

Proyecto con Django

Andres Felipe Gonzalez Berdugo

Servicio Nacional De Aprendizaje

Centro Minero

Análisis y desarrollo de software (ADSO)

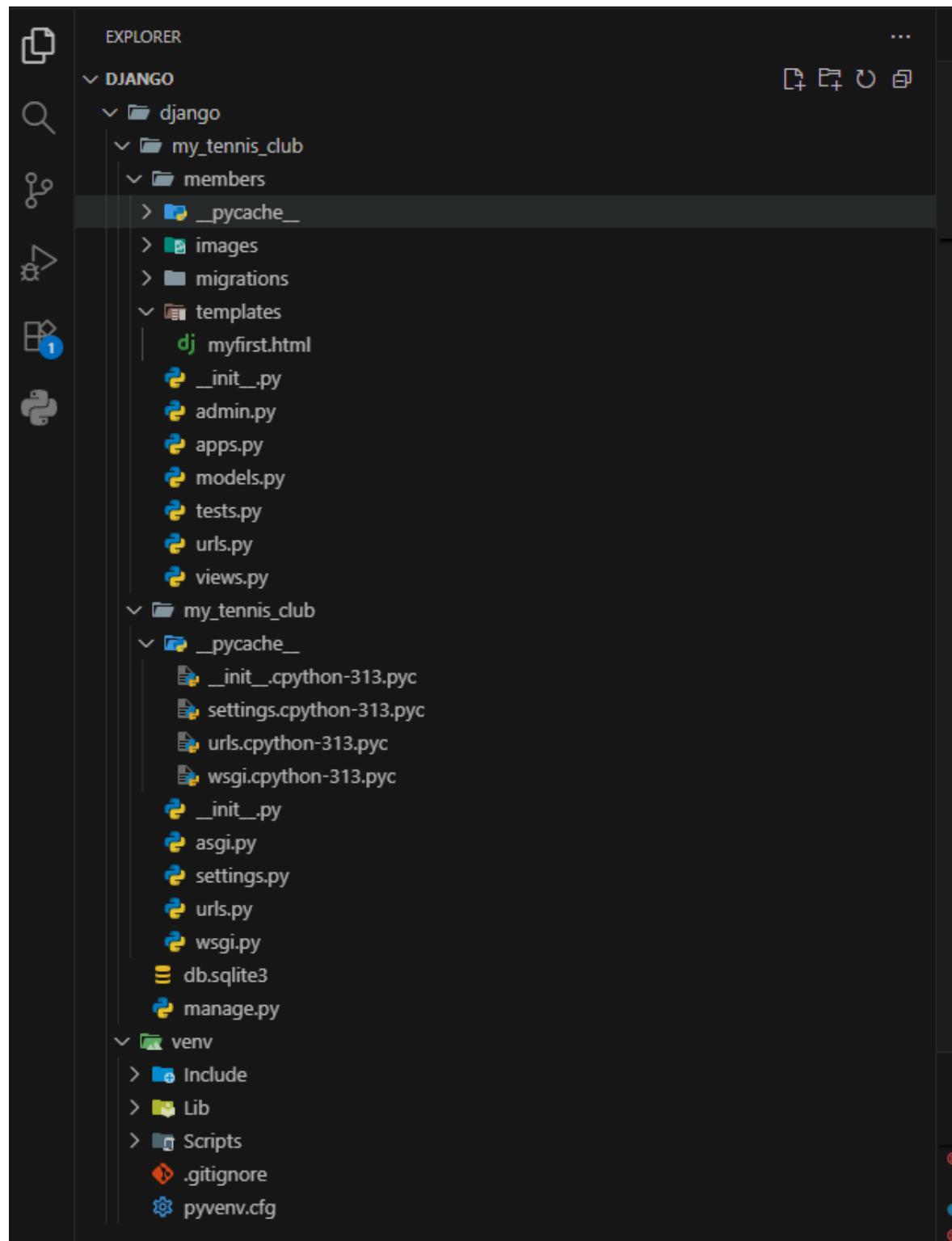
Ficha:2993648

Andrés Felipe Sandoval Higuera

Sogamoso, Boyacá

Noviembre 2025

Orden carpetas Django



La estructura principal que se sigue es que creamos la carpeta principal,

FRAMEWORK_DJANGO y dentro de este se encuentran los siguientes componentes:

