

M07 - GUIA PROJECTE PRÀCTICA03

Exercici 1 - Controladors (1 punt)

En aquest primer exercici crearem els controladors necessaris per gestionar la nostra aplicació i a més actualitzarem el fitxer de rutes perquè els utilitzi.

Comencem per afegir els dos controladors que ens faran falta: `CatalogController.php` y `HomeController.php`. Per a fer-ho, heu d'utilitzar la comanda de Artisan que permet crear un controlador "buit", és a dir, sense mètodes (veure material de suport).

Link que he trobat per la realització de la comanda:

<https://quickadminpanel.com/blog/list-of-21-artisan-make-commands-with-parameters/>

```
adria@adria-HP-ProDesk-600-G1-SFF:~/Escritorio/Servidor/miweb/proyecto_rellotges$ php artisan make:controller catalog/CatalogController.php
Controller created successfully.
adria@adria-HP-ProDesk-600-G1-SFF:~/Escritorio/Servidor/miweb/proyecto_rellotges$ php artisan make:controller HomeController.php
Controller created successfully.
```

A continuació anem a afegir els mètodes d'aquests controladors. A la següent taula resum podem veure un llistat dels mètodes per controlador i les rutes que tindran associades:

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

Recordeu que els mètodes `getShow` i `getEdit` hauran de rebre com a paràmetre el `$id` de l'elementa mostrar o d'editar, de manera que la definició de mètode al controlador haurà de ser com la següent:

```
web.php x CatalogController.php.php x HomeController.php.php x
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController.php extends Controller
{
    // Pantalla principal
    public function getHome (){
        return view('home');
    }
}
```

```
web.php x CatalogController.php.php x
<?php

namespace App\Http\Controllers\catalog;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class CatalogController.php extends Controller
{
    // Llistat rel·lotges
    public function getIndex (){
        return view('catalog.index');
    }

    // Vista detall rel·lotge
    public function getShow ($id){
        return view('catalog.show', array('id'=>$id));
    }

    // Afegir rel·lotge
    public function getCreate (){
        return view('catalog.create');
    }

    // Modificar rel·lotge
    public function getEdit ($id){
        return view('catalog.edit', array('id'=>$id));
    }
}
```

Finalment canviarem el fitxer de rutes `routes/web.php` perquè totes les rutes que teníem definides (excepte les de login i logout, que les deixarem tal com estan) apuntin als nous mètodes dels controladors, per exemple:

```
Route::get('/', 'HomeController @ getHome');
```

```
web.php x

// Pantalla principal
Route::get('/', 'HomeController @ getHome');

// Login usuari
Route::get('/login', function () {
    return view('auth.login');
});

// Logout usuari
Route::get('/logout', function () {
    return view('home');
});

// Llistat rellotges
Route::get('/catalog', 'CatalogController @ getIndex');

// Vista detall rellotge
Route::get('/catalog/show/{id}', 'CatalogController @ getShow');

// Afegir rellotge
Route::get('/catalog/create', 'CatalogController @ getCreate');

// Modificar rellotge
Route::get('/catalog/edit/{id}', 'CatalogController @ getEdit');
```

El codi "hardcoded" que havíem afegit a cada ruta amb el "return" (al fitxer de rutes, `web.php`) l'heude moure a mètode del controlador corresponent.

Exercici 2 - Completar les vistes (2 punts)

En aquest exercici acabarem els mètodes dels controladors que hem creat en l'exercici anterior i a més completarem les vistes associades.

→ Mètode HomeController@getHome

En aquest mètode de moment només farem una redirecció a l'acció que mostra el llistat de pel·lícules del catàleg:

```
return redirect()->action('CatalogController@getIndex');
```

Més endavant haurem de comprovar si l'usuari està loguejat o no (protegir rutes) i, en cas que no ho estigui, redirigir-lo al formulari de login.

```
class HomeController.php extends Controller
{
    // Pantalla principal
    public function getHome (){
        //return view('home');
        return redirect()->action('CatalogController @ getIndex');
    }
}
```

→ Mètode CatalogController@getIndex

Aquest mètode ha de mostrar un llistat de totes les pel·lícules que té el videoclub. El llistat de pel·lícules el podeu obtenir del fitxer `array_películas.php` facilitat amb els materials. Aquest array de pel·lícules ho heu de copiar com a variable membre de la classe (més endavant lesemmagatzemarem a la base de dades). En el mètode del controlador simplement haurem de modificar la generació de la vista per a passar-li aquest array de pel·lícules complet (`$this->arrayPelículas`).

