

**Configuracion del proyecto:**

Paso 1: Crear una carpeta

Ingresar a la carpeta donde quieres tu proyecto mediante el commad prompt de windonws

Pip install virtualenv : instalaciones

Para corroborar:

Virtualenv –version

Virutalenv venv : Crear el entorno virutal

.\venv\Scripts\active

Python –version

Pip --vesion

Ingresar a visual stuidio arrastrando tu proyecto

F1 + Select interpreter

Seleccionar python version + (´venv´. Venv)

CREATE NEW TERMINAL

Crear Django

Pip install django

Comprobar las instalaciones:

Django-admin –version

Python -m django --version

Crear un nuevo proyecto

Django-admin startproject “nombre” .

Django-admin startproject mysite .



Python manage.py runserver : Sale consola con el servidor, creándote una base sql, te sale un puerto por defecto pero si esta ocupado debes asignarle uno nuevo

Python manage.py runserver 1000

Para salir control + c

Python manage.py –help : muestra comandos básicos

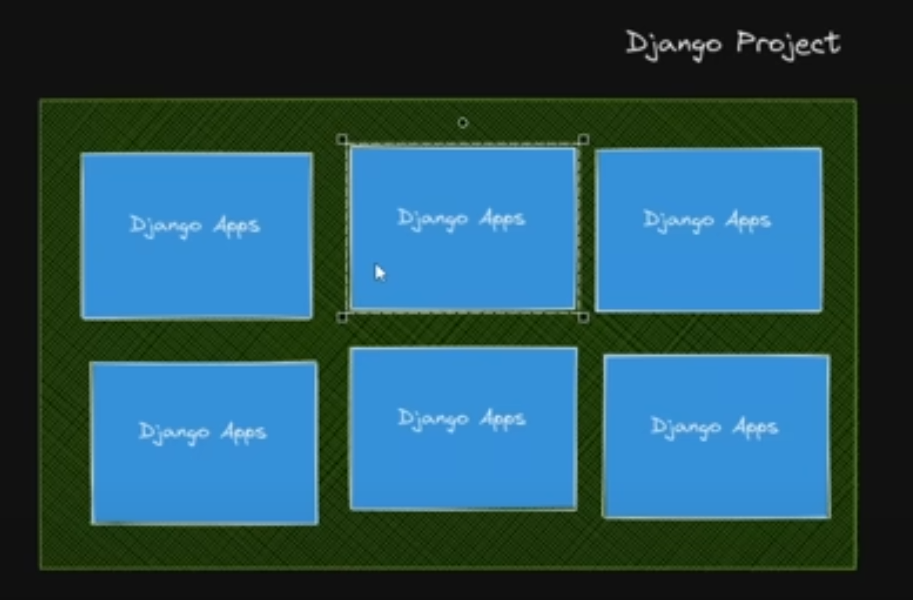
**Estructura de las carpetas:**

1. **Mysite/**: Es la carpeta principal de tu proyecto Django, que contiene la configuración y componentes clave del proyecto.

* **\_\_\_pycache\_\_/**: Carpeta que contiene código compilado en Python para mejorar la velocidad de ejecución del proyecto.
* **\_\_init\_\_.py**: Archivo vacío que le indica a Python que esta carpeta debe tratarse como un paquete. Permite importar archivos de este directorio.
* **asgi.py**: Archivo que define la configuración de la interfaz de servidor ASGI (Asynchronous Server Gateway Interface), utilizado para manejar peticiones web asincrónicas.
* **settings.py**: Archivo de configuración principal del proyecto. Aquí defines las configuraciones de base de datos, aplicaciones instaladas, rutas de archivos estáticos, entre otras.
  + **Nota**: Modo producción/ modo desarrollo
    - **Se activa en**:
      * **Debug** = True (desarrollo)
      * **Debug = False (producción)**
  + 
* **urls.py**: Archivo donde defines todas las rutas o URLs del proyecto, es decir, qué vistas se ejecutan cuando se accede a una URL específica.
* **wsgi.py**: Define la configuración para la interfaz de servidor WSGI (Web Server Gateway Interface), usada para desplegar aplicaciones Django en servidores web.

