# 📂 Estructura de un Proyecto Django

Cuando creas un proyecto con:

sh

Copiar código

django-admin startproject mi\_proyecto

Se genera la siguiente estructura:

markdown

Copiar código

mi\_proyecto/

│── manage.py

│── mi\_proyecto/

│ │── \_\_init\_\_.py

│ │── settings.py

│ │── urls.py

│ │── asgi.py

│ └── wsgi.py

│── app1/

│ │── migrations/

│ │── \_\_init\_\_.py

│ │── admin.py

│ │── apps.py

│ │── models.py

│ │── tests.py

│ └── views.py

│── db.sqlite3

└── requirements.txt

### ****📌 Explicación de los archivos y carpetas principales****

### ****1️⃣ Archivos raíz del proyecto****

| **Archivo/Carpeta** | **Explicación** |
| --- | --- |
| manage.py | Script para ejecutar comandos de Django (migraciones, servidor, etc.). |
| db.sqlite3 | Base de datos SQLite (si no has configurado otra como MySQL o PostgreSQL). |
| requirements.txt | Lista de dependencias del proyecto para instalar con pip install -r requirements.txt. |

### ****2️⃣ Carpeta Principal del Proyecto (****mi\_proyecto/****)****

| **Archivo** | **Explicación** |
| --- | --- |
| \_\_init\_\_.py | Hace que la carpeta sea tratada como un paquete de Python. |
| settings.py | Configuración del proyecto (base de datos, apps, seguridad, etc.). |
| urls.py | Definición de las rutas de la aplicación. |
| asgi.py | Configuración para servidores ASGI (WebSockets, tiempo real). |
| wsgi.py | Configuración para servidores WSGI (Apache, Gunicorn, etc.). |

### ****3️⃣ Aplicaciones Django (****app1/****)****

Cuando creas una aplicación con:

sh

Copiar código

python manage.py startapp app1

Se genera la siguiente estructura:

| **Archivo** | **Explicación** |
| --- | --- |
| migrations/ | Archivos de migración para la base de datos. |
| admin.py | Configuración para registrar modelos en el panel de administración. |
| apps.py | Configuración de la aplicación. |
| models.py | Definición de las bases de datos (tablas). |
| tests.py | Pruebas automatizadas de la app. |
| views.py | Lógica de las vistas (qué debe hacer cada URL). |

### ****4️⃣ Otras Carpetas Comunes****

Si agregas más configuraciones, puedes tener estas carpetas:

| **Carpeta** | **Explicación** |
| --- | --- |
| static/ | Archivos estáticos como CSS, JS e imágenes. |
| templates/ | Archivos HTML de las vistas. |
| media/ | Archivos subidos por los usuarios (imágenes, documentos). |

## ****🚀 Cómo Funciona Django Internamente****

1️⃣ **El usuario accede a una URL** → Django busca la URL en urls.py.  
2️⃣ **La URL llama a una vista (**views.py**)** → Aquí se ejecuta la lógica del negocio.  
3️⃣ **Si se necesitan datos**, la vista consulta el modelo en models.py.  
4️⃣ **Django renderiza un template** (HTML) y lo envía como respuesta.

# Pasos para crear un superusuario en Django con SQLite:

* **Asegúrate de que tus migraciones estén al día:**

Antes de crear el superusuario, es importante asegurarte de que todas las migraciones se hayan realizado correctamente. Ejecuta el siguiente comando para aplicar las migraciones:

bash

Copiar código

python manage.py migrate

* **Crea un superusuario:**

Una vez que hayas realizado las migraciones, puedes crear el superusuario ejecutando el siguiente comando:

Bash

Copiar código

python manage.py createsuperuser

Este comando te pedirá que ingreses un nombre de usuario, correo electrónico y contraseña para el superusuario. Asegúrate de elegir una contraseña segura.

