

- Configuración de Entorno Virtual (venv)
- 2.1. ¿Por qué un venv?
- 2.2. Creación del venv
- 2.3 Instalar Django
- 3.1. Crear el proyecto Django
- 3.2. Primer arranque del servidor
- 3.3 Crear la primera app
- 3.4. Verificar
- 5.2 Conectar la plantilla con una vista
- 1. Archivo home.html
- 2. Archivo `Definir la vista en accounts/views.py
- 3. Configurar la URL en mi\_proyecto/urls.py
- 4. Probar
  - 5.3 Añadir estilos y scripts básicos

## Configuración de Entorno Virtual (venv)

### 2.1. ¿Por qué un venv?

---

- Aísla las dependencias de Python para evitar conflictos con otras versiones o proyectos.
- Facilita compartir el proyecto con otros (solo deben instalarse las mismas dependencias).

### 2.2. Creación del venv

---

- **Linux / Mac**

```
python3 -m venv venv  
source venv/bin/activate
```

- **Windows**

```
python -m venv .venv
.venv\Scripts\activate
```

Una vez activo, verás **(venv)** en la terminal.

## 2.3 Instalar Django

---

```
pip install django
django-admin --version
```

## 3.1. Crear el proyecto Django

---

- Con el entorno virtual (venv) activado, sitúate en la carpeta donde quieres ubicar tu proyecto y ejecuta:

```
django-admin startproject mi_proyecto
```

- Se generará una carpeta **mi\_proyecto/** con la siguiente estructura:

```
mi_proyecto/
├── mi_proyecto/
│   ├── settings.py
│   ├── urls.py
│   └── ...
└── manage.py
```

## 3.2. Primer arranque del servidor

---

- Entra en la carpeta **mi\_proyecto**:

```
cd mi_proyecto
```

- Inicia el servidor de desarrollo:

```
python manage.py runserver
```

- Abre <http://127.0.0.1:8000/> en tu navegador para ver la pantalla por defecto de Django.

## 3.3 Crear la primera app

---

- Para organizar las funcionalidades, crea una app llamada, por ejemplo, **accounts**:

```
python manage.py startapp accounts
```

- En **settings.py** (dentro de **mi\_proyecto/mi\_proyecto**), busca el bloque **INSTALLED\_APPS** y añade:

```
'accounts',
```

- De ese modo, Django reconocerá la nueva app.

## 3.4. Verificar

---

- Vuelve a ejecutar **python manage.py runserver**.
- Si no aparecen errores, la app ha sido registrada correctamente.
- Seguirás viendo la pantalla inicial de Django hasta que personalices rutas y plantillas.

---

### 5.1 Crear la carpeta **templates**

- Puedes colocarla dentro de la carpeta raíz de tu proyecto (por ejemplo, **mi\_proyecto/templates**) o dentro de la app **accounts** (**accounts/templates**).
- Configura la ruta de plantillas en **settings.py** para que Django sepa dónde buscar los archivos HTML:

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'], # Ajusta según tu estructura
        ...
    },
]
```

---

## 5.2 Conectar la plantilla con una vista

---

### 1. Archivo `home.html`

---

Crea un fichero `home.html` en la carpeta `templates`:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Home Page</title>
</head>
<body>
    <h1>¡Bienvenido a mi proyecto Django!</h1>
</body>
</html>
```

### 2. Archivo `Definir la vista en `accounts/views.py`

---

Crea un fichero `home.html` en la carpeta `templates`:

```
from django.shortcuts import render
```

```
def home(request): return render(request, 'home.html')
```

```
def home(request):
    return render(request, 'home.html')
```

## 3. Configurar la URL en mi\_proyecto/urls.py

Crea un fichero `home.html` en la carpeta `templates`:

```
from django.urls import path
from accounts.views import home

urlpatterns = [
    path('', home, name='home'),
]
```

## 4. Probar

- Crea un fichero `home.html` en la carpeta `templates`:
- Ejecuta `python manage.py runserver`.
- Abre `http://127.0.0.1:8000/` para ver la nueva página.

---

## 5.3 Añadir estilos y scripts básicos

- Crea una carpeta `static/` (por ejemplo, `mi_proyecto/static`) y coloca tu CSS o JS allí.
- Configura `STATIC_URL` en `settings.py`:

```
STATIC_URL = '/static/'
STATICFILES_DIRS = [BASE_DIR / 'static']
```

- En `home.html`, enlaza el CSS:

```
{% load static %}
<link rel="stylesheet" type="text/css" href="{% static 'css/style.css' %}">
```

- Añade un pequeño script para probar que funciona:

```
<script>alert('Hola desde JS');</script>
```