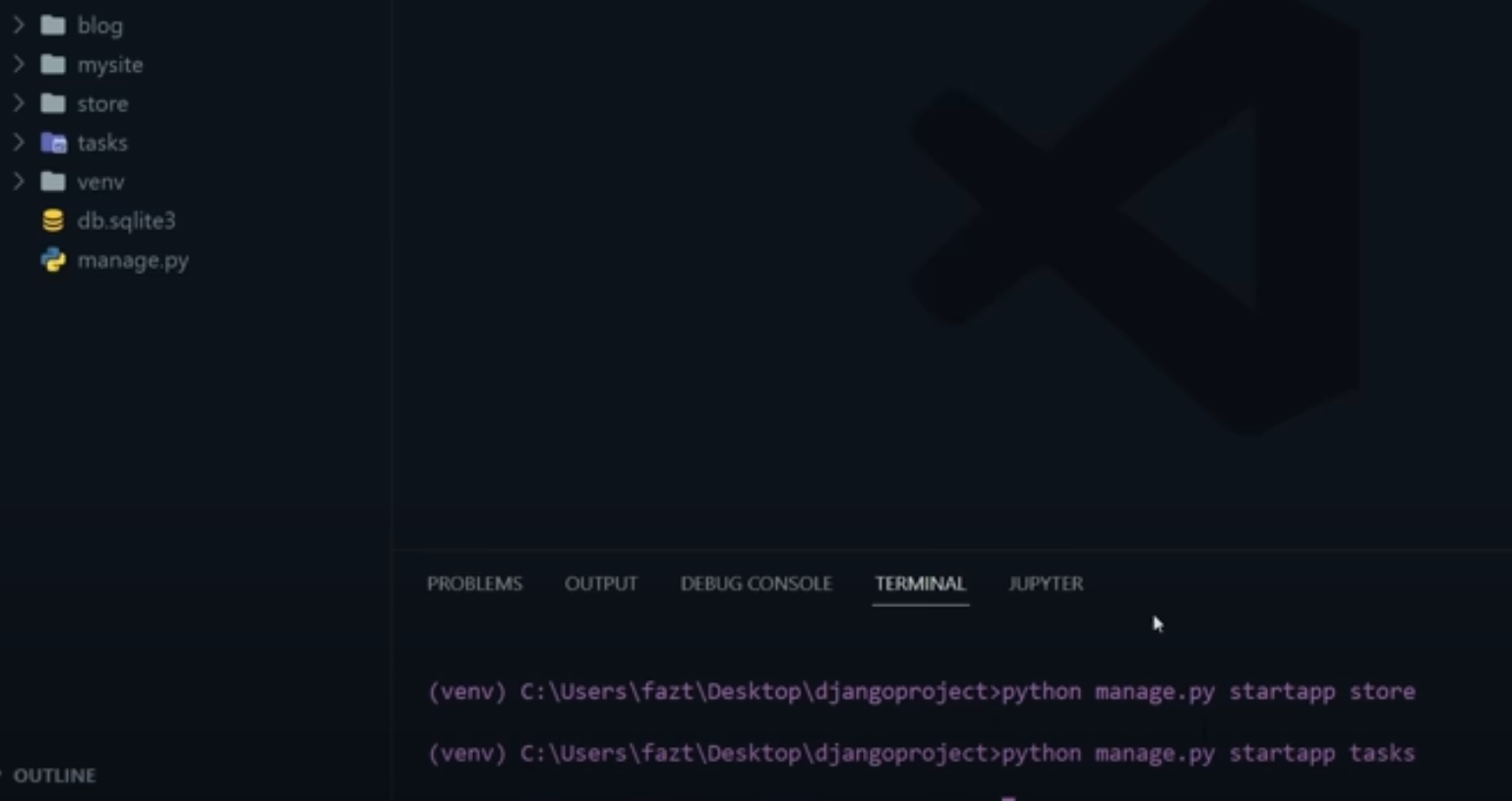
1. **venv/**: Carpeta que contiene el entorno virtual de Python. En este entorno virtual se instalan todas las dependencias y bibliotecas necesarias para tu proyecto sin afectar las instalaciones globales de tu sistema.
2. **db.sqlite3**: Base de datos predeterminada que utiliza SQLite. Esta es una base de datos ligera y sencilla que Django crea automáticamente para almacenar los datos de tu aplicación.
3. **manage.py**: Archivo de utilidad que se utiliza para interactuar con el proyecto. Permite ejecutar comandos importantes, como iniciar el servidor de desarrollo, crear bases de datos, migrar cambios en los modelos, entre otros.

**Aplicaciones en Django**



Python manage.py startapp “nombre”

Ejemplos:



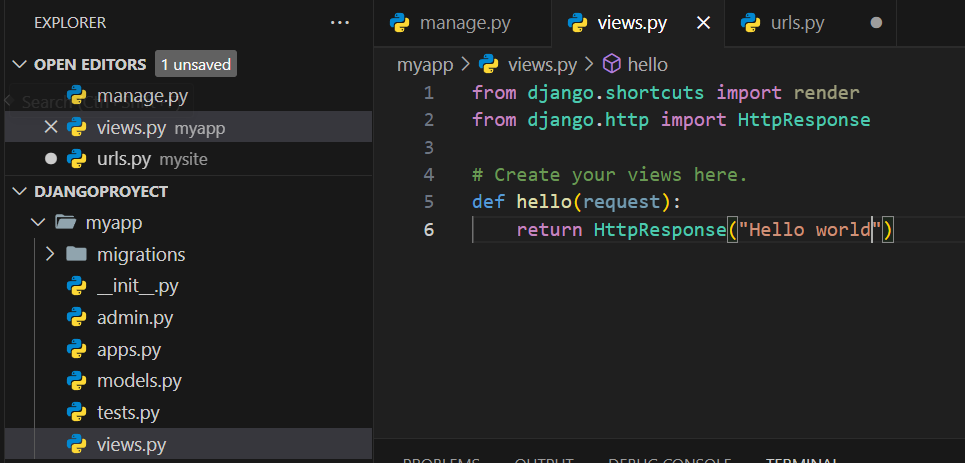
ESTRUCURA DE LA APLICACIÓN

**myapp/**: Esta es la carpeta de la aplicación que acabas de crear. En Django, las aplicaciones son componentes modulares que puedes reutilizar en diferentes proyectos o en múltiples instancias dentro de un mismo proyecto.