1 - Canviem les rutes al document `web.php`, doncs sinó no funcionaràn de manera correcta.

```
// Llistat rel·lotges
Route::get('/catalog', 'catalog\CatalogController@getIndex');

// Vista detall rel·lotge
Route::get('/catalog/show/{id}', 'catalog\CatalogController@show');

// Afegir rel·lotge
Route::get('/catalog/create', 'catalog\CatalogController@create');

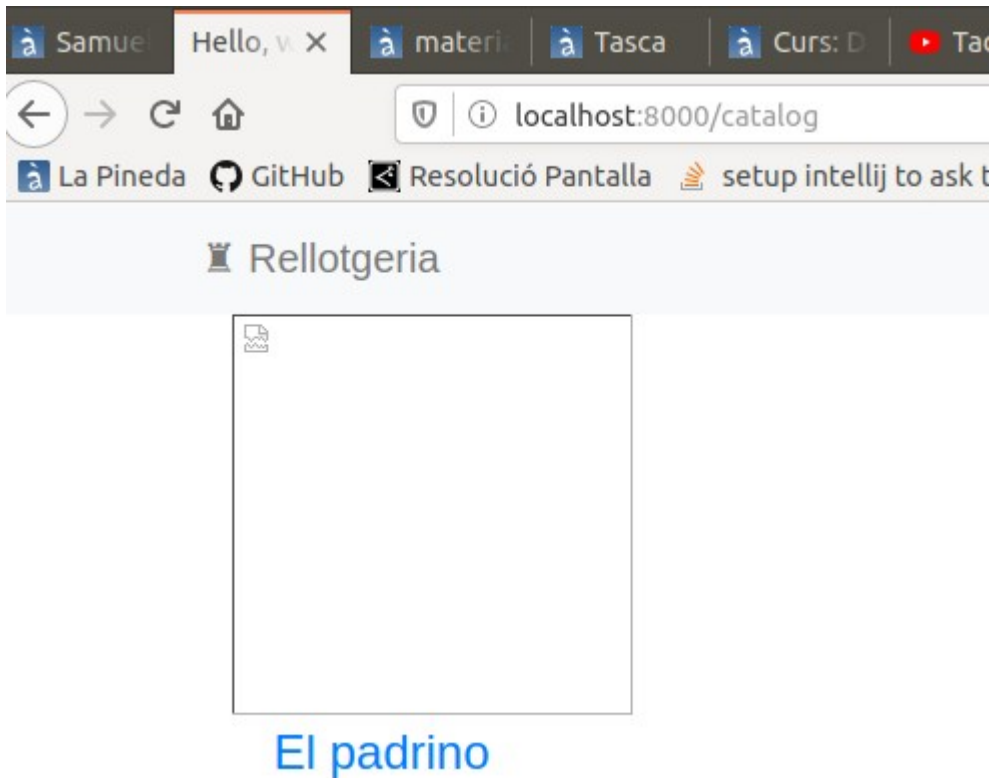
// Modificar rel·lotge
Route::get('/catalog/edit/{id}', 'catalog\CatalogController@edit');
```

2 - Enganxem el array al controlador i el passem a la view:

```
private $array_rellotges = array(
    array(
        'title' => 'El padrino',
        'year' => '1972',
        'director' => 'Francis Ford Coppola',
        'poster' => 'http://ia.media-imdb.com/images/M/
            MV5BMjEyMjc5NDI4MF5BMl5BanBnXkFtZTcwMDA5Mzg3OA@@._V1_SX214_AL_.jpg',
        'rented' => false,
        'synopsis' => 'Don Vito Corleone (Marlon Brando) es el respetado y
            temido jefe de una de las cinco familias de la mafia de Nueva York.
            Tiene cuatro hijos: Connie (Talia Shire), el impulsivo Sonny (James
            Caan), el pusilánime Freddie (John Cazale) y Michael (Al Pacino),
            que no quiere saber nada de los negocios de su padre. Cuando
            Corleone, en contra de los consejos de \'Il consigliere\' Tom Hagen (
            Robert Duvall), se niega a intervenir en el negocio de las drogas,
            el jefe de otra banda ordena su asesinato. Empieza entonces una
            violenta y cruenta guerra entre las familias mafiosas.'
    )
);

// Llistat rel·lotges
public function getIndex (){

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



Mètode CatalogController@show

Aquest mètode s'utilitza per mostrar la vista detall d'una pel·lícula. Hem de tenir en compte que el mètode corresponent rep un identificador que (de moment) es refereix a la posició de la pel·lícula en l'array. Per tant, haurem d'agafar aquesta pel·lícula de l'array (`$this->arrayPelículas[$id]`) i passar-se-la a la vista.

En aquesta vista crearem dues columnes, la primera columna per mostrar la imatge de la pel·lícula i la segona per a incloure tots els detalls. A continuació s'inclou l'estructura HTML que hauria de tenir aquesta pantalla:

