de Windows con permisos de administrador.



Para comprobar que Django esté instalado correctamente, ejecutar el intérprete de Python con el comando **python** desde el mismo símbolo del sistema, importar la biblioteca **django** y luego comprobar su versión con el método .VERSION

```
Administrador: Símbolo del sistema - python

C:\Windows\system32>python

Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import django
>>> django.VERSION
(4, 0, 5, 'final', 0)
>>>
```

Instalación virtual de Django

Instalar este framework en una máquina virtual puede ser útil para trabajar con distintas versiones de Python y/o Django y probar sus funcionalidades.

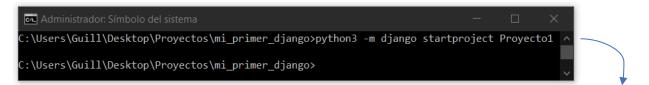


Creación de un primer proyecto

Para comenzar, se debe crear un directorio donde almacenar todo el proyecto que se va a crear.

Una vez hecho lo anterior, se procede a abrir el **Símbolo del sistema**, en el cual nos vamos a dirigir al directorio creado y ejecutar el comando:

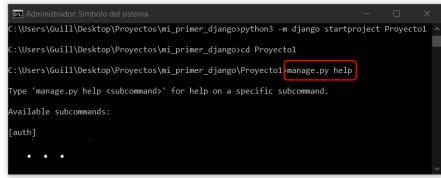
python3 -m django startproject NOMBRE_DEL_PROYECTO



Este proceso creará una carpeta con el nombre del proyecto elegido.

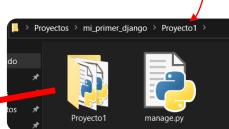
Dentro de dicha carpeta, se encontrará un archivo de Python llamado **manage**. Se puede observar todos los comandos disponibles mediante el uso del siguiente comando (se debe estar dentro de la carpeta del

proyecto):



Dentro de la subcarpeta con el mismo nombre del proyecto, se encontrarán otros archivos Python, los cuales son:

- __init___
- settings configuración del proyecto
- urls almacena las url del proyecto
- wsgi relativo al servidor web que se utilizará en el proyecto





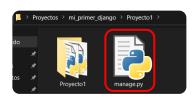
Base de datos

Los gestores de bases de datos soportados por Django son:

- SQLite3 (por defecto en Python)
- PostgreSQL (recomendado)
- MySQL
- Oracle

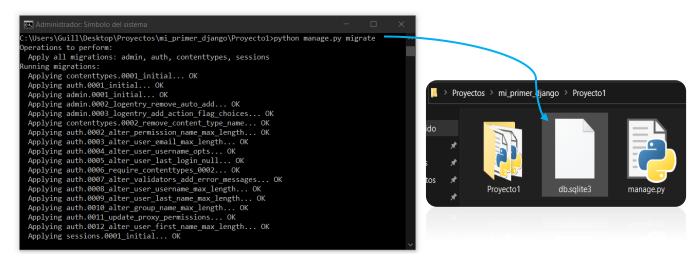
También se puede trabajar con otros gestores de bases de datos si se utiliza un "conector" de terceros.

Una vez creado un proyecto, se puede crear una primer base de datos de SQLite3, la cual viene instalada en Python por defecto.



Para ello, se procede de la siguiente manera:

- 1. Acceder en Símbolo del sistema al directorio donde se encuentra nuestro manage.py
- 2. Ejecutar el comando: python manage.py migrate
- 3. Se creará un archivo db.sqlite3 en la carpeta de nuestro proyecto



Una vez inicializada la base de datos, el proyecto ya estará en funcionamiento.

Se puede comprobar mediante la ejecución del servidor de nuestro proyecto (django tiene uno muy básico por defecto), dirigirse a la dirección correspondiente del servidor y nos deberá aparecer la página de bienvenida de django. Para esto, se procede a ejecutar el siguiente comando: **python manage.py runserver**