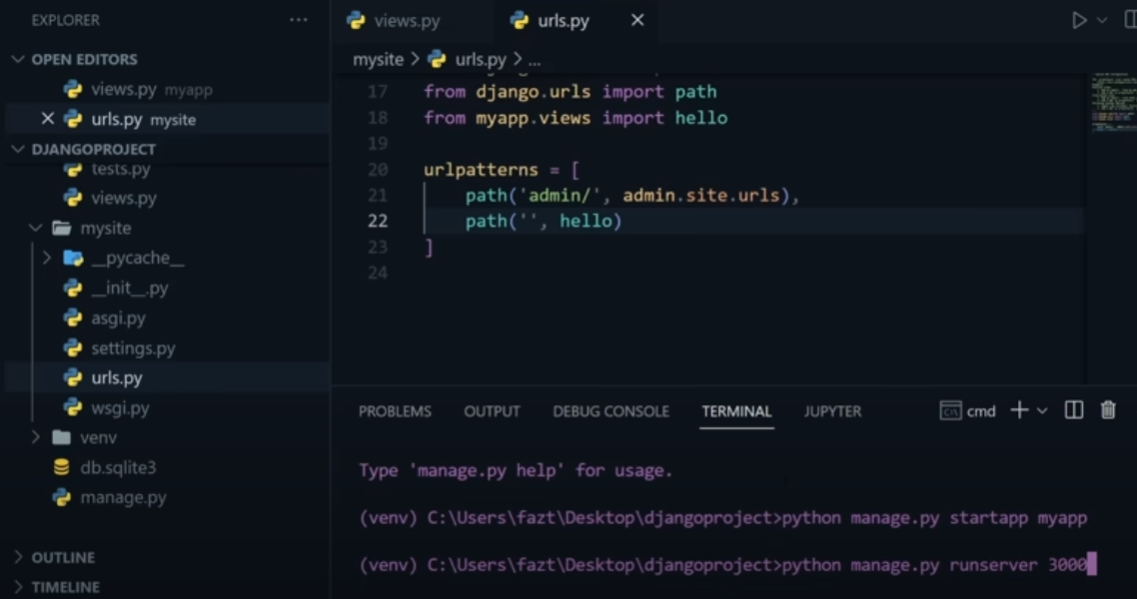
1. **migrations/**: Carpeta que contiene los archivos relacionados con las migraciones de la base de datos, que son los cambios realizados a los modelos (esquemas de base de datos) en la aplicación.
   * 1. **\_\_init\_\_.py**: Archivo que le indica a Python que esta carpeta debe tratarse como un paquete. Inicialmente vacío, pero necesario para el manejo del paquete.
   * **\_\_init\_\_.py**: Similar al archivo de \_\_init\_\_ en otras carpetas, señala que el directorio myapp es un paquete de Python, permitiendo importar archivos desde él.
2. **admin.py**: Archivo donde puedes registrar los modelos para que sean gestionados a través del panel de administración de Django. Aquí defines cómo se verán y se gestionarán los datos de tu aplicación desde la interfaz administrativa.
3. **apps.py**: Contiene la configuración de la aplicación. Define el nombre de la aplicación y otra configuración opcional. Este archivo se utiliza en la configuración del proyecto (por ejemplo, en INSTALLED\_APPS dentro de settings.py).
4. **models.py**: Archivo donde defines los modelos de datos de la aplicación, que son clases de Python que representan y gestionan las tablas de la base de datos.
5. **tests.py**: Archivo donde defines pruebas o tests automáticos para verificar que las diferentes funcionalidades de la aplicación funcionan correctamente.
6. **views.py**: Archivo donde defines las vistas de la aplicación. Las vistas son las funciones o clases que determinan qué contenido se muestra al usuario cuando accede a una URL específica de la aplicación.