```
// Vista detall rellotge
public function getShow ($id){
    return view('catalog.show', array('id'=>$this->array_rellotges[$id]));
}
```



```

CatalogController.php x show.blade.php x index.blade.php x
@extends('layouts.master')
@section('content')

<div class="row">
  <div class="col-sm-4">
    
  </div>
  <div class="col-sm-8">
    <h4 style="min-height:45px;margin:5px 0 10px 0;font-weight: bold;">
      {{ $id['title'] }} ~ {{ $id['year'] }}
    </h4>
    <p><b>Color:</b> {{ $id['color'] }}</p>
    <p><b>Descripció:</b></p>
    <p style="margin-left:20px">{{ $id['synopsis'] }}</p>
    <p><b>Estat:</b>

      {{-- {{isset ($id['rented']) ? 'Sí':'No'}} --}}
      @if ($id['rented'])

        Rellotge disponible!</p><br/><p>
        <button type="button" class="btn btn-primary">Llogar rellotge</button>
      >
      @else

        Rellotge actualment llogat!</p><br/><p>
        <button type="button" class="btn btn-danger">Retornar rellotge</button>
      </p>

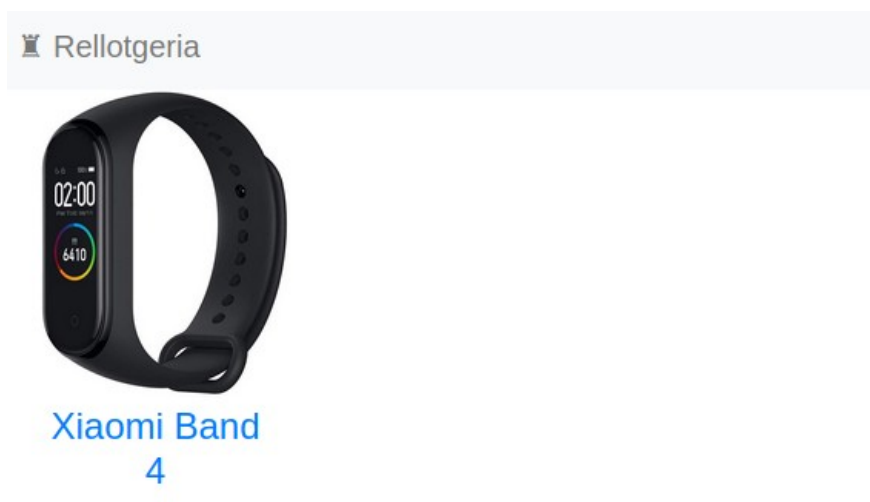
    @endif

    <button type="button" class="btn btn-warning">Editar Rellotge</button>
    <button type="button" class="btn btn-light">Tornar al llistat</button>
  </div>
</div>

@stop

```

Resultat:





Xiaomi Band 4 ~ 2019

Color: Negro

Descripció:

Xiaomi Band 4 Pulsera de Fitness Inteligente Monitor de Ritmo cardíaco 135 mAh Pantalla Color Bluetooth 5.0 más Reciente 2019 (Negro).

Estat: Relotge actualment llogat!

Retornar rellotge

Editar Relotge

Tornar al llistat

Mètode CatalogController@getCreate

Aquest mètode retorna la vista `catalog.create` per a afegir una nova pel·lícula. Per a crear aquest formulari a la vista corresponent ens podem basar en el contingut de la plantilla `catalog_create.php`. Aquesta plantilla té una sèrie de "TODO's" que cal completar. En total haurà de tenir els següents camps:

Label	Name	Tipo de campo
Títol	title	text
Any	year	text
Director	director	text
Pòster	poster	text
Resum	synopsis	textarea

A més tindrà un botó al final que mostrarà text "Afegir pel·lícula".

