

Durante esta clase utilizaremos la base de datos **peliculas\_clase\_4** utilizada durante el módulo de MySQL

## Modelado de Datos

1. Utilizando el comando **php artisan make:model NombreDelModelo:**
  - a. Generar un modelo para la tabla Pelicula
  - b. Generar un modelo para la tabla Genero
  - c. Generar un modelo para la tabla Actor
2. Observar los resultados
3. Agregar a Actor el método **getNombreCompleto**.

## Conexión a la base de datos

1. Modificar el archivo **database.php** para que pueda conectarse con nuestra base de datos
2. Modificar el archivo **.env** en el directorio raíz de Laravel con las credenciales de MySQL. Luego reiniciar **php artisan serve** para que tome los cambios.

## Acceso a la base de datos

1. Crearemos una ruta **/actores**
2. Esta ruta redirigirá al método **ActorController@directory**
3. Este controlador mostrará la vista **actores.blade.php**
4. En la vista **actores.blade.php** ingresar un título Actores en una etiqueta **<h1>**
5. Ingresar a **localhost:8000/actores** verificar que se vea el título correctamente
6. Utilizando Eloquent y el modelo creado previamente obtendremos todos los actores dentro de la variable **\$actores** en el controlador ActorController. (Se recomienda utilizar el método **all**, recordar ingresar la línea "use App\Actor" para importar el modelo de actor)
7. Dicha variable será enviada a la vista la cual debe mostrar un listado de todos los actores (mostrando los nombres completos, utilizando Blade)
8. Crearemos una ruta **/actor/{id}**
9. Esta ruta redirigirá al método **ActorController@show**
10. Este controlador mostrará la vista **actor.blade.php**
11. Utilizando Eloquent y el modelo creado previamente obtendremos el actor con el id correspondiente y guardaremos en la variable **\$actor** (se recomienda utilizar el método **find**).
12. Dicha variable será enviada a la vista la cual debe mostrar todos los datos del actor.
13. Modificar la vista **actores.blade.php** para que el nombre de cada actor sea un link a al detalle de los datos de cada actor.

14. Agregar un `<form>` dentro de **actores.blade.php** que nos tenga únicamente un campo de texto. Este formulario tendrá el objetivo de buscar actores.
15. Crear una nueva ruta **/actores/buscar**. El formulario creado en el punto anterior debe dirigir a esta ruta.
16. Esta nueva ruta redirigirá al método **ActorController@search**
17. Utilizando Eloquent y el modelo creado previamente obtendremos actores cuyo nombre contenga los caracteres buscados. Estos se almacenarán en la variable **\$actores**. (se recomienda utilizar el método **where**).
18. Dicha variable será enviada a la vista ya utilizada **actores.blade.php**
19. Se agregará en **actores.blade.php** un segundo formulario con el botón "Limpiar" que envíe a la ruta original sin filtros por búsqueda
  
20. Crearemos una ruta **/actores/agregar**
21. Esta ruta redirigirá al método **ActorController@add**
22. Este controlador mostrará la vista **agregarActor.blade.php**
23. La vista **agregarActor.blade.php** debe tener un formulario con los datos fundamentales de un actor (Nombre y Apellido)
24. Al enviar dicho formulario debe enviar (a través de **POST**) a **/actores/agregar**
25. Crearemos una ruta **/actores/agregar** pero con el método **POST**
26. Esta ruta redirigirá al método **ActorController@store**
27. Este método guardará un actor nuevo y redirigirá a **/actores** (se recomienda utilizar **new** para crear un actor nuevo y el método **save**). Recordar setear la variable "public \$timestamps = false" en el modelo.
  
28. Crearemos una ruta **/actor/{id}/editar**
29. Esta ruta redirigirá al método **ActorController@edit**
30. Este controlador mostrará la vista **editarActor.blade.php**
31. La vista **editarActor.blade.php** debe tener un formulario con los datos de un actor (Nombre y Apellido), muy similar a **agregarActor.blade.php** con los campos completados por defecto con los datos de ese actor{id}
32. Al enviar dicho formulario debe enviar (a través de **POST**) a **/actor/{id}/editar**
33. Crearemos una ruta **/actor/{id}/edit** pero con el método **POST**
34. Esta ruta redirigirá al método **ActorController@update**
35. Este método actualizará el actor seleccionado y redirigirá a **/actores** (se recomienda utilizar los método **find** y **save**). Recordar que el controller podría recibir más de un parámetro, por ejemplo: (request, id).
  
36. Agregaremos en **actor.blade.php** un botón de "Eliminar".
37. Este botón redirigirá a la ruta **/actor/{id}/delete** con el método **POST**
38. Crearemos una ruta **/actor/{id}/delete** que responda al método **POST**
39. Esta ruta redirigirá al método **ActorController@delete**

40. Este controlador borrará el usuario seleccionado (se recomienda utilizar los métodos **find** y **delete**)
41. Luego este controlador redirigirá a **/actores**