

GENER 2025

# SPRINT 2

# VIDEOS APP

REALIZADO POR:

Natalia Nowak

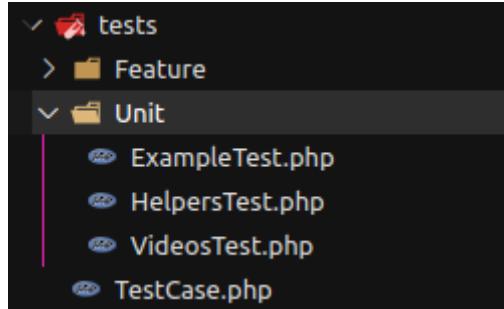
# ÍNDEX

<b>Configuracions inicials.....</b>	<b>2</b>
Correcció d'errors Sprint 1.....	2
Configuració PHPUnit.....	4
Instal·lació i configuració Larastan.....	5
<b>Migracions i models.....</b>	<b>7</b>
Crea la migració de video.....	7
Creació de models.....	8
<b>Controladors i helpers.....</b>	<b>9</b>
<b>Seeders.....</b>	<b>11</b>
<b>Disseny i vistes.....</b>	<b>12</b>
Layouts i rutes.....	12
Vista localhost.....	14
<b>Proves unitàries i funcionals.....</b>	<b>17</b>

# Configuracions inicials

## Correcció d'errors Sprint 1

- Tots els testos es troben a la carpeta tests/Unit



- Les credencials de l'usuari per defecte ara s'obtenen de config

```
public function createDefaultUser(): User
{
    $credentials = config('default-users.default_user');

    // Crear el usuario
    return User::query()->create([
        'name' => $credentials['name'],
        'email' => $credentials['email'],
        'password' => Hash::make($credentials['password']),
    ]);
}
```

- S'ha afegit l'assignació de l'equip a l'usuari

```
public function test_users_associated_with_team(): void
{
    $team = \App\Models\Team::factory()->create();
    $user = User::factory()->create([
        'name' => 'Haru',
        'email' => 'Haru03@gmail.com',
        'password' => Hash::make('passwordharu'),
    ]);
    $user->teams()->attach($team);

    $this->assertTrue($user->teams->contains($team));
}
```

- Tots els testos surten favorables

```
PASS Tests\Feature\VideosTest
✓ users can view videos                               0.02s
✓ users cannot view not existing videos              0.01s

Tests:    7 skipped, 46 passed (86 assertions)
Duration: 1.61s

alumne@natalianowak:~/Documents/2nDAM/GLOBAL_MP7-MP8-MP9/Sp
rint2/VideosAppNatalia2$ 
```

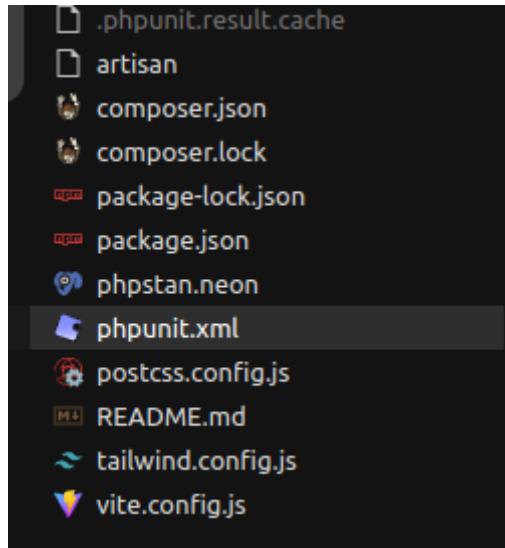
- S'ha afegit una assignació de `current_team_id` als testos, helpers i seeder

```
public function run(): void
{
    // Crear usuarios por defecto
    User::factory()->create([
        'name' => 'Usuari per Defecte',
        'email' => 'default@example.com',
        'password' => bcrypt('password'),
    ]);

    DefaultVideoHelper::createDefaultVideos();
}
```

## Configuració PHPUnit

- Editem l'arxiu PHPUnit.xml



- Descomentem les dues línies (DB\_CONNECTION i DB\_DATABASE) per a utilitzar un entorn de dades de prova

