**1. Crea un entorno virtual**

Desde la terminal, navega a la carpeta raíz de tu proyecto y ejecuta:



**python -m venv venv**

Esto creará una carpeta llamada venv en tu proyecto donde se almacenará el entorno virtual.

**2. Activa el entorno virtual**

Según el sistema operativo que estés usando, la activación es diferente:

* Windows



**.\venv\Scripts\activate**

* Linux / macOS

**source venv/bin/activate**

Si se activa correctamente, deberías ver algo como (venv) al inicio de la línea en la terminal.

**3. Instala las dependencias**

Una vez activado el entorno virtual, dirigete al archivo principal del proyecto e instala las dependencias con: 



pip install -r requirements.txt

**Descripción de cada archivo y carpeta**

|  |  |
| --- | --- |
| Archivo/Carpeta | Propósito |
| app/\_\_init\_\_.py | Inicializa la aplicación Flask y la base de datos. |
| app/models.py | Define las clases del modelo SQLAlchemy. |
| app/routes.py | Define las rutas y la lógica para la API y las vistas. |
| app/templates/ | Contiene las plantillas HTML para las vistas. |
| app/static/ | Contiene los archivos estáticos como CSS y JS. |
| .env | Almacena las credenciales de Supabase de manera segura. |
| config.py | Configura la conexión con la base de datos y otras opciones de Flask. |
| requirements.txt | Lista las dependencias para instalar con pip. |
| run.py | Archivo para arrancar el servidor Flask. |

**✅ ¿Por qué esta estructura es mejor?**

✔️ Está organizada por **módulos** (más fácil de mantener y escalar).  
✔️ Mantiene los archivos estáticos y las plantillas en sus propias carpetas.  
✔️ Sigue el patrón **MVC (Modelo-Vista-Controlador)**, que es una práctica común en proyectos Flask.