

Exercici 1 - Migració de la taula usuaris (0.5punts)

En primer lloc anem a crear la taula de la base de dades per emmagatzemar els usuaris que tindran accés a la plataforma de gestió del vostre projecte.

Com hem vist a la teoria, Laravel ja inclou una migració amb el nom `create_users_table` per a la taula `users` amb tots els camps necessaris. Obrirem aquesta migració i comprovarem que els camps inclosos coincideixen amb els de la següent taula:

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->string('name');
        $table->string('email')->unique();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

Comprova també que al mètode `down` de la migració es desfacin els canvis que es fan en el mètode `up` (en aquest cas seria eliminar la taula).

```
public function down()
{
    Schema::dropIfExists('users');
}
```

Finalment farem servir la comanda d'Artisan que afegeix les noves migracions i comprovarem amb PHPMyAdmin que la taula s'ha creat correctament amb tots camps indicats.

```
adria@adria-HP-ProDesk-600-G1-SFF:~/Escritorio/Servidor/miweb/projecte_rellotges
$ php artisan migrate:refresh
Rolling back: 2020_02_17_072829_create_rellotges_table
Rolled back: 2020_02_17_072829_create_rellotges_table (0.08 seconds)
Rolling back: 2019_08_19_000000_create_failed_jobs_table
Rolled back: 2019_08_19_000000_create_failed_jobs_table (0.05 seconds)
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_100000_create_password_resets_table (0.05 seconds)
Rolling back: 2014_10_12_000000_create_users_table
Rolled back: 2014_10_12_000000_create_users_table (0.06 seconds)
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (0.15 seconds)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (0.12 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (0.07 seconds)
Migrating: 2020_02_17_072829_create_rellotges_table
Migrated: 2020_02_17_072829_create_rellotges_table (0.07 seconds)
```

Exercici 2 - Seeder d'usuaris (0.5 punts)

Ara procedirem a omplir la taula users amb dades per defecte per a poder provar l'aplicació web. Editarem el "fitxer de llavors" situat a database/seeds/DatabaseSeeder.php i seguirem els següents passos:

```
private $array_usuaris = array(
    array(
        'name' => 'Adrià',
        'email' => 'adria@gmail.com',
        'password' => 'adria'
    ),
    array(
        'name' => 'Javier',
        'email' => 'javi@gmail.com',
        'password' => 'javi'
    ),
    array(
        'name' => 'Cristian',
        'email' => 'cristian@gmail.com',
        'password' => 'cristian'
    )
);
```

- Creem un mètode privat (dins de la mateixa classe) anomenat seedUsers() que s'haurà de cridar des del mètode run de la següent forma:

```
private function seedUsers(){
    DB::table('users')->delete();
    foreach ($this->array_usuaris as $usuari) {
        # code...
        $u=new users;
        $u->name=$usuari['name'];
        $u->email=$usuari['email'];
        $u->password=$usuari['password'];
        $u->save();
    }
}
```

```
public function run()
{
    self :: seedCatalog();
    $this->command->info('Taula catàleg inicialitzem amb dades!');

    self :: seedUsers();
    $this->command->info('Taula usuaris inicialitzada amb dades!');
}
```

- Dins del nou mètode seedUsers() realitzem les següents accions:
 - En primer lloc esborrem el contingut de la taula users.
 - A continuació creem un parell d'usuaris de prova. Recorda que per a guardar el password cal encriptar-lo manualment usant el mètode bcrypt (revisa la secció "Registre d'un usuari" del material de suport).

```
private function seedUsers(){
    DB::table('users')->delete();
    foreach ($this->array_usuaris as $usuari) {
        # code...
        $u=new User;
        $u->name=$usuari['name'];
        $u->email=$usuari['email'];
        $u->password=Hash::make($usuari['password']);
        $u->save();
    }
}
```

```
<?php
use Illuminate\Database\Seeder;
use App\rellotges;
use App\User;
```