```
<div class="card-header text-center">
  Modificar Rellotge
</div>
<div class="card-body" style="padding:30px">

  <form action="#" method="POST">

    {{ csrf_field() }}

    <div class="form-group">
      <label for="title">Modificar Rellotge</label>
      <input type="text" name="title" id="title" class="form-control">
    </div>

    <div class="form-group">
      <label for="year">Any</label>
      <input type="text" name="title" id="title" class="form-control">
    </div>

    <div class="form-group">
      <label for="color">Color</label>
      <input type="text" name="title" id="title" class="form-control">
    </div>

    <div class="form-group">
      <label for="imagen">Link Imatge</label>
      <input type="text" name="title" id="title" class="form-control">
    </div>

    <div class="form-group">
      <label for="synopsis">Resum</label>
      <textarea name="synopsis" id="synopsis" class="form-control" rows=
        "3"></textarea>
    </div>

    <div class="form-group text-center">
      <button type="submit" class="btn btn-primary" style="padding:8px
        80px;margin-top:25px;">
        Modificar Rellotge!
      </button>
    </div>
  </form>
</div>
```

Afegir Rellotge

Nom

Any

Color

Link Imatge

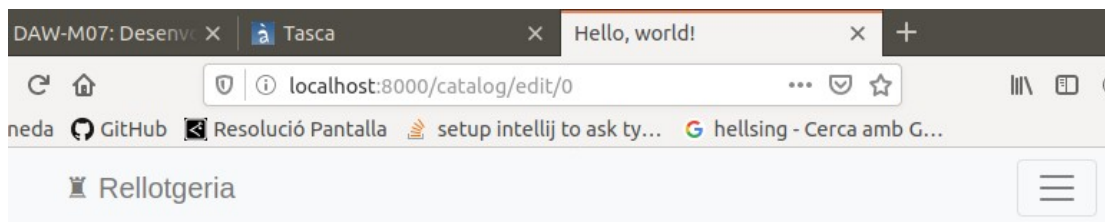
Resum

Afegir Rellotge!

Mètode CatalogController@getEdit

Aquest mètode permetrà modificar el contingut d'una pel·lícula. El formulari serà exactament igual al d'afegir pel·lícula, així que podem fer copy/paste en aquesta vista i simplement canviar els següents punts:

- El títol per "Modificar pel·lícula".
- El text del botó d'enviament per "Modificar pel·lícula".
- Afegir tot just sota de l'obertura del formulari el camp ocult que indica que el formulari s'enviarà per PUT. Recordeu que Laravel inclou el mètode `method_field('PUT')` que ens ajudarà a fer això.



Modificar Relotge

Modificar Relotge

Any

Color

Link Imatge

Resum

Modificar Relotge!

```

<div class="card-header text-center">
  Modificar Rellotge
</div>
<div class="card-body" style="padding:30px">

  <form action="#" method="POST">

    {{ method_field('PUT') }}
    {{ csrf_field() }}

    <div class="form-group">
      <label for="title">Modificar Rellotge</label>
      <input type="text" name="title" id="title" class="form-control">
    </div>

    <div class="form-group">
      <label for="year">Any</label>
      <input type="text" name="title" id="title" class="form-control">
    </div>

    <div class="form-group">
      <label for="color">Color</label>
      <input type="text" name="title" id="title" class="form-control">
    </div>

    <div class="form-group">
      <label for="imagen">Link Imatge</label>
      <input type="text" name="title" id="title" class="form-control">
    </div>

    <div class="form-group">
      <label for="synopsis">Resum</label>
      <textarea name="synopsis" id="synopsis" class="form-control" rows=
        "3"></textarea>
    </div>

    <div class="form-group text-center">
      <button type="submit" class="btn btn-primary" style="padding:8px
        80px;margin-top:25px;">
        Modificar Rellotge!
      </button>
    </div>
  </form>
</div>

```