

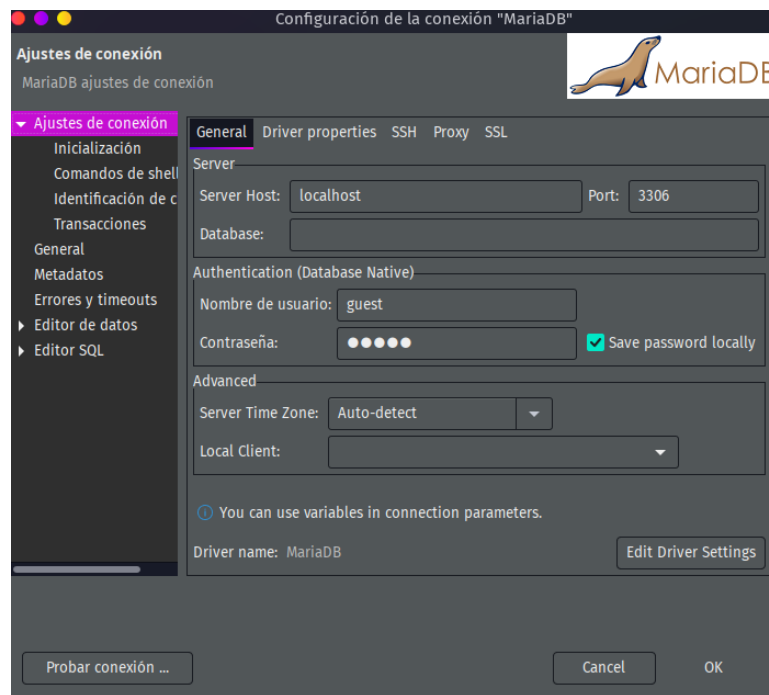
Video Club – Base de Datos

Estudiantes:

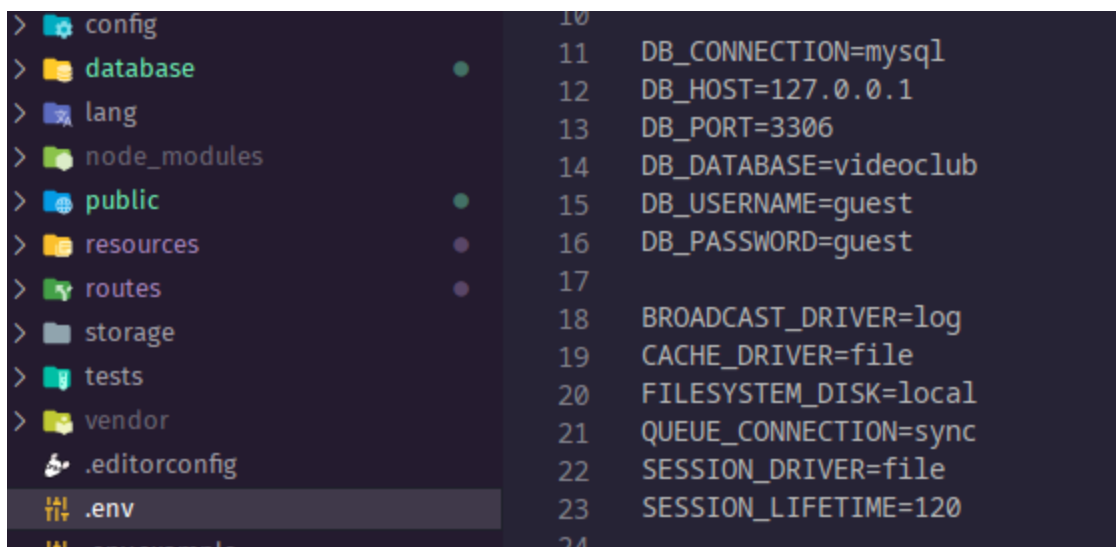
Daniel Estiven Guancha

Edward Malte

1. Se crea la base de datos en MySQL con el nombre de videoclub y el usuario que administrara esta base de datos sera guest y se la administra con Dbeaver



