

Proyecto Pelis - Documentación

Este documento cubre la creación y configuración del proyecto Laravel para una página de películas (Pelis), así como la interacción con la base de datos MySQL.

1. Controlador PelisController

El controlador `PelisController` se encarga de manejar la lógica para obtener y mostrar la lista de películas almacenadas en la base de datos.

Código del controlador:

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Pelis;
```

```
use Illuminate\Http\Request;
```

```
class PelisController extends Controller
```

```
{  
    public function index()  
    {  
        $pelis = Pelis::all(); // Obtener todas las películas  
        return view('index', compact('pelis')); // Pasar los datos a la vista  
    }  
}
```

```
}
```

```
-----
```

2. Rutas (web.php)

La ruta para acceder a las películas está configurada como una ruta de recurso que apunta al controlador `PelisController`.

Código de la ruta:

```
-----
```

```
<?php
```

```
use Illuminate\Support\Facades\Route;
```

```
use App\Http\Controllers\PelisController;
```

```
Route::resource('pelis', PelisController::class);
```

```
-----
```

3. Factory y Seeder

Para poblar la base de datos con películas de prueba, se creó una factory y un seeder. El seeder inserta 10 registros aleatorios en la base de datos utilizando Faker.

Código de la Factory (PelisFactory.php):

```
-----
```

```
<?php
```

```
namespace Database\Factories;
```

```
use App\Models\Pelis;  
  
use Illuminate\Database\Eloquent\Factories\Factory;
```

```
class PelisFactory extends Factory  
{  
    public function definition()  
    {  
        return [  
            'name' => $this->faker->name(),  
            'año' => $this->faker->year()  
        ];  
    }  
}
```

Código del Seeder (PelisSeeder.php):

```
<?php
```

```
namespace Database\Seeders;
```

```
use App\Models\Pelis;
```

```
use Illuminate\Database\Seeder;
```

```
class PelisSeeder extends Seeder
```

```
{  
  
    public function run()  
  
    {  
  
        Pelis::factory(10)->create();  
  
    }  
  
}
```

4. Interacción con la base de datos

El proyecto se conectó a una base de datos MySQL. La base de datos `laravel` fue creada, y se otorgaron los permisos al usuario `ximo` para interactuar con la base de datos.

Comandos usados para la base de datos:

```
mysql> CREATE DATABASE laravel;  
  
mysql> GRANT ALL PRIVILEGES ON laravel.* TO 'ximo'@'localhost' IDENTIFIED BY 'password';  
  
mysql> FLUSH PRIVILEGES;
```

Migraciones ejecutadas:

```
$ php artisan migrate
```

Seeder ejecutado para poblar la base de datos:

```
$ php artisan db:seed --class=PelisSeeder
```

Verificación de los registros con Tinker:

```
$ php artisan tinker
```

App\Models\Pelis::all();