Finalment haurem d'executar la comanda d'Artisan que processa les llavors. Un cop fet això comprovarem a PHPMyAdmin que s'han afegit els usuaris a la taula users .

```
adria@adria-HP-ProDesk-600-G1-SFF:~/Escritorio/Servidor/miweb/proyecto_rellotges$ php artisan db:seed
Taula catàleg inicialitzem amb dades!
Taula usuaris inicialitzada amb dades!
Database seeding completed successfully.
```

+ Opciones

↶

↷

				id	name	email	password	remember_token	created_at	updated_at
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	Adrià	adria@gmail.com	\$2y\$10\$5itaXdmDwMOP7xttdwt1C.znFsZenOlv7DCUIZDjKnr...	NULL	2020-02-26 08:29:03	2020-02-26 08:29:03
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	Javier	javi@gmail.com	\$2y\$10\$z0rMujmVpCi8T5ZfsBwSOVmGKn0Zx8D1j2JIRkoV5Y...	NULL	2020-02-26 08:29:03	2020-02-26 08:29:03
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	Cristian	cristian@gmail.com	\$2y\$10\$28gAWQ4EpFQ.VpnMI3Kfk.tta8.bVl9poTxpotjYnn...	NULL	2020-02-26 08:29:03	2020-02-26 08:29:03

Exercici 3 - Sistema d'autenticació (1 punt)

En aquest exercici completarem el sistema d'autenticació. En primer lloc executa la comanda de Artisan `php artisan make:auth` per a generar totes les rutes i vistes necessàries per al control d'usuaris.

```
adria@adria-HP-ProDesk-600-G1-SFF:~/Escritorio/Servidor/miweb/proyecto_r
ellotges$ composer require laravel/ui
Using version ^1.2 for laravel/ui
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing laravel/ui (v1.2.0): Downloading (100%)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: laravel/ui
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
```

```
adria@adria-HP-ProDesk-600-G1-SFF:~/Escritorio/Servidor/miweb/proyecto_r
ellotges$ php artisan ui vue --auth
Vue scaffolding installed successfully.
Please run "npm install && npm run dev" to compile your fresh scaffoldin
g.

The [auth/login.blade.php] view already exists. Do you want to replace
it? (yes/no) [no]:
> yes

The [home.blade.php] view already exists. Do you want to replace it? (y
es/no) [no]:
> yes

Authentication scaffolding generated successfully.
```

```
adria@adria-HP-ProDesk-600-G1-SFF:~/Escritorio/Servidor/miweb/proyecto_r
ellotges$ sudo apt install npm
[sudo] contraseña para adria:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
```


- Afegeix un middleware de tipus grup que apliqui el filtre auth per protegir totes les rutes del catàleg (excepte l'arrel "/" i les d'autenticació).

```
Auth::routes();
Route::get('/home', 'HomeController@index')->name('home');
// Pantalla principal
Route::get('/', 'HomeController@getHome|_');
```

```
Route::group(['middleware' => 'auth'], function(){

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

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

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

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

Laravel

Adrià ▼

Dashboard

You are logged in!

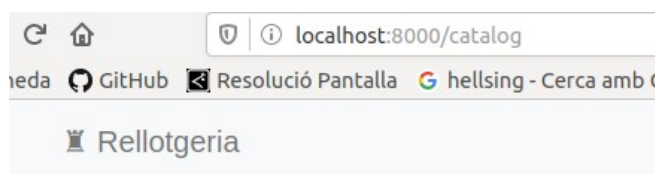
- Revisa mitjançant la comanda d'Artisan "php artisan route:list" les noves rutes i que el filtre auth s'apliqui correctament.