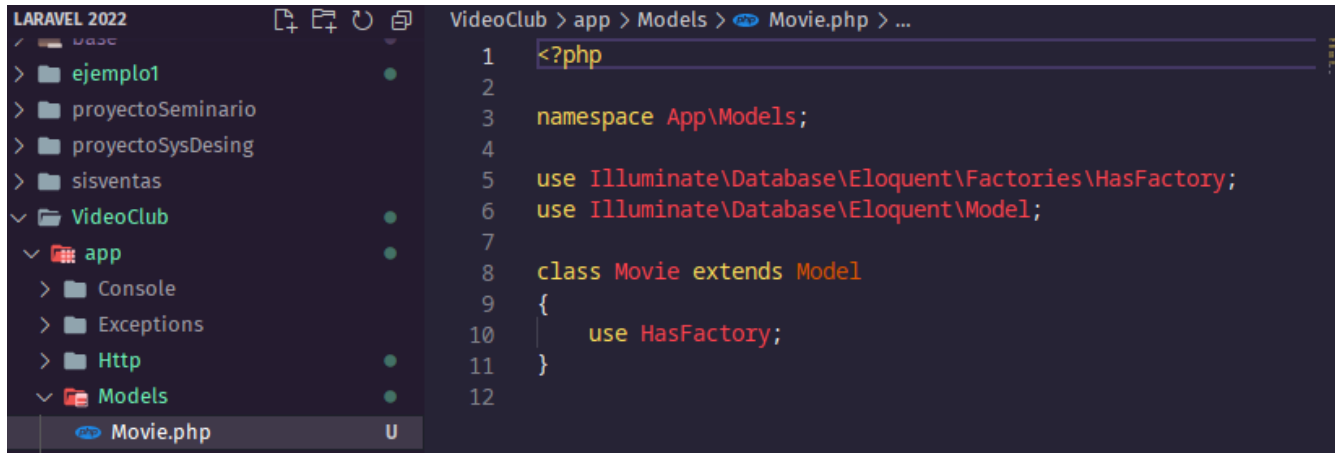
se configura el archivo llamado .env del proyecto video club y se configura los parametros de conexión con labase de datos



con el siguiente comando se crea el modelo Movie .

```
danielstvn@danielstvn in repo: VideoClub on main2 via  
took 451ms  
λ php artisan make:model Movie
```

Creara el siguiente Modelo el cual se utilizara a continuación.

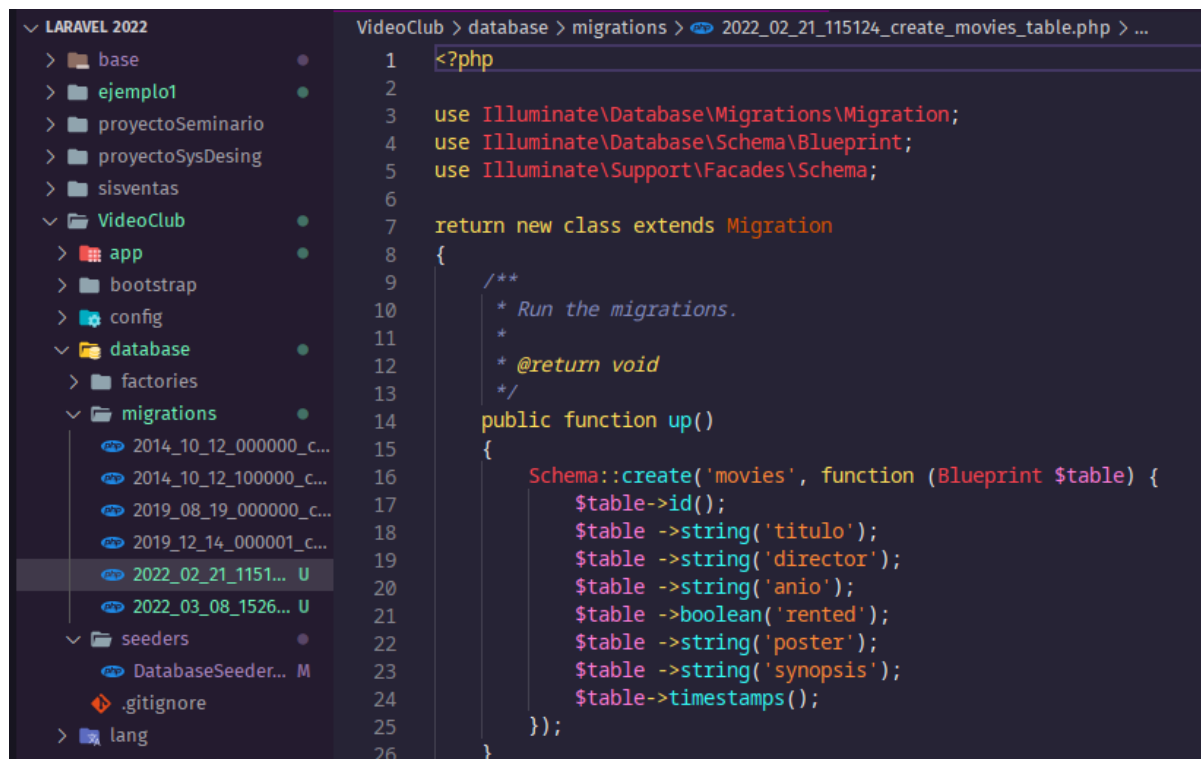


```
VideoClub > app > Models > Movie.php > ...  
1 <?php  
2  
3 namespace App\Models;  
4  
5 use Illuminate\Database\Eloquent\Factories\HasFactory;  
6 use Illuminate\Database\Eloquent\Model;  
7  
8 class Movie extends Model  
9 {  
10     use HasFactory;  
11 }  
12
```

