INFORME DE ACTIVIDAD

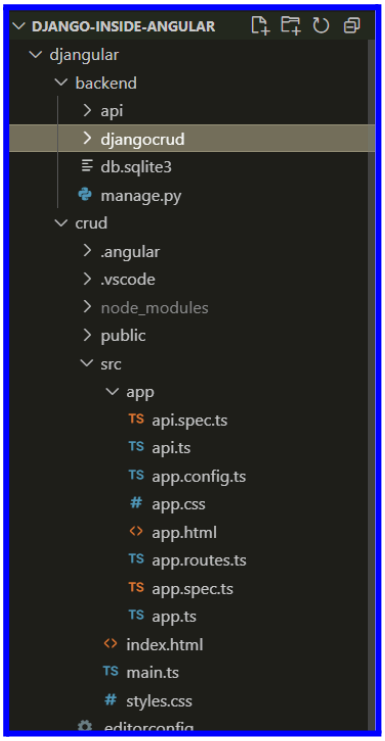
INFORMACIÓN BÁSICA					
ASIGNATURA:	**PROGRAMACIÓN WEB 2**				
TÍTULO DE LA PRÁCTICA:	*15 - Djangular*				
NÚMERO DE PRÁCTICA:	*15*	**AÑO LECTIVO:**	*2025 - A*	**NRO. SEMESTRE:**	*III*
FECHA DE PRESENTACIÓN	*08/06/2025*	**HORA DE PRESENTACIÓN**	*23:59:00*		
INTEGRANTE (s): *Layme Salas, Rodrigo Fabricio*				**NOTA:**	
DOCENTE: *Mag. Corrales Delgado Carlo José Luis*					

SOLUCIÓN Y RESULTADOS

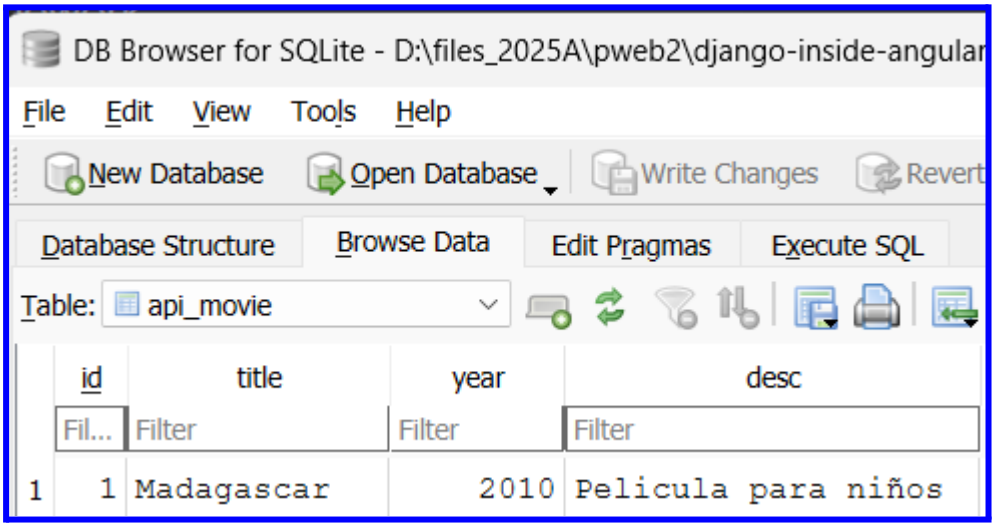
I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS

LINK GITHUB DE LA ACTIVIDAD:
https://github.com/F4brici0L4yme/django-inside-angular.git