```
adria@adria-HP-ProDesk-600-G1-SFF:~/Escritorio/Servidor/miweb/proyecto_rellotges$ php artisan route:list
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		App\Http\Controllers\HomeController@getHome	web,auth
	GET HEAD	api/user		Closure	api,auth:api
	GET HEAD	catalog		App\Http\Controllers\catalog\CatalogController@getIndex	web,auth
	GET HEAD	catalog/create		App\Http\Controllers\catalog\CatalogController@getCreate	web,auth
	GET HEAD	catalog/edit/{id}		App\Http\Controllers\catalog\CatalogController@getEdit	web,auth
	GET HEAD	catalog/show/{id}		App\Http\Controllers\catalog\CatalogController@getShow	web,auth
	GET HEAD	home	home	App\Http\Controllers\HomeController@getHome	web,auth
	GET HEAD	login	login	App\Http\Controllers\Auth\LoginController@showLoginForm	web,guest
	POST	login	login	App\Http\Controllers\Auth\LoginController@login	web,guest
	POST	logout	logout	App\Http\Controllers\Auth\LoginController@logout	web
	GET HEAD	password/confirm	password.confirm	App\Http\Controllers\Auth\ConfirmPasswordController@showConfirmForm	web,auth
	POST	password/confirm		App\Http\Controllers\Auth\ConfirmPasswordController@confirm	web,auth
	POST	password/email	password.email	App\Http\Controllers\Auth\ForgotPasswordController@sendResetLinkEmail	web
	GET HEAD	password/reset	password.request	App\Http\Controllers\Auth\ForgotPasswordController@showLinkRequestForm	web
	POST	password/reset	password.update	App\Http\Controllers\Auth\ResetPasswordController@reset	web
	GET HEAD	password/reset/{token}	password.reset	App\Http\Controllers\Auth\ResetPasswordController@showResetForm	web
	POST	register	register	App\Http\Controllers\Auth\RegisterController@register	web,guest
	GET HEAD	register	register	App\Http\Controllers\Auth\RegisterController@showRegistrationForm	web,guest

Modifica el controlador LoginController perquè quan es realitzi el login et redirigeixi a la ruta/catalog. Hauràs de modificar la seva propietat redirectTo per a afegir la ruta de redirecció(revisa l'apartat "Autenticació d'un usuari" del material de suport).

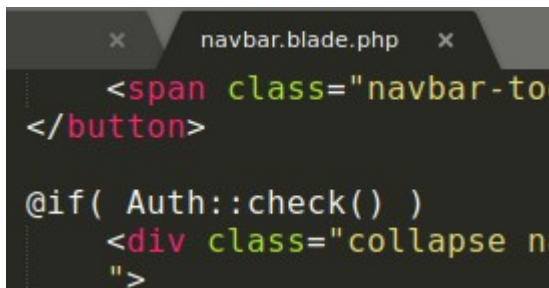
```
public function getHome()
{
    return redirect()->action('catalog\CatalogController@getIndex');
}
```



Xiaomi Band

Modifica la vista de login generada per Laravel (resources/views/auth/login.blade.php) perquè en lloc d'utilitzar el seu layout utilitzi el que vam crear en els primers exercicis (resources/views/layouts/master.blade.php).

Finalment edita la vista resources/views/partials/navbar.blade.php que havíem copiat de les plantilles i canvia la línia `@if(true || Auth::check())` per `@if(Auth::check())`. D'aquesta forma el menú només es mostrarà quan l'usuari estigui autenticat.



```
navbar.blade.php
<span class="navbar-to
</button>

@if( Auth::check() )
  <div class="collapse n
">
```


Exercici 4 - Afegir i editar pel·lícules (1 punt)

En primer lloc afegirem les rutes que ens faran falta per a recollir les dades a l'enviar els formularis. Editarem el fitxer de rutes i afegirem dues rutes (també protegides pel filtre auth):

- Una ruta de tipus POST per a la url catalog/create que apuntarà al mètode postCreate del controlador CatalogController.

```
// ----  
Route::post('/catalog/create', 'catalog\CatalogController@postCreate');
```

- I una altra ruta tipus PUT per a la url catalog/edit/{id} que apuntarà al mètode putEdit del controlador CatalogController.

```
// ----  
Route::put('/catalog/edit/{id}', 'catalog\CatalogController@putEdit');
```