**HELLO WORLD**

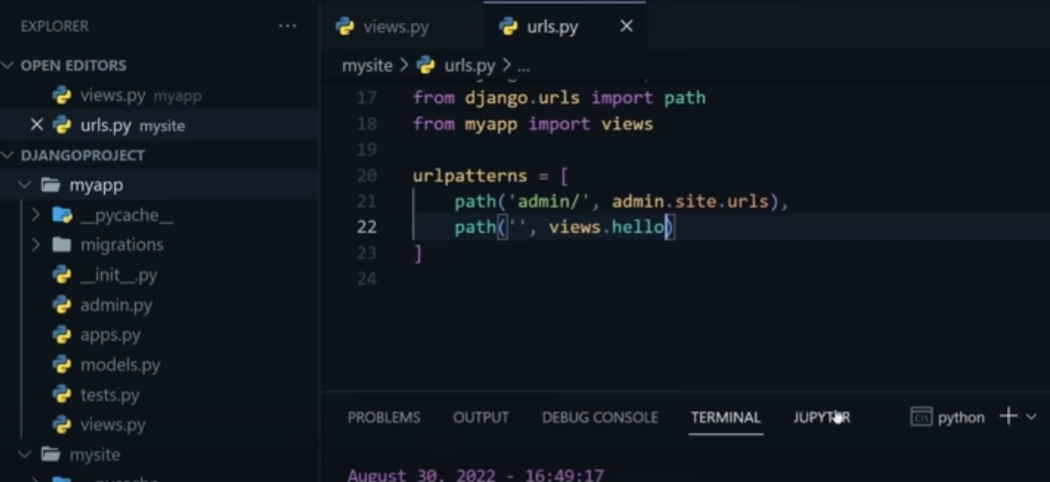
FUNCION PARA CREAR EL MENSAJE



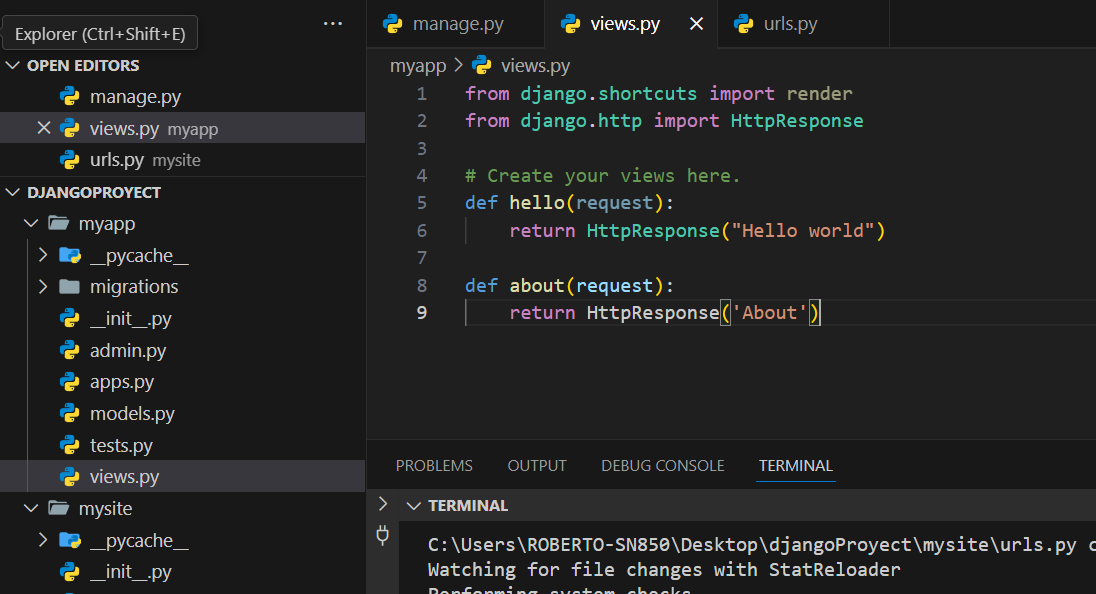
REFERENCIA EN LA PARTE CENTRAL

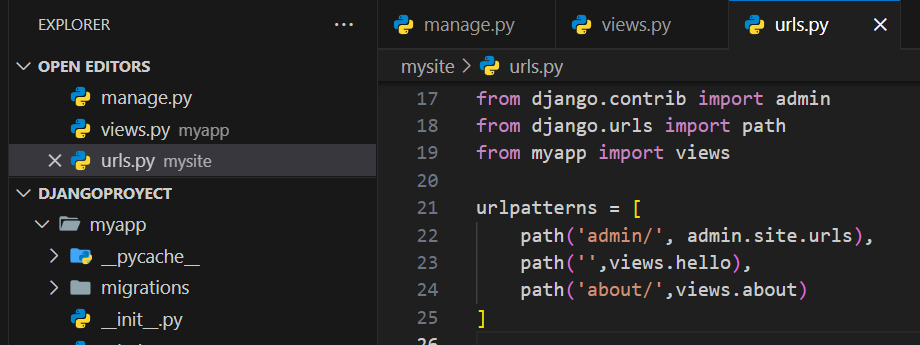


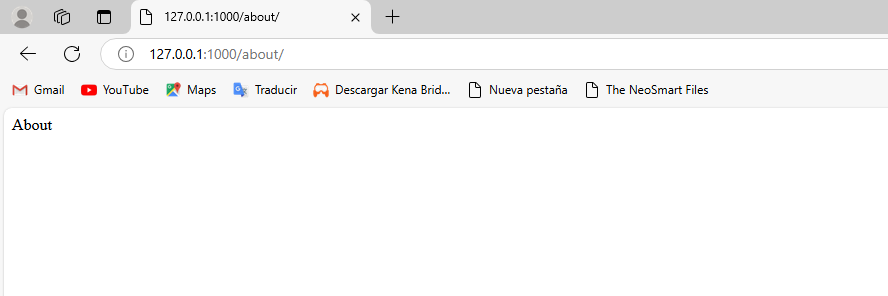
OTRA FORMA DE HACER EL LLAMADO:



