

VIDEOCLUB CAPÍTULO 2

Incluir el CDN de Tailwind en el fichero `app.blade.php`.

Ejercicio 1 - Controladores

Vamos a crear los controladores necesarios para gestionar nuestra aplicación y además actualizaremos el fichero de rutas para que los utilice.

Empezamos por añadir los dos controladores que nos van a hacer falta: *CatalogController.php* y *HomeController.php*.

A continuación, vamos a añadir los métodos a esos controladores. En la siguiente tabla se muestran los métodos por controlador y las rutas asociadas:

Ruta	Controlador	Método
/	HomeController	getHome
catalog	CatalogController	getIndex
catalog/show/{id}	CatalogController	getShow
catalog/create	CatalogController	getCreate
catalog/edit/{id}	CatalogController	getEdit

En el fichero *web.php* hay que modificar las rutas para que apunten a los métodos de los controladores, los cuales deben devolver las vistas correspondientes.

Tened en cuenta que los métodos *getShow* y *getEdit* tienen que recibir como parámetro el *\$id* del elemento a mostrar o editar, por lo que la definición del método en el controlador tendrá que ser como sigue:

```
public function getShow($id){  
    return view('catalog.show', array('id'=>$id));  
}
```

Ejercicio 2 – Completar las vistas

En este ejercicio vamos a terminar los métodos de los controladores que hemos creado en el ejercicio anterior y además completaremos las vistas asociadas.

Método HomeController@getHome

Este método mostrará la vista `welcome.blade.php`

Método CatalogController@getIndex

Este método tiene que mostrar un listado de todas las películas que tiene el videoclub. El listado de películas se puede obtener del fichero *array_películas.php* de la Moodle. Este array hay que copiarlo como una variable miembro de la clase (más adelante se sustituirá por una BD). En el método del controlador simplemente tendremos que modificar la generación de la vista para pasarle este array de películas completo (*\$this->arrayPelículas*)

En la vista correspondiente sólo hay que recorrer el array:

```
<div class="py-12">
  <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
    <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
      <div class="p-6 text-gray-900">
        <div class="grid grid-cols-4">
          @foreach ($arrayPelículas as $key=>$película)
            <a href="{{url('catalog/show/'.$key)}}">
              
              <h4 style="min-height:45px;margin:5px 0 10px 0">
                {{ $película['title'] }}
              </h4></a>
            @endforeach
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Se ha añadido a cada película un enlace para que al pulsar sobre ella nos lleve a otra página donde se muestra el detalle de esa película; la ruta se genera con la posición de la película dentro del array.

Método CatalogController@show

Este método se utiliza para mostrar la vista de detalle de una película. Hay que tener en cuenta que el método correspondiente recibe un identificador que (de momento) se refiere a la posición de la película en el array. Por lo tanto, tendremos que coger dicha película del array (*\$this->arrayPelículas[\$i]*) y pasársela a la vista.

En la vista vamos a crear dos columnas, la primera para mostrar la imagen de la película y la segunda para mostrar los detalles de esta:

```

<div class="py-12">
  <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
    <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
      <div class="p-6 text-gray-900">
        <div class="grid grid-rows-1">
          <div class="grid grid-cols-2">
            <div>
              IMAGEN DE LA PELÍCULA
            </div>
            <div>
              DATOS DE LA PELÍCULA
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Para mostrar el estado de la película consultaremos el valor de *render* del array, que podrá tener dos posibles valores:

- En caso de estar disponible (false) aparecerá el estado “Película disponible” y un botón azul para “Alquilar película”.
- En el caso de estar alquilada (true) aparecerá el estado “Película actualmente alquilada” y un botón rojo para “Devolver película”.

Además, habrá que incluir dos botones más, un botón que nos llevará a editar la película y otro para volver al listado de películas.

NOTA: los botones de alquilar y devolver de momento no van a tener funcionalidad.

Usar flowbite.com para el diseño de los botones.