Ejemplo de la terminal:

pgsql

Copiar código

Username (leave blank to use 'admin'): admin

Email address: admin@example.comPassword: \*\*\*\*\*\*\*\*Password (again): \*\*\*\*\*\*\*\*Superuser created successfully.

* **Accede al panel de administración:**

Después de crear el superusuario, puedes acceder al panel de administración de Django. Si estás corriendo el servidor localmente, abre un navegador y visita la siguiente URL:

arduino

Copiar código

http://127.0.0.1:8000/admin/

Inicia sesión con el nombre de usuario y la contraseña que configuraste al crear el superusuario.

## Solución de problemas comunes:

* Si tienes problemas al crear el superusuario debido a una base de datos corrupta o problemas con las migraciones, puedes intentar eliminar el archivo db.sqlite3 y luego ejecutar de nuevo python manage.py migrate para reiniciar la base de datos.

bash

Copiar código

rm db.sqlite3 # O borra el archivo manualmente

python manage.py migrate

python manage.py createsuperuser

# Crear una aplicación para administrar el sitio principal

Dentro del directorio de tu proyecto, corre el siguiente comando para crear la aplicación:

bash

Copiar código

python manage.py startapp sitio\_principal

Esto creará una carpeta llamada sitio\_principal con la estructura básica de una aplicación Django.

### ****2. Registrar la aplicación en el archivo**** settings.py

Para que Django reconozca la aplicación, abre el archivo settings.py de tu proyecto y agrega 'sitio\_principal' a la lista de INSTALLED\_APPS:

python

Copiar código

INSTALLED\_APPS = [

# otras apps

'sitio\_principal',

]

### ****3. Crear un modelo básico (opcional)****

Si deseas administrar datos relacionados con tu sitio, como noticias, productos o contenido dinámico, crea un modelo en sitio\_principal/models.py. Por ejemplo:

python

Copiar código

from django.db import models

class Noticia(models.Model):

titulo = models.CharField(max\_length=200)

contenido = models.TextField()

fecha\_publicacion = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return self.titulo

Luego, realiza las migraciones:

bash

Copiar código

python manage.py makemigrations

python manage.py migrate

### ****4. Crear vistas para tu sitio (sitio\_principal/views.py)****

Puedes empezar creando algunas vistas para mostrar la información en tu sitio. Ejemplo:

python

Copiar código

from django.shortcuts import renderfrom .models import Noticia

def home(request):

noticias = Noticia.objects.all()

return render(request, 'sitio\_principal/home.html', {'noticias': noticias})

### ****5. Crear una URL para la vista (sitio\_principal/urls.py)****

Ahora, necesitamos crear un archivo urls.py dentro de sitio\_principal para definir las rutas de las vistas. Crea el archivo sitio\_principal/urls.py y agrega:

python

Copiar código

from django.urls import pathfrom . import views

urlpatterns = [

path('', views.home, name='home'),

]

Luego, incluye este archivo urls.py en el archivo principal urls.py del proyecto:

python

Copiar código

from django.contrib import adminfrom django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('sitio\_principal.urls')), # Incluye las rutas de tu aplicación

]

### ****6. Crear un template (sitio\_principal/templates/sitio\_principal/home.html)****

Crea una carpeta templates/sitio\_principal/ dentro de la carpeta sitio\_principal y dentro de esa carpeta, crea un archivo home.html. Aquí puedes agregar el HTML que desees:

html

Copiar código

<!DOCTYPE html><html lang="es"><head>

<meta charset="UTF-8">

<title>Mi Sitio Web Principal</title></head><body>

<h1>Bienvenido a nuestro sitio web</h1>

<ul>

{% for noticia in noticias %}

<li>{{ noticia.titulo }} - {{ noticia.fecha\_publicacion }}</li>

{% endfor %}

</ul></body></html>

### ****7. Ejecutar el servidor y ver la vista****

Ejecuta el servidor de desarrollo:

bash

Copiar código

python manage.py runserver

Accede a http://127.0.0.1:8000/ en tu navegador y deberías ver la página de inicio con las noticias que agregaste.