MENSAJE ENRUTADO:

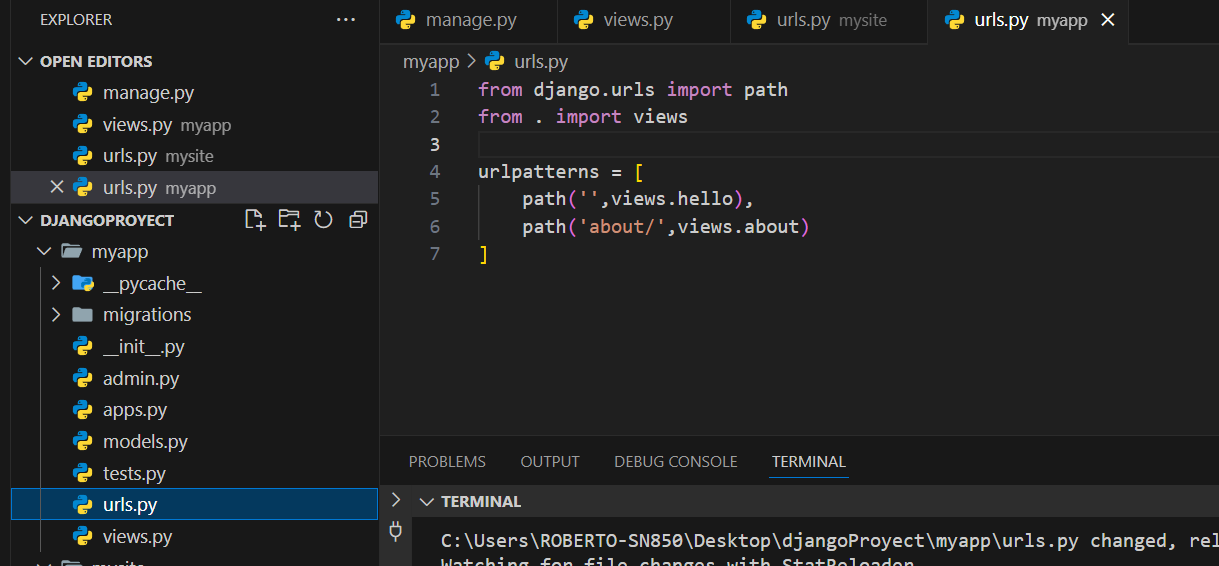




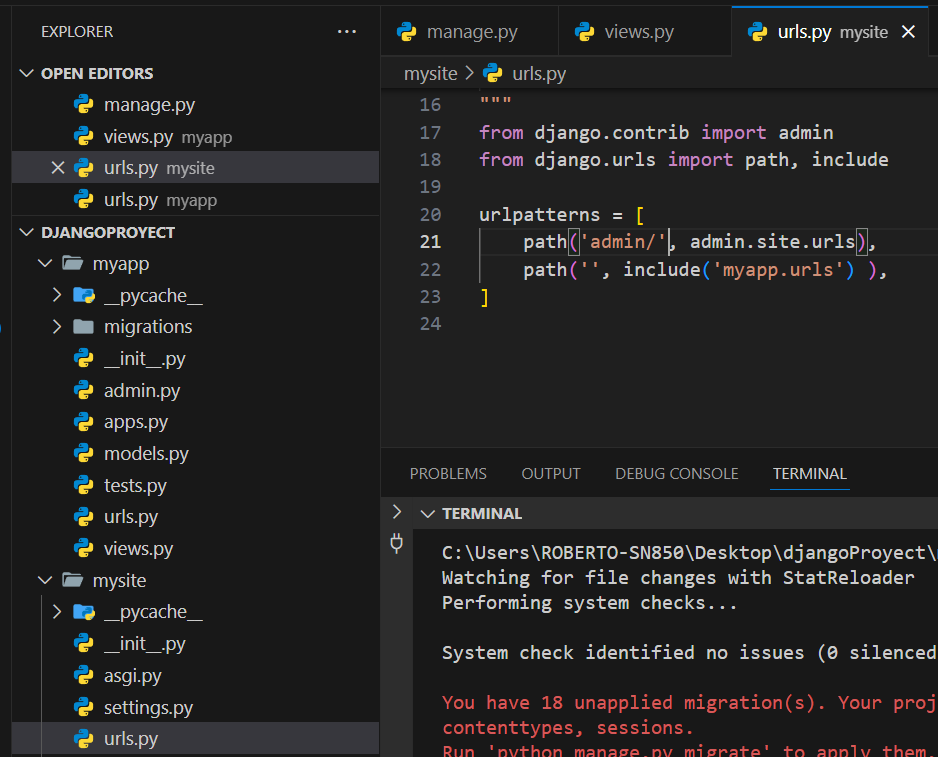


EN EL EJEMPLO ANTERIOR SE VISUALISA LA FORMA EN QUE MYAPP INTERACTURA CON MYSITE DE MANERA INEFICAZ DEBIDO A QUE LAS LLAMADAS LAS HACE EN MYSITE. LA INFECIENCIA VIENE QUE SI ES QUE EXISTE MUCHAS APPS LAS LLAMADAS SE SATURARIAN EN MYSITE. EN ESTE EJEMPLO UTILIZAREMOS UN ARCHIVO URLS.PY DENTRO DE MYPP PARA MEJORAR EL ORDEN Y EFICIENCIA

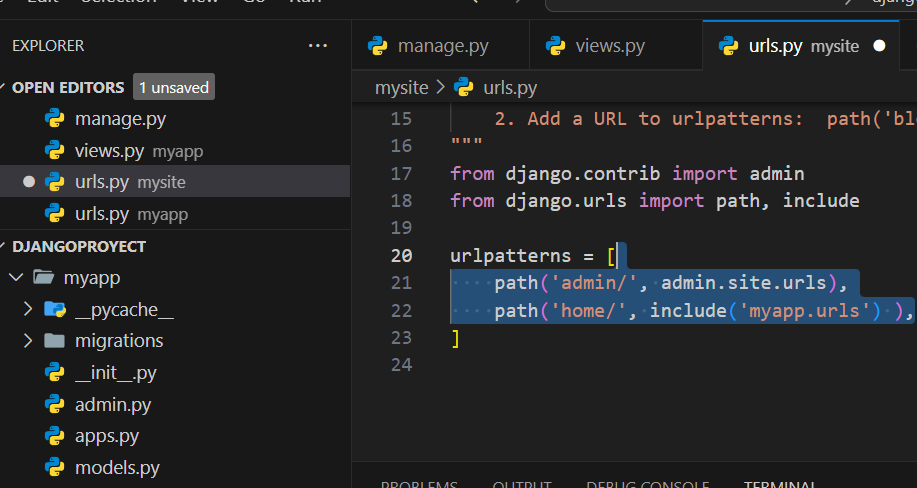
CREAR urls.py E IMPORTAR LAS LIBRERIAS, posterior hacer las referencias