A continuació editarem la vista catalog/edit.blade.php amb els següents canvis:

- Revisarem que el mètode d'enviament del formulari sigui de tipus PUT.

```
<form action="#" method="PUT">  
    {{ method_field('PUT') }}  
    {{ csrf_field() }}
```

- Modificarem tots els inputs perquè com a valor de camp posi el valor corresponent dels elements del vostre projecte. Per exemple, en el cas del videoclub, en el primer input hauríem d'afegir `value="{{ $pelicula->title }}"`. Feu el mateix per a la resta de camps del vostre projecte (videoclub: year , director , pòster i synopsis). En el cas del videoclub l'únic camp diferent serà el de synopsis ja que l' input és tipus textarea . En aquest cas el valor l'hauréu de posar directament entre l'etiqueta d'obertura i la de tancament.

```
<div class="form-group">
  <label for="title">Modificar Rellotge</label>
  <input type="text" name="title" id="title" class="
form-control" value="{{ $id['title'] }}">
</div>

<div class="form-group">
  <label for="year">Any</label>
  <input type="text" name="title" id="title" class="
form-control" value="{{ $id['year'] }}">
</div>

<div class="form-group">
  <label for="color">Color</label>
  <input type="text" name="title" id="title" class="
form-control" value="{{ $id['color'] }}">
</div>

<div class="form-group">
  <label for="imagen">Link Imatge</label>
  <input type="text" name="title" id="title" class="
form-control" value="{{ $id['imagen'] }}">
</div>

<div class="form-group">
  <label for="synopsis">Resum</label>
  <textarea name="synopsis" id="synopsis" class="
form-control" rows="3">{{ $id['synopsis'] }}</textarea>
</div>
```

Modificar Rellotge**Modificar Rellotge****Any****Color****Link Imatge****Resum**

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

Modificar Rellotge!

Finalment hem d'actualitzar el controlador `CatalogController` amb els dos nous mètodes. En els dos casos hem de fer servir la injecció de dependències per afegir la classe `Request` com a paràmetre d'entrada (revisa la secció "Dades d'entrada" del material de suport). A més per a cada mètode farem:

- Al mètode `postCreate` crearem una nova instància del model (`videoclub: Movie`), assignarem el valor de tots els camps d'entrada (`videoclub: title, year, director, pòster i synopsis`) i els guardem. Finalment, després de guardar, farem una redirecció a la ruta/`catalog`.

Afegir Rellotge

Nom

Casio ~ MTP-1303PL-2FVEF

Any

2018

Color

Blau

Link Imatge

isl-images-amazon.com/images/I/71qrYlXor5L._AC_UY445_.jpg

Resum

Peso del producto: 43,4 g

Número de modelo del producto: MTP-1303PL-2FVEF

ASIN: B07GBFKQPC

Producto en Amazon.es desde: 10 de agosto de 2018

Afegir Rellotge!

**Casio ~ MTP-1303PL-2FVEF ~ 2018****Color:** Blau**Descripció:**

fdadsf

Estat: Rellotge actualment llogat![Retornar rellotge](#)[Editar Rellotge](#)[Tornar al llistat](#)

- Al mètode putEdit busquem l'element (videoclub: pel·lícula) amb l'identificador passat per paràmetre, actualitzem els seus camps i els guardem. Finalment realitzarem una redirecció a la pantalla amb la vista detall de l'element en qüestió editat.

Modificar Rellotge

Modificar Rellotge

Any

Color

Link Imatge

Resum

Aquesta descripció ha sigut modificada.

Modificar Rellotge!

```
<form action="#" method="POST">
  {{ method_field('PUT') }}
  {{ csrf_field() }}
```



Casio ~ MTP-1303PL-2FVEF ~ 2018

Color: Blau

Descripció:

Aquesta descripció ha sigut modificada.

Estat: Relotge actualment llogat!

[Retornar rellotge](#)[Editar Relotge](#)[Tornar al llistat](#)