Con el siguiente comando se crea la migración.

```
danielstvn@danielstvn in repo: VideoClub on main2  
via took 451ms  
λ php artisan make:migration create_movies_table
```

Se creara un archivo, en el cual se especifica la informacion que tendra la tabla que se creara en la base de datos, a continuacion se indica los campos que requeridos para este ejercicio.



```
VideoClub > database > migrations > 2022_02_21_115124_create_movies_table.php > ...  
1 <?php  
2  
3 use Illuminate\Database\Migrations\Migration;  
4 use Illuminate\Database\Schema\Blueprint;  
5 use Illuminate\Support\Facades\Schema;  
6  
7 return new class extends Migration  
8 {  
9     /**  
10      * Run the migrations.  
11      *  
12      * @return void  
13      */  
14     public function up()  
15     {  
16         Schema::create('movies', function (Blueprint $table) {  
17             $table->id();  
18             $table->string('titulo');  
19             $table->string('director');  
20             $table->string('anio');  
21             $table->boolean('rented');  
22             $table->string('poster');  
23             $table->string('synopsis');  
24             $table->timestamps();  
25         });  
26     }  
27 }
```

Con el siguiente comando se crea la migración en la base de datos

```
danielstvn@danielstvn in repo: VideoClub on □ main2 via □ took 451ms  
λ php artisan make:migration
```

en la base de datos aparecera la tabla movies con los campos especificados en la migración.

The screenshot shows a database management interface with a sidebar on the left displaying a tree view of the database structure. The 'movies' table is highlighted under the 'videoclub' database. The main panel on the right shows the 'Propiedades' (Properties) tab for the 'movies' table. The table is configured with the following settings:

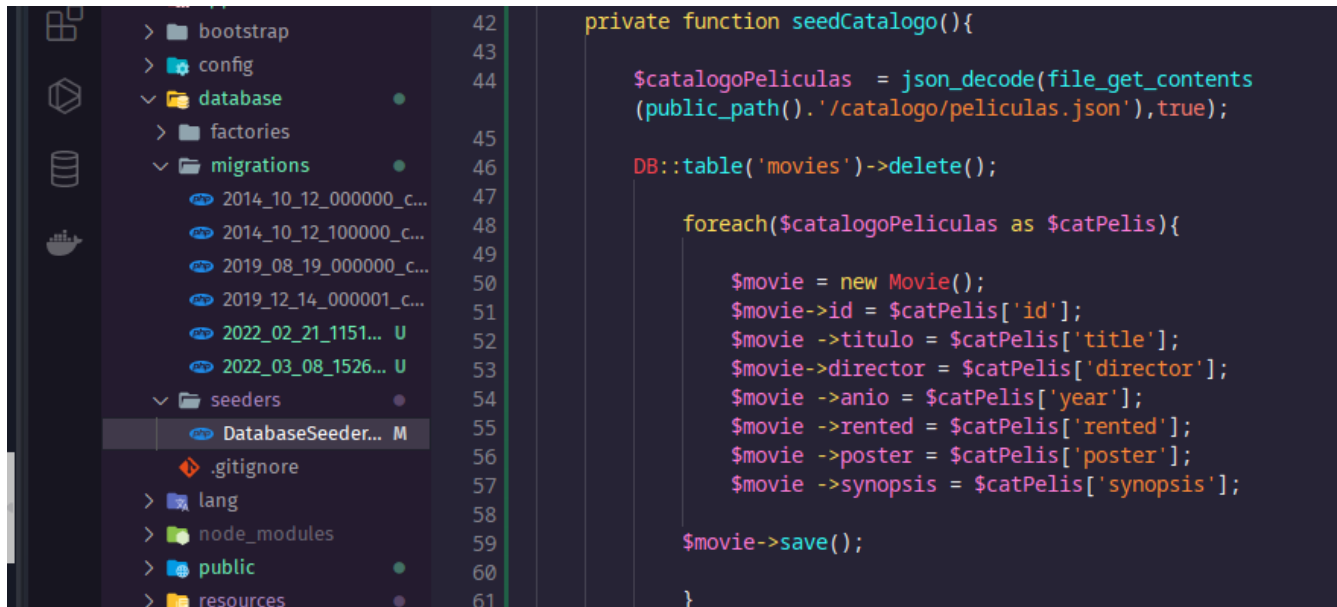
- Table Name: movies
- Engine: InnoDB
- Auto Increment: 22
- Charset: utf8mb4
- Collation: utf8mb4_unicode_ci
- Description: (empty)