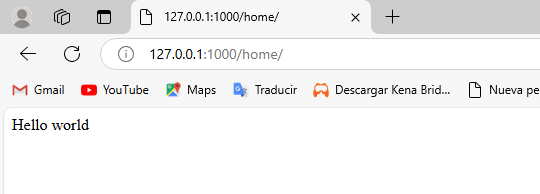


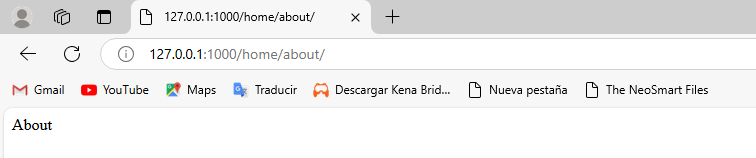
MODIFICAR LAS REFERENCIAS EN MYSITE



**USO DE PREFIJO PARA ENRUTAR LAS FUNCIONES ANTERIORMENTE DESCRITAS**

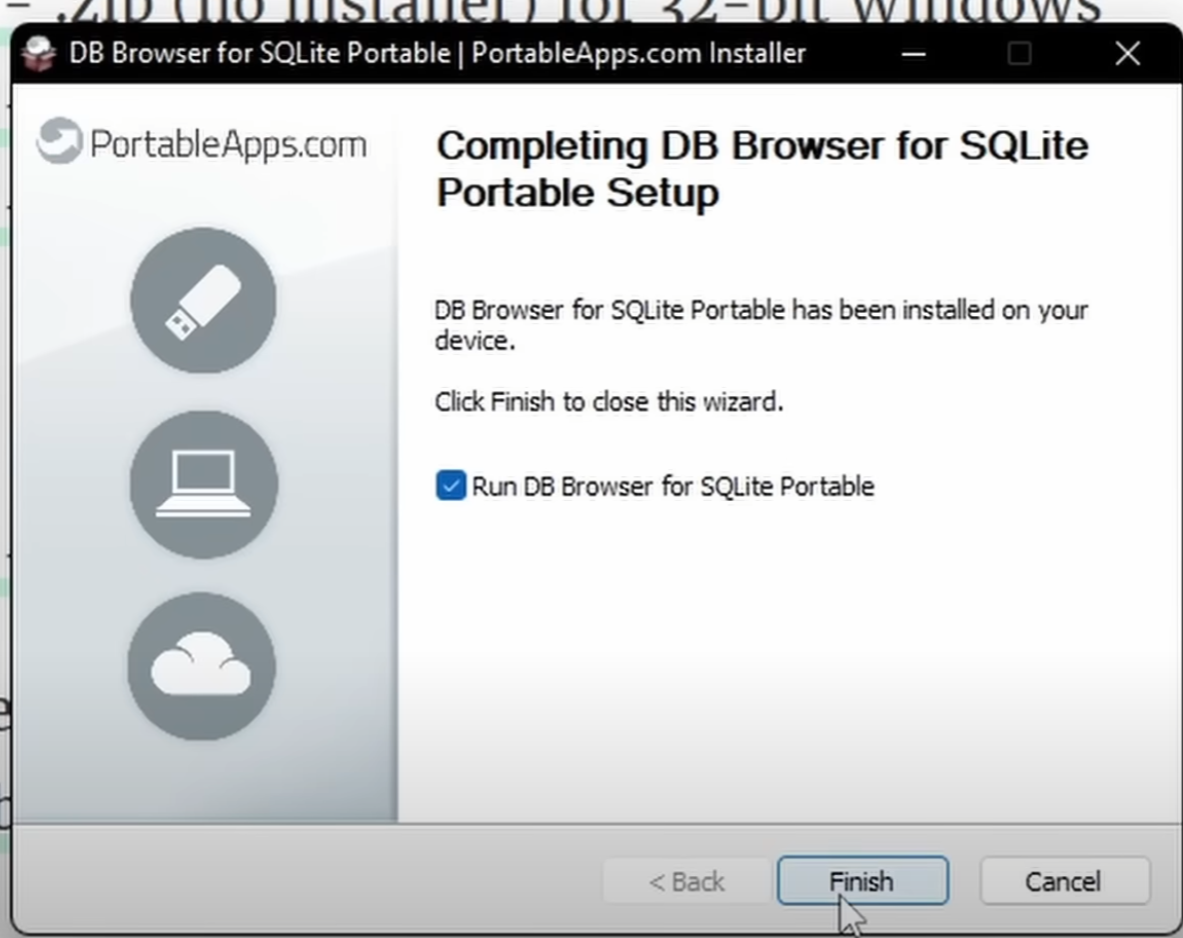


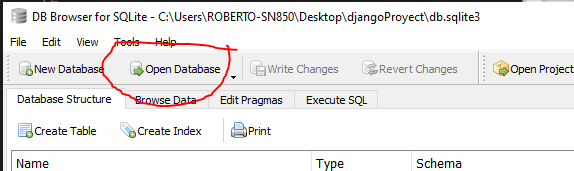




**DB SQLITE**

Descarga y configura acepta. Todo.

