



Proyecto Ecommerce Introducción Entornos Virtuales y creación del Proyecto:

Bien como se menciona en la grabación para crear un entorno virtual podemos realizar el proceso de varias maneras, sin embargo, ¿antes de esto que es un entorno virtual?



¿Qué es un entorno virtual en Python?

Un entorno virtual es un espacio aislado que se crea para un proyecto en Python, donde puedes instalar paquetes y dependencias sin afectar el resto del sistema ni otros proyectos.

¿Por qué es útil?

Evita conflictos de versiones entre proyectos.

Permite tener un entorno limpio y controlado para cada aplicación.

Facilita la portabilidad y reproducción del entorno (por ejemplo, usando requirements.txt).



Creando el entorno virtual en distintos sistemas Operativos:

Comando recomendado: pip install virtualenvwrapper-win y pulsas enter:

Al haber instalado el entorno virtual procedemos a crearlos con el comando mkvirtualenv

```
(ecommerce_django) C:\Users\Estudiante\ecommerce_django\ecommerce\ecommerce>cd..  
(ecommerce_django) C:\Users\Estudiante\ecommerce_django\ecommerce>mkvirtualenv ecommerce
```

Ahora para acceder a el usamos el comando workon seguido del nombre del entorno

```
(ecommerce_django) C:\Users\Estudiante\ecommerce_django\ecommerce\ecommerce>cd..  
(ecommerce_django) C:\Users\Estudiante\ecommerce_django\ecommerce>workon ecommerce_django
```

Ahora recuerda que debes instalar django dentro del proyecto con el comando
Pip install django

```
(ecommerce_django) C:\Users\Estudiante\ecommerce_django\ecommerce>pip install django
```



Ahora verificamos las librerías que tenemos instaladas en nuestro entorno con el comando **pip freeze**

```
C:\WINDOWS\system32 x + v  
(ecommerce_django) C:\Users\Estudiante\ecommerce_django\ecommerce\ecommerce>cd .  
(ecommerce_django) C:\Users\Estudiante\ecommerce_django\ecommerce>pip freeze
```

Luego vamos a crear el proyecto con el comando `django-admin startproject ecommerce`

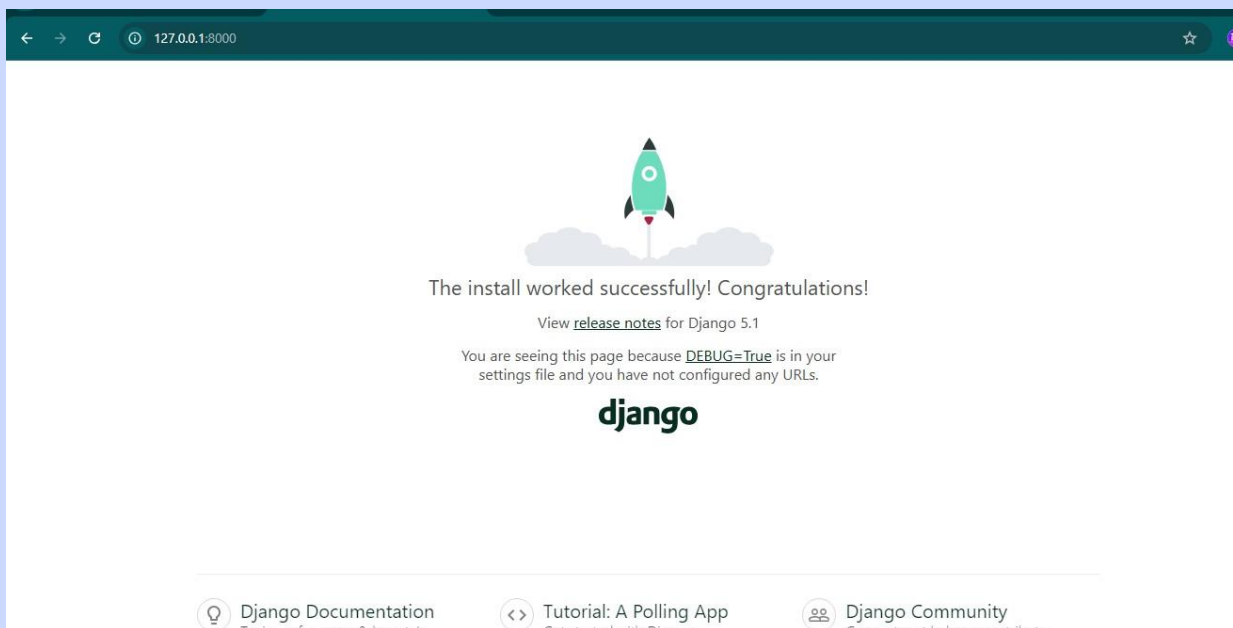
```
C:\WINDOWS\system32\cmd.exe  
(ecommerce_django) C:\Users\juan\ecommerce_django>pip install django  
Collecting django  
  Using cached Django-5.1.5-py3-none-any.whl.metadata (4.2 kB)  
Collecting asgiref<4,>=3.8.1 (from django)  
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)  
Collecting sqlparse>=0.3.1 (from django)  
  Using cached sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)  
Collecting tzdata (from django)  
  Using cached tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)  
Using cached Django-5.1.5-py3-none-any.whl (8.3 MB)  
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)  
Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)  
Using cached tzdata-2025.1-py2.py3-none-any.whl (346 kB)  
Installing collected packages: tzdata, sqlparse, asgiref, django  
Successfully installed asgiref-3.8.1 django-5.1.5 sqlparse-0.5.3 tzdata-2025.1  
  
(ecommerce_django) C:\Users\juan\ecommerce_django>pip freeze  
asgiref==3.8.1  
Django==5.1.5  
sqlparse==0.5.3  
tzdata==2025.1  
  
(ecommerce_django) C:\Users\juan\ecommerce_django>django-admin startproject ecommerce  
  
(ecommerce_django) C:\Users\juan\ecommerce_django>cd ecommerce  
  
(ecommerce_django) C:\Users\juan\ecommerce_django\ecommerce>_
```