Below these settings, a table lists the columns of the 'movies' table:

Column Name	#	Data Type	Not Null	Aut
id	1	bigint(20) unsigned	[v]	
titulo	2	varchar(255)	[v]	
director	3	varchar(255)	[v]	
anio	4	varchar(255)	[v]	
rented	5	tinyint(1)	[v]	
poster	6	varchar(600)	[v]	
synopsis	7	varchar(1200)	[v]	
created_at	8	timestamp	[]	
updated_at	9	timestamp	[]	

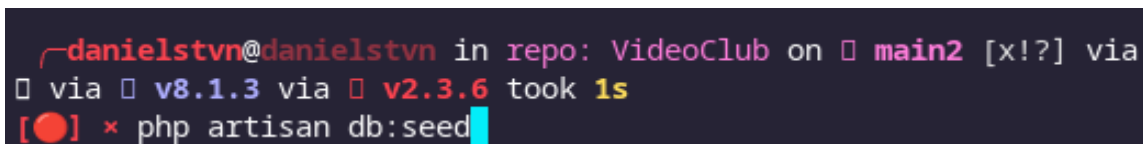
Acontinuacion se generan la semilla para cargar la tabla con registros desde la aplicación.

Se obtienen los datos iniciales desde un archivo .JSON el cual contiene el catalogo de peliculas inicial, se obtiene cada registro de este archivo y utilizando el modelo Movie creado anteriormente se envia a la base de datos esa informacion.



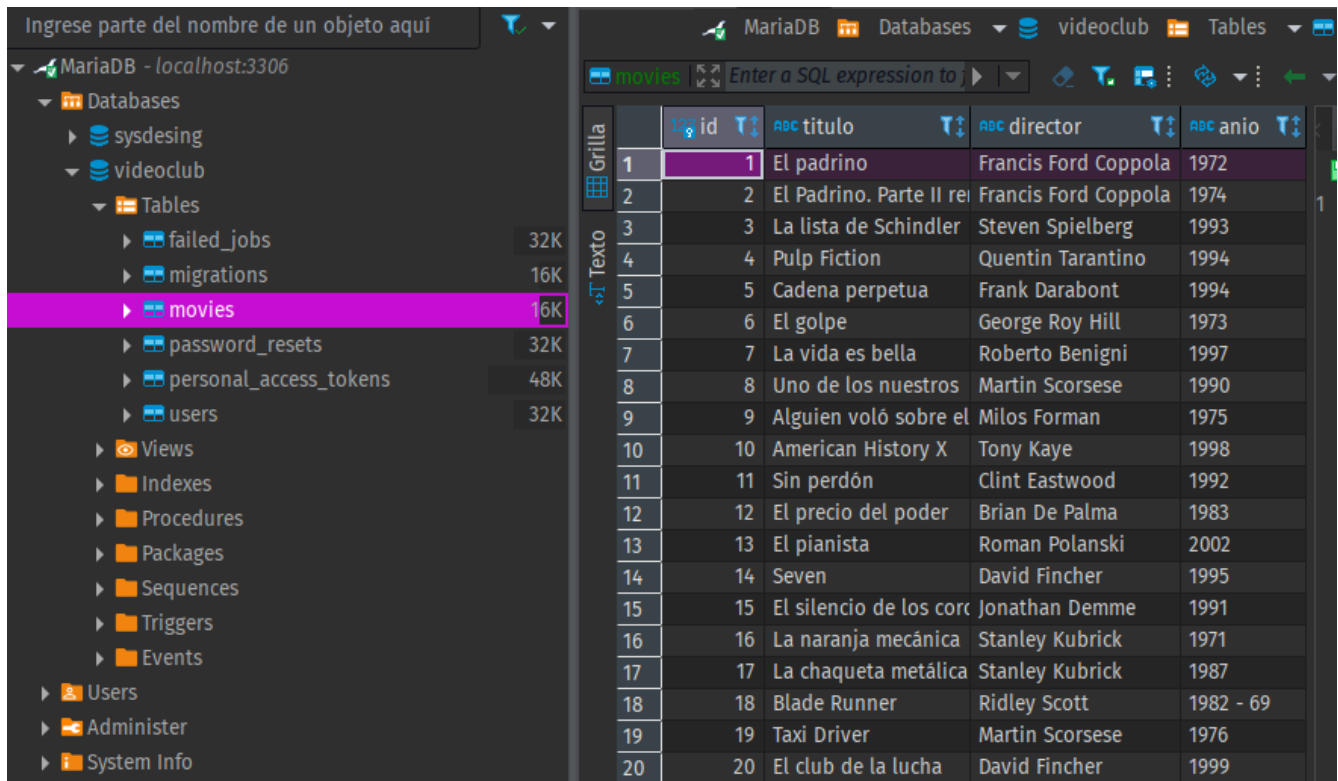
```
private function seedCatalogo(){  
    $catalogoPeliculas = json_decode(file_get_contents(  
        public_path().'/catalogo/peliculas.json'),true);  
  
    DB::table('movies')->delete();  
  
    foreach($catalogoPeliculas as $catPelis){  
        $movie = new Movie();  
        $movie->id = $catPelis['id'];  
        $movie->titulo = $catPelis['title'];  
        $movie->director = $catPelis['director'];  
        $movie->anio = $catPelis['year'];  
        $movie->rented = $catPelis['rented'];  
        $movie->poster = $catPelis['poster'];  
        $movie->synopsis = $catPelis['synopsis'];  
  
        $movie->save();  
    }  
}
```