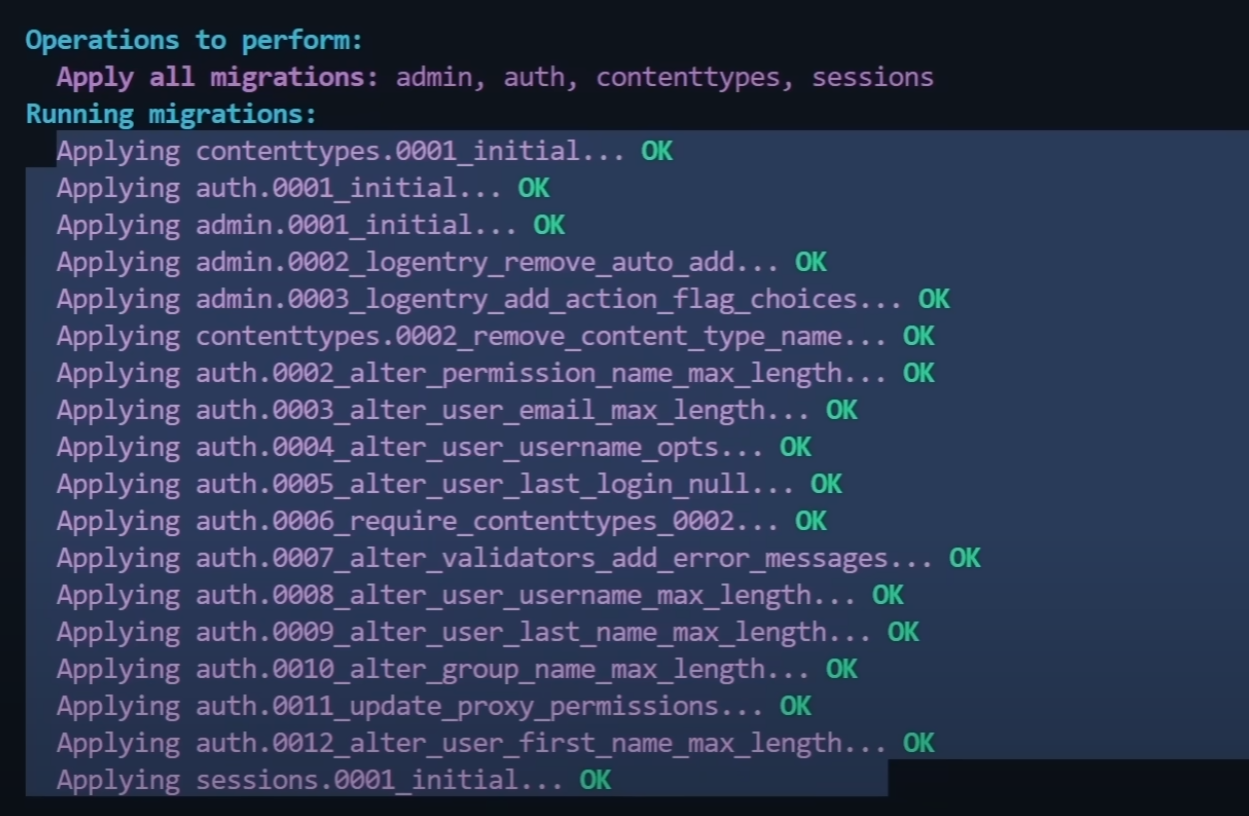




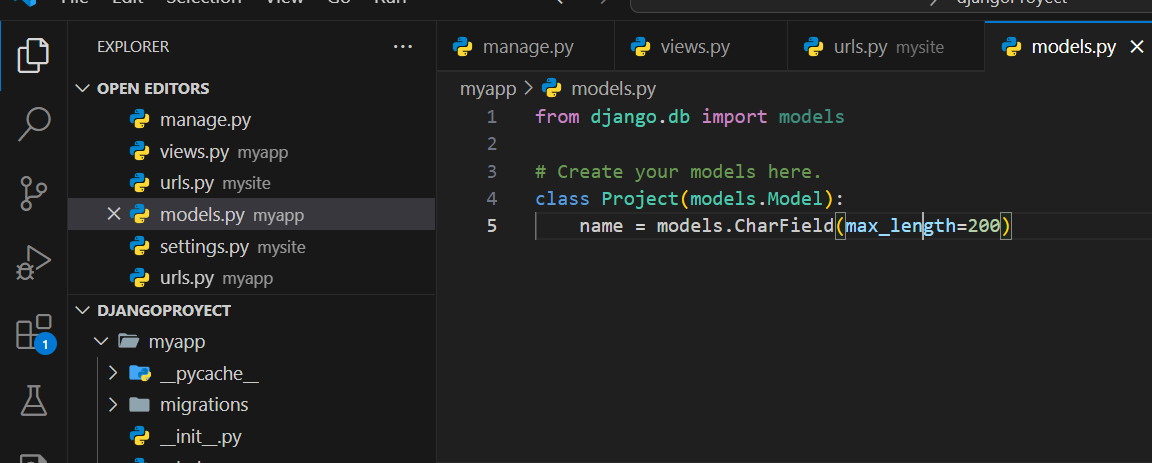
BUSCA TU PROYECTO Y SEÑALA LA BASE DE DATOS CREEDA EN ELLA

Python manage.py makemigrations

Python manage.py migrate

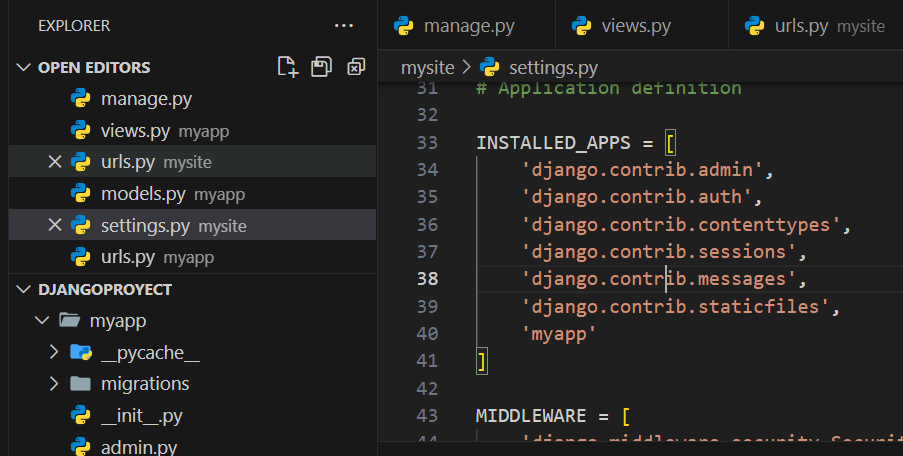


Crea una base de datos en myapp



python manage.py runserver

Guardar la referencia de las actualización de myapp en mysite



UNA VEZ RELACIONADO, EJECUTAMOS:

Python manage.py makemigrations

O Python manage.py makemigratios “referencia”

Ejemplos: Python manage.py makemigratios myapp