La vida es bella

Año: 1997 Director: Roberto Benigni

Resumen: En 1939, a punto de estallar la Segunda Guerra Mundial (1939-1945), el extravagante Guido llega a Arezzo (Toscana) con la intención de abrir una librería. Allí conoce a Dora y, a pesar de que es la prometida del fascista Ferruccio, se casa con ella y tiene un hijo. Al estallar la guerra, los tres son internados en un campo de exterminio, donde Guido hará lo imposible para hacer creer a su hijo que la terrible situación que están padeciendo es tan sólo un juego.

Estado: Película actualmente alquilada

Devolver película

Editar película

Volver al listado

Método CatalogController@getCreate

Este método devuelve la vista *“catalog.create”* para añadir una nueva película. Para crear este formulario en la vista correspondiente nos podemos basar en el contenido de la vista de registro de usuario.

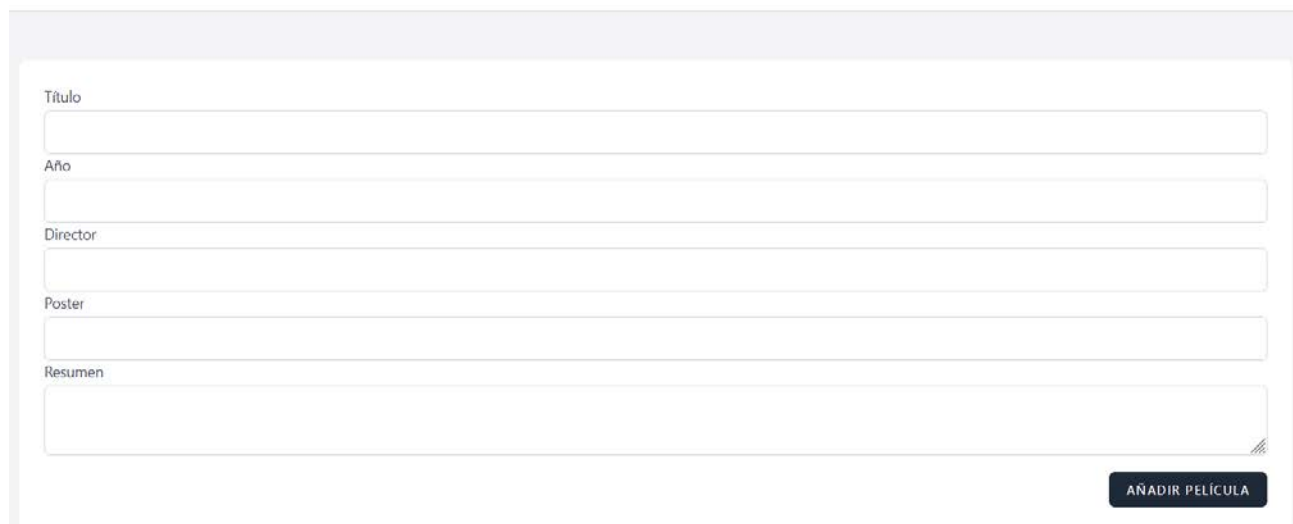
Los campos que debe tener el formulario son:

Label	Name	Tipo de campo
Título	title	text
Año	year	text
Director	director	text
Poster	poster	text
Resumen	synopsis	textarea

Además, tendrá un botón al final con el texto *“Añadir película”*.

NOTA: De momento el formulario no funcionará.

VIDEOCLUB



Método CatalogController@getEdit

Este método permitirá modificar el contenido de una película. El formulario será exactamente igual que el de añadir película, así que se puede copiar en esta vista y cambiar los siguientes puntos:

- El título por *“Modificar Película”*
- El texto del botón de envío por *“Modificar película”*
- Añadir justo debajo de la apertura del formulario la directiva de Blade para indicar que el formulario se va a enviar por PUT: **@method('PUT')**

De momento no habrá que hacer nada más.