Con el comando
se envia los datos semilla que se almacenaran en la base de datos



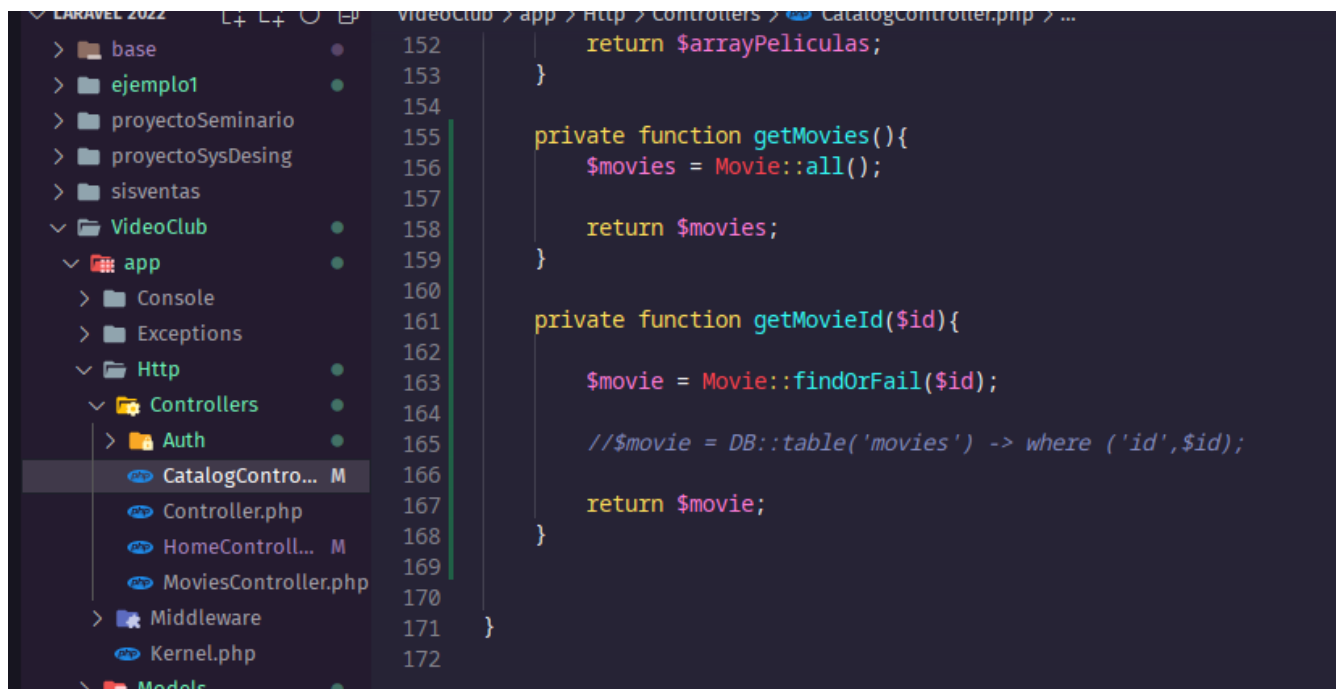
```
└─danielstvn@danielstvn in repo: VideoClub on main2 [x!?] via  
  └─ via v8.1.3 via v2.3.6 took 1s  
  [●] × php artisan db:seed
```