Se crea la carpeta llamada my_tennis_club la cual es el proyecto y se organiza de la siguiente forma:

my_tennis_club

manage.py

my_tennis_club/

__init__.py

asgi.py

settings.py

urls.py

wsgi.py

Se crea la carpeta members en la cual es la aplicación y tiene la siguiente organización:

my_tennis_club

manage.py

my_tennis_club/

members/

migrations/

__init__.py

__init__.py

admin.py

apps.py

models.py

tests.py

views.py

Código HTML

![Screenshot of a code editor showing Django project structure and code. The code editor shows a file named 'myfirst.html' with the following content:
<pre>
119 <body>
120 <div id=](https://www.youtube.com/embed/srkHntDf1I4 "Video Explicativo de Django")
The terminal tab shows a git commit command:

```

$ git commit -m "se suben los archivos de trabajo"
[master 9111010560] MNG464 ~/Desktop/django [master]
create mode 100644 django/my_tennis_club/my_tennis_club/_pycache_/_wsgi.cpython-313.pyc
create mode 100644 django/my_tennis_club/my_tennis_club/wsgi.py
create mode 100644 django/my_tennis_club/my_tennis_club/settings.py
create mode 100644 django/my_tennis_club/my_tennis_club/migrations.py

```

Este es el código HTML de la página en la cual se podrá visualizar información acerca de Django

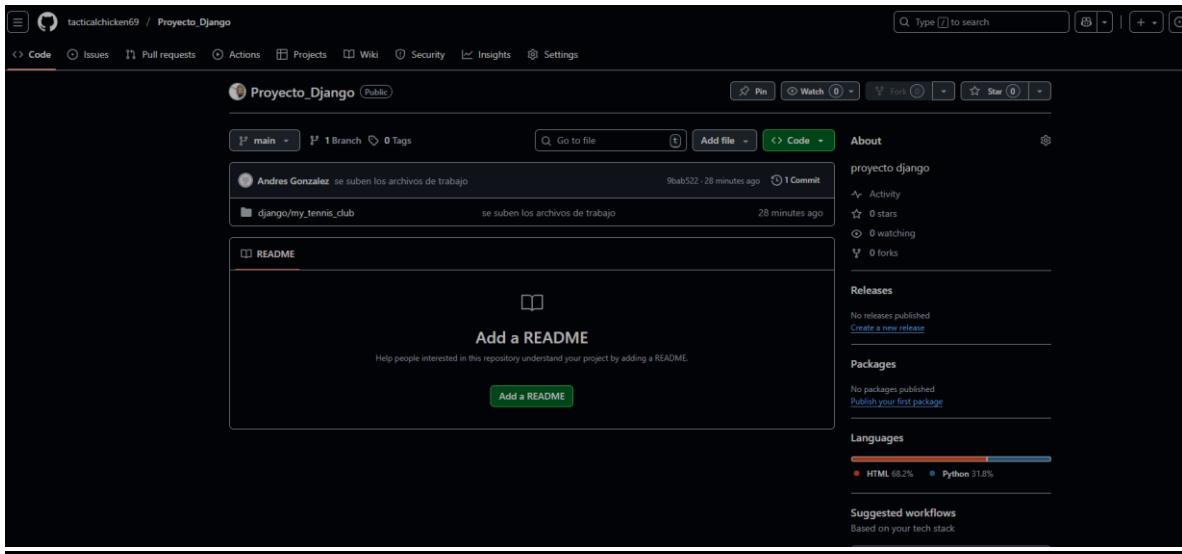
Visualización de HTML

La función principal de Django es servir como un framework web de alto nivel en Python que facilita el desarrollo rápido, seguro y escalable de aplicaciones y sitios web siguiendo con un patrón MVC.

- **Modelo:** Controla el comportamiento de los datos.
- Define los datos almacenados.
- Se encuentra en forma de clases de Python (el modelo es un objeto).
- Cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros.
- Posee métodos para la gestión de los datos.
- **Vista:** Se presenta en forma de funciones o clases en Python
- su propósito es determinar qué datos serán visualizados.
- El ORM de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista. La vista también se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Lo más importante a entender con respecto a la vista es que no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla.
- **Plantilla (Template/Controlador):** Es básicamente una página HTML con algunas etiquetas extrañas propias de Django, no solamente crea contenido en HTML (también XML, CSS, Javascript, CSV, etc.).
- La plantilla (Template) recibe los datos de la vista y luego los organiza para la presentación al navegador web. Las etiquetas que Django usa para las plantillas permiten que sea flexible para los diseñadores del frontend, incluso tiene estructuras de datos como if, por si es necesaria una presentación lógica de los datos, estas estructuras son limitadas para evitar un desorden poniendo cualquier tipo de código Python.

En este pantallazo se da un vistazo a la página web al ser ejecutada

Imagen repositorio GITHUB



En este repositorio de git hub se encuentran los archivos del proyecto, a continuación el link del repositorio https://github.com/tacticalchicken69/Proyecto_Django.git