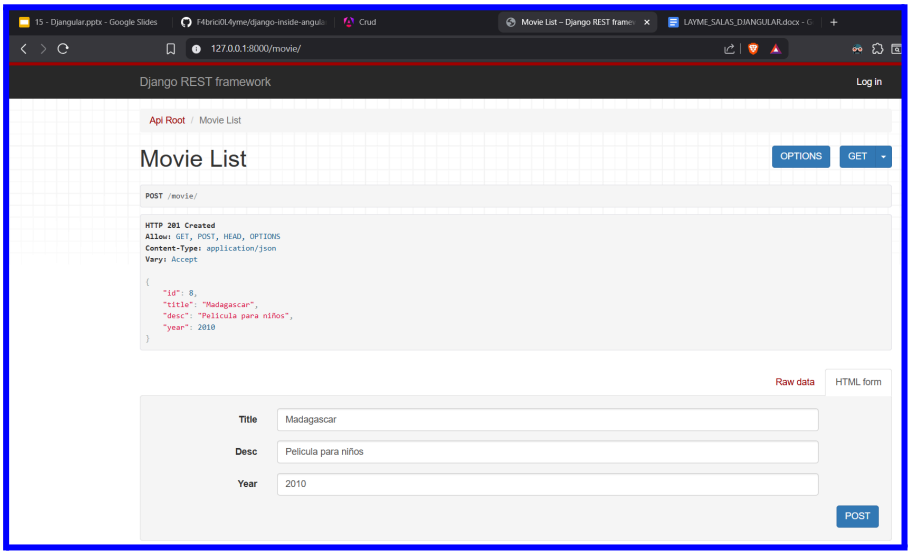
Vistas de los pasos seguidos en la guía



Así organicé el proyecto, backend almacena a django y crud a angular (frontend)



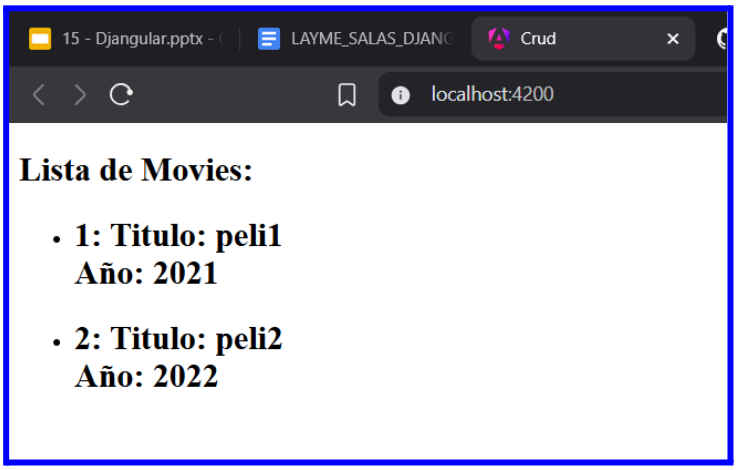
Esta es la DB que queda después de crear el modelo y la api



Con serializer se crea esta vista, para operaciones de GET, POST, etc.

```
13  export class App {
14    movies = [{ id: 1, title: 'peli1', year: 2021 }, { id: 2, title: 'peli2', year: 2022 }];
15
16    constructor(private api: ApiService) {
17      this.getMovies();
18    }
19
20    getMovies = () => {
21      this.api.getAllMovies().subscribe(
22        data => {
23          console.log(data);
24          //this.movies = data
```

En app.ts se exporta esta información para app.html



Se muestra el arreglo creado

```
20  getMovies = () => {
21    this.api.getAllMovies().subscribe(
22      data => {
23        console.log(data);
24        this.movies = data
```

Descomentamos la línea que reemplaza el arreglo data por la de la DB

Ahora se muestran nuestros datos de sqlite, la información colocada en django se muestra en el frontend

```
(venv) D:\files_2025A\pweb2\django-inside-angular\djangular\crud>git log --graph --pretty=oneline --abbrev-commit --all
* c1d5484 (HEAD -> main, origin/main) elimina archivos pasados - nuevos archivos con nombres establecidos en la guía
* 46a5535 crea html para app - lista las peliculas con valores colocados en app.ts
* 810cb67 genera api y conecta con app.ts para recibir solicitudes http
* c715376 agrega corsheaders a installedApps - MovieViewSet creado - actualiza urls para registrar movie db al frontend
* 8709841 crea serializer - modelo movie - agrega a installed_apps api y rest_framework
* 539e38d incializa proyecto djangodruc y app api en backend
* 97eb57f agregar gitignore - inicializa los proyectos angular y django
```

Lista de commits final con el comando: ***git log --graph --pretty=oneline --abbrev-commit --all***

II. CONCLUSIONES

Django es perfecto para crear servicios de api y Angular es un gran framework para tratar esos datos dada la experiencia que tenemos con JavaScript-TypeScript. Su combinación es ideal por la Db de django, apis y la organización de archivos de Angular.

RETROALIMENTACIÓN GENERAL

Seguiremos usando Angular y Django para la creación de web-apps y en especial para el proyecto final.

REFERENCIAS Y BIBLIOGRAFÍA

- ***[1]*** *Angular, "Angular – Angular", angular.dev. [En línea]. Disponible en: https://angular.dev/.*
- ***[2]*** *Angular, "Learn Angular - Tutorials", angular.dev. [En línea]. Disponible en: https://angular.dev/tutorials/learn-angular.*