Creamos un nuevo proyecto desde cero

*composer create-project laravel/laravel NOMBRE\_PROYECTO*

Una vez que terminó, si todo salió bien, podemos levantar el servidor, para ello:

*cd NOMBRE\_PROYECTO*

*php artisan serve*

Si no se puede entrar por alguna razón, lo más probable es que haya habido un error en la creación del **.env**, por eso debemos hacer del **.env.example** un archivo **.env** para así generar la key necesaria para levantar el servidor. Para esto corremos en consola:

*cp .env.example .env // esto genera un .env a partir del .env.example*

*php artisan key:generate*

Intentamos levantar el servidor de nuevo

*php artisan serve*

Si todo salió bien y en la consola no vemos errores. Entramos a la dirección (en el browser) que nos muestra la consola, podemos ver la vista por defecto de laravel:

*http://127.0.0.1:8000 (o su equivalente http://localhost:8000)*

Hasta el momento todo deberá haber salido perfecto.

Ahora, usando la base de datos de películas actual que hemos venido manejando Configuraremos nuestro archivo **.env** con los datos de la base de datos:

DB\_CONNECTION = mysql

DB\_HOST = 127.0.0.1

DB\_PORT = 3306

DB\_DATABASE = movies\_db

DB\_USERNAME = root

DB\_PASSWORD =

*La contraseña es la misma que usamos para entrar al workbench (generalmente no tiene password o es root)*

¡Recordar que cuando hacemos cambios en el **.env** es necesario bajar y volver a levantar para que todo esté en orden!

Ahora, borraremos nuestra ruta default en el archivo **web.php** y arrancamos de cero con una ruta que use get:

*Route::get('/', 'HomeController@index);*

Para esto, necesitaremos crear el controlador necesario:

*php artisan make:controller HomeController*

Dentro del controlador, generamos el método index, que se usará en la ruta  '/':

*public function index() {*

*return view('index');*

*}*

Teniendo el sistema **ruta-controlador** armado, nos falta la vista! La crearemos a mano con el formato '**nombreDeVista.blade.php**' en la carpeta **resources/views**:

Para ello, antes vamos a crear un template que puedan usar el resto de vistas, tipo **layout.blade.php.**El cual se usará para que todas las vistas extiendan de él. Deberá quedar así:

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title> **@yield('title' )** </title>

<link rel="stylesheet" href="/css/bootstrap.min.css"> Fue a Bootstrap, descargó el archivo y luego lo referenció.

    </head>

<body>

<div class="container">

**@yield('content')**

</div>

    </body>

</html>

Tener en cuenta que en este **layout.blade.php** podemos vincular todo lo que necesitemos (css, bootstrap, javascript, etc).

Como sabemos que vamos a trabajar con una base de datos **movies\_db**, empezamos por armar los modelos correspondientes necesarios!

**Ojo**, siempre tener en cuenta que el modelo por convención se escribe con la primera letra mayúscula y en singular.

*php artisan make:model Actor*

*Esto correspondería a la tabla actors (minúscula y en plural)*

*php artisan make:model Movie*

*Esto correspondería a la tabla movies (minúscula y en plural)*

*php artisan make:model Serie*

*Esto correspondería a la tabla series (minúscula y en plural)*

En la vista **index**, deberán estar los enlaces que permitan ingresar a las secciones:

* Listado de películas
* Listado de Actores
* Listados de Series

Ahora vamos a mostrar  las películas de la base de datos siguiendo el sistema ruta-controlador-vista:

*Route::get('/movies, 'MoviesController@index');*

php artisan make:controller MoviesController

En el controlador deberá existir el método index que traiga todas las películas de la base de datos y las muestre en una lista.

public function index() {

$movies = Movie::all();

*// usamos all() para traer todas las películas de la base de datos*

return view('movies', compact(movies));

*// Retornamos la vista con la variable*

}

En la vista podemos hacer algo así:

*@foreach ($movies as $movie)*

*{{$movie->title}}*

*@endforeach*

Tener en cuenta que el listado de películas debe verse en una lista desordenada.

Después de esto, repetiremos lo mismo para mostrar todos los actores y todas las series.

Adicionalmente crearemos una vista tipo **addMovie.blade.php**, que muestre un formulario que permita agregar una película a la base de datos.

Para ello debemos tener en cuenta que en el modelo **Movie**, necesitaremos poner algo así:

*$protected guarded = []*;

Adicionalmente el método que guarde la película, deberá hacer las validaciones correspondientes, antes de guardar la película en la DB. Tener en cuenta que necesitamos tener dos rutas para hacer esto, una irá por **get** (muestra el formulario) y otra irá por **post**(guarda la película)

*Route::get('/movies/add, 'MoviesController@addForm'');*

*// Ruta get que muestra la vista del formulario*

*Route::post('/movies/save, 'MoviesController@saveMovie'');*

*// Ruta post que guarda la película*

**CHALLENGES:**

Hacer relaciones para los modelos **Movie** y **Serie** que muestren los actores que actuaron en cada serie/película.

Adicionalmente hacer una vista que muestre el detalle de cada película/serie, tipo:

* Titulo
* Rating
* Género
* Fecha de lanzamiento
* Actores