en la base de datos aparecen el catalogo de peliculas que se envia desde la aplicación como semilla



id	titulo	director	anio
1	El padrino	Francis Ford Coppola	1972
2	El Padrino. Parte II re	Francis Ford Coppola	1974
3	La lista de Schindler	Steven Spielberg	1993
4	Pulp Fiction	Quentin Tarantino	1994
5	Cadena perpetua	Frank Darabont	1994
6	El golpe	George Roy Hill	1973
7	La vida es bella	Roberto Benigni	1997
8	Uno de los nuestros	Martin Scorsese	1990
9	Alguien voló sobre el	Milos Forman	1975
10	American History X	Tony Kaye	1998
11	Sin perdón	Clint Eastwood	1992
12	El precio del poder	Brian De Palma	1983
13	El pianista	Roman Polanski	2002
14	Seven	David Fincher	1995
15	El silencio de los corc	Jonathan Demme	1991
16	La naranja mecánica	Stanley Kubrick	1971
17	La chaqueta metálica	Stanley Kubrick	1987
18	Blade Runner	Ridley Scott	1982 - 69
19	Taxi Driver	Martin Scorsese	1976
20	El club de la lucha	David Fincher	1999

en el controlador de catalogo :CatalogController se modifica la obtencion de los datos, en esta ocasion a una base de datos, para esto se crea el siguiente metodo, el cual obtendra todas las peliculas para ser mostradas en el catalogo.



```
return $arrayPeliculas;
}

private function getMovies(){
    $movies = Movie::all();

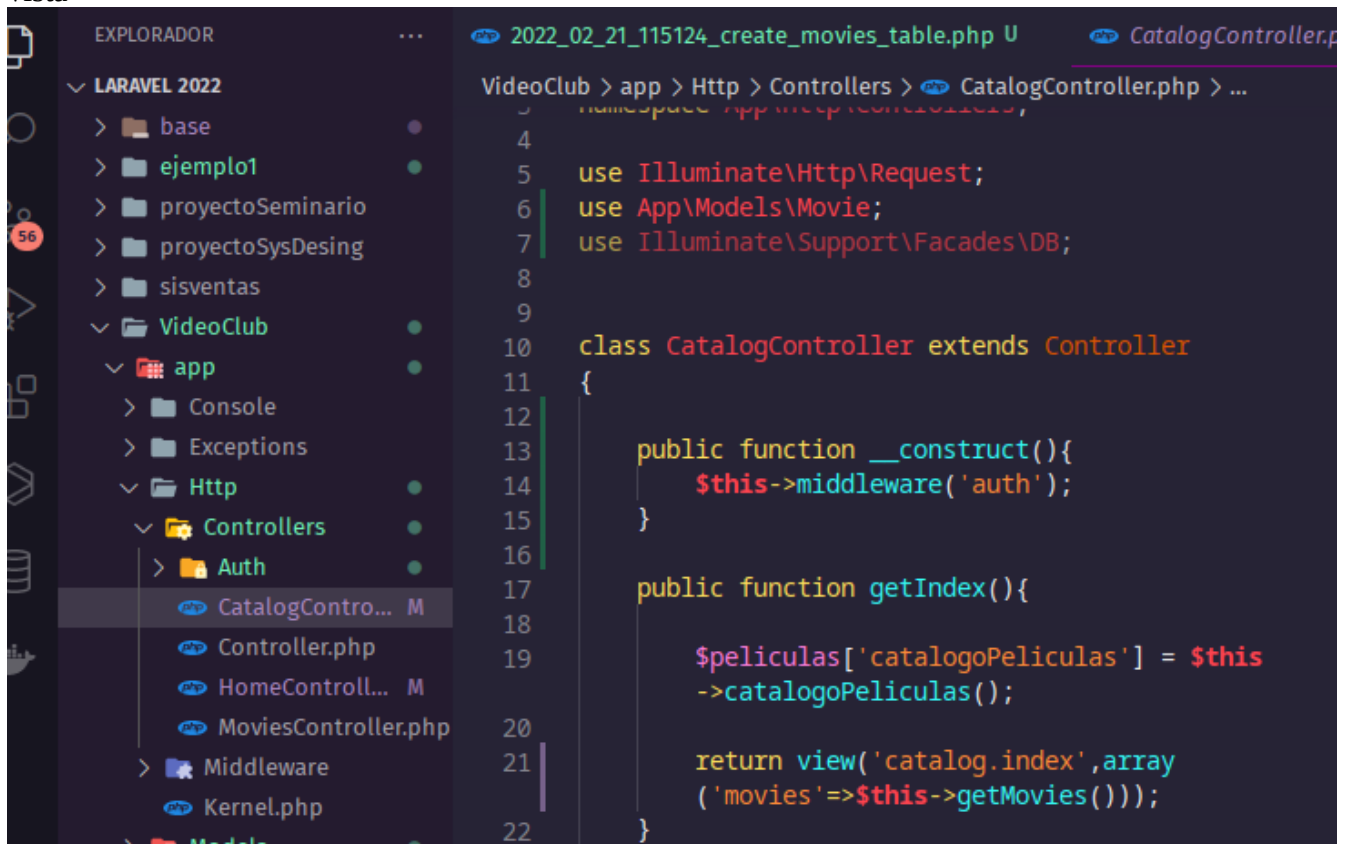
    return $movies;
}

private function getMovieId($id){
    $movie = Movie::findOrFail($id);

    //$movie = DB::table('movies') -> where ('id',$id);

    return $movie;
}
```