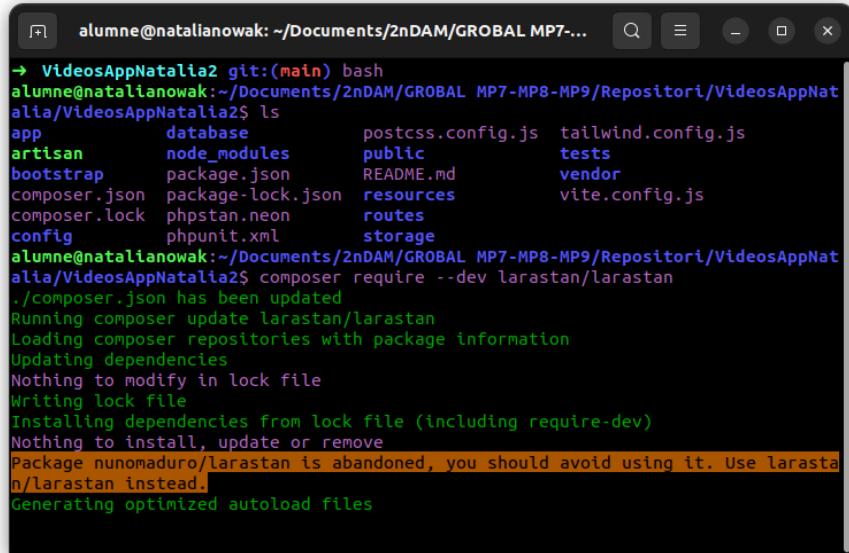
```
<php>
    <env name="APP_ENV" value="testing"/>
    <env name="APP_MAINTENANCE_DRIVER" value="file"/>
    <env name="BCRYPT_ROUNDS" value="4"/>
    <env name="CACHE_STORE" value="array"/>

    <env name="DB_CONNECTION" value="sqlite"/>
    <env name="DB_DATABASE" value=":memory:"/>

    <env name="MAIL_MAILER" value="array"/>
    <env name="PULSE_ENABLED" value="false"/>
    <env name="QUEUE_CONNECTION" value="sync"/>
    <env name="SESSION_DRIVER" value="array"/>
    <env name="TELESCOPE_ENABLED" value="false"/>
</php>
```

## Instal·lació i configuració Larastan

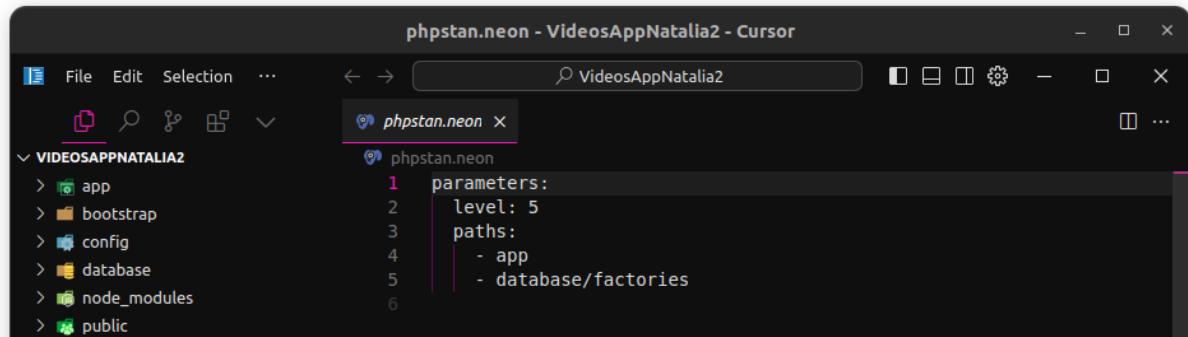
- Per a configurar Larastan executem la següent comanda en l'arrel del nostre projecte:



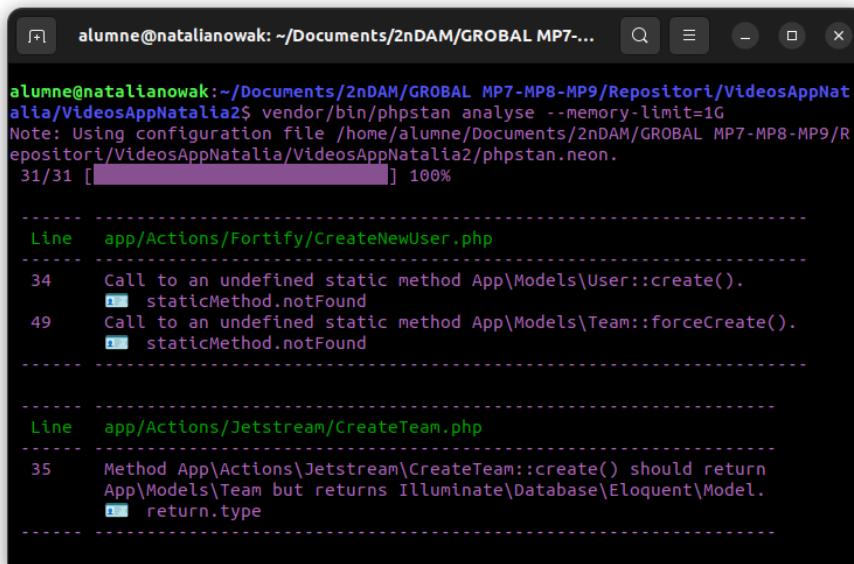
```
alumne@natalianowak: ~/Documents/2nDAM/GROBAL MP7-MP8-MP9/Repositori/VideosAppNatalia/VideosAppNatalia2$ ls
app           database      postcss.config.js  tailwind.config.js
artisan       node_modules  public            tests
bootstrap     package.json  README.md        vendor
composer.json package-lock.json resources      vite.config.js
composer.lock phpsstan.neon   routes
config        phpunite.xml  storage

alumne@natalianowak:~/Documents/2nDAM/GROBAL MP7-MP8-MP9/Repositori/VideosAppNatalia/VideosAppNatalia2$ composer require --dev larastan/larastan
./composer.json has been updated
Running composer update larastan/larastan
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Package nunomaduro/