Ahora solo debemos arrancar el servidor con el comando **Python manage.py runserver**

```
(ecommerce_django) C:\Users\Estudiante\ecommerce_django\ecommerce>py manage.py runserver
```

Ahora simplemente copiamos en la barra de búsqueda del navegador la dirección http que nos arroja



¡Y perfecto lo hemos logrado!



Terminología

Comandos

pip install django

Definición:

Instala el framework Django en nuestro entorno de Python (normal o virtual).

Importancia:

Es el primer paso para crear aplicaciones web con Django.

pip freeze

Definición:

Muestra una lista de todos los paquetes instalados en el entorno actual con sus versiones.

Importancia:

Es útil para generar un archivo requirements.txt y asegurar que otros puedan replicar el mismo entorno.

pip install virtualenvwrapper-win

Definición:

Instalamos una herramienta en Windows que facilita la creación y manejo de entornos virtuales.

Importancia:

Simplifica el uso de entornos virtuales en Windows, ya que proporciona comandos como mkvirtualenv, workon, etc.

django-admin startproject nombre_del_proyecto

Definición:

Crea la estructura básica de un nuevo proyecto Django.

Importancia:

Es el punto de partida para comenzar a construir una aplicación web con Django.



📁 Archivos importantes en un proyecto Django

settings.py

Definición:

Archivo de configuración principal del proyecto Django.

Importancia:

Contiene configuraciones de base de datos, rutas estáticas, aplicaciones instaladas, idiomas, zona horaria, seguridad, etc.

Este archivo es clave para controlar el comportamiento global del proyecto recuérdalo siempre.

urls.py

Definición:

Archivo donde se definen las rutas (URLs) que Django debe atender.

Importancia:

Sirve como mapa que conecta las URLs del navegador con las vistas (views) del proyecto.

Podemos definir rutas directamente o incluir rutas de apps específicas.

manage.py

Definición:

Script que actúa como interfaz de línea de comandos para interactuar con nuestro proyecto Django.

Importancia:

Lo usamos para ejecutar comandos como: iniciar el servidor, hacer migraciones, crear usuarios, correr tests, etc.

Ejemplo: `python manage.py runserver`, `python manage.py makemigrations`