En el metodo getIndex se obtiene del metodo getMovies un array el cual se envia como parametro a la vista



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure for 'Laravel 2022', with the 'CatalogController.php' file selected under 'Http > Controllers'. The code editor shows the following PHP code:

```
VideoClub > app > Http > Controllers > CatalogController.php > ...
namespace App\Http\Controllers;

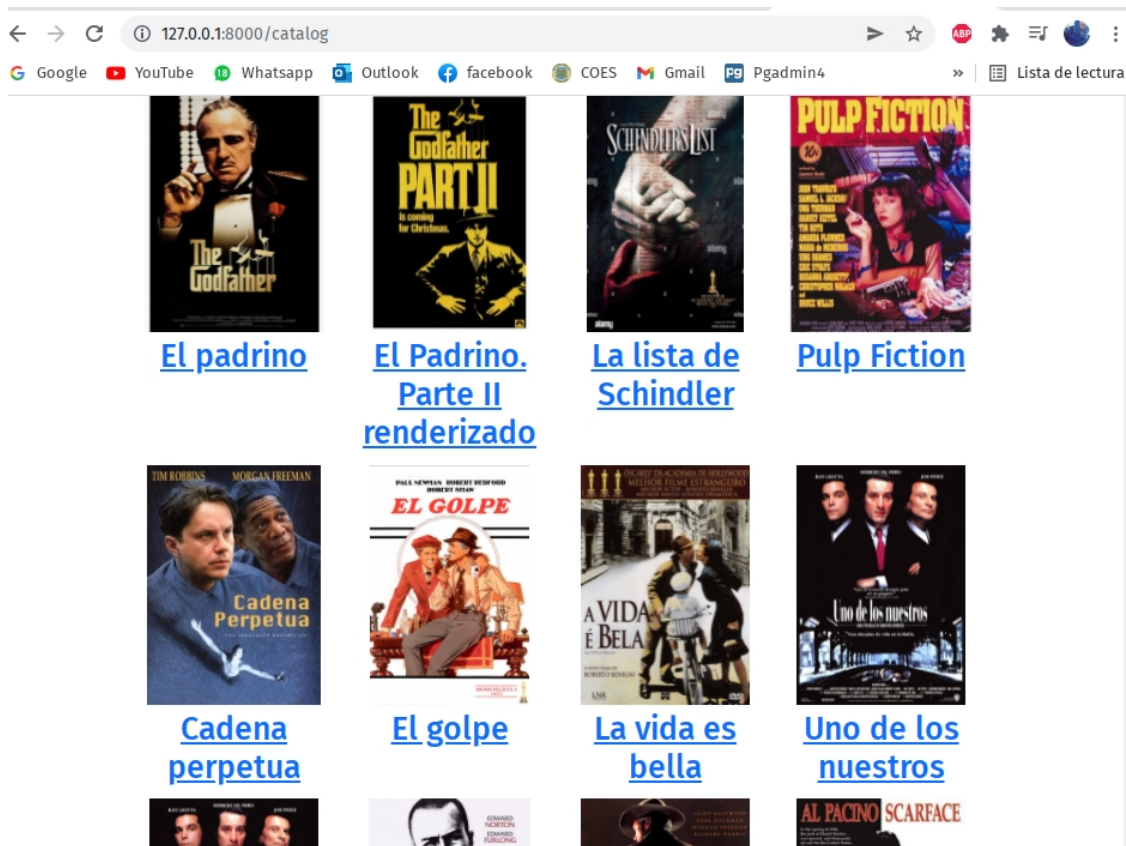
use Illuminate\Http\Request;
use App\Models\Movie;
use Illuminate\Support\Facades\DB;

class CatalogController extends Controller
{
    public function __construct(){
        $this->middleware('auth');
    }

    public function getIndex(){
        $peliculas['catalogoPeliculas'] = $this->catalogoPeliculas();

        return view('catalog.index',array(
            'movies'=>$this->getMovies()));
    }
}
```

el resultado es el siguiente:



Editar película : se crea un metodo para buscar una película mediante su ID

```
VideoClub > app > Http > Controllers > CatalogController.php > ...

private function getMovieId($id){

    $movie = Movie::findOrFail($id);

    //$movie = DB::table('movies') -> where ('id',$id);

    return $movie;

}
```

en el metodo getEdit se envia como parametro un arreglo con la película

```
> base 28
> ejemplo1 29
> proyectoSeminario 30
> proyectoSysDesing 31
> sisventas 32
VideoClub 33

public function getEdit($id){

    return view('catalog.edit',array
    ('movie'=>$this->getMovieId($id))) ;

}
```

como resultado:



El Padrino. Parte II renderizado

Año: 1974

Director: Francis Ford Coppola

Resumen: Continuación de la historia de los Corleone por medio de dos historias paralelas: la elección de Michael Corleone como jefe de los negocios familiares y los orígenes del patriarca, el ya fallecido Don Vito, primero en Sicilia y luego en Estados Unidos, donde, empezando desde abajo, llegó a ser un poderosísimo jefe de la mafia de Nueva York.

Estado:

Devolver película

Editar película

Volver al Listado

Editar película

titulo	<input type="text" value="El Padrino. Parte II renderizado"/>
Año	<input type="text" value="1974"/>
Director	<input type="text" value="Francis Ford Coppola"/>
Poster	<input type="text" value="https://pics.filmaffinity.com/El_padrino_Parte_II-617194725-large.jpg"/>
Resumen	<div>Continuación de la historia de los Corleone por medio de dos historias paralelas: la elección de Michael Corleone como jefe de los negocios familiares y los orígenes del patriarca, el ya fallecido Don Vito, primero en Sicilia y luego en Estados Unidos, donde, empezando desde abajo, llegó a ser un poderosísimo jefe de la mafia de Nueva York.</div>

Guardar Cambios

Volver al Listado

Crear Pelicula: mediante el formulario se registra los datos de la nueva pelicula

Añadir película

Título

Encanto

Año

2021

Director

Byron Howard, Jared Bush

Poster

<https://pics.filmaffinity.com/Encar>

Resumen

Pelicula Encanto prueba 2021

Añadir película

mostrara un mensaje indicando que se creo la nueva peliculas

127.0.0.1:8000 dice

Se creo una nueva Pelicula Coreectamente! :)

Aceptar

en el controlador CatalogController se crea el metodo addMovie el cual obtiene los datos del formulario enviado anteriormente con Request, utiliza el modelo Movie para crear la nueva pelicula, finalmente se redirecciona a la pagina de catalogo para su visualización.

```
39 }
40
41 public function addMovie(Request $request){
42
43     $arrayPeliculas = json_decode(file_get_contents
44         (public_path().'/catalogo/peliculas.json'),true);
45
46     $titulo = request('title');
47     $anio = request('anio');
48     $director = request('director');
49     $poster = $request->get('poster');
50     $synopsis = $request->get('synopsis');
51     $id_newMovie = sizeof($arrayPeliculas) +2;
52
53
54     $mensaje = "Se creo una nueva Pelicula Coreectamente! :)";
55
56     /// comienza con la base de datos
57
58     $movie = new Movie();
59     $movie->id = $id_newMovie;
60     $movie->titulo = $titulo;
61     $movie->director = $director;
62     $movie->anio = $anio;
63     $movie->rented = false;
64     $movie->poster = $poster;
65     $movie->synopsis = $synopsis;
66
67     $movie->save();
68
69     echo "<script> alert('".$mensaje."'); </script>";
70     return view('catalog.index',array
71         ('movies'=>$this->getMovies()));
72 }
```

en la base de datos aparecera registrada la nueva peliculas

19	19	Taxi Driver	Martin Scorsese	1976
20	20	El club de la lucha	David Fincher	1999
21	21	as	as	as
22	22	Encanto	Byron Howard, Jared E	2021

en la pagina de catalogo aparecera la nueva pelicula ingresada



[La chaqueta metálica](#)



[Blade Runner](#)



[Taxi Driver](#)



[El club de la lucha](#)



[as](#)

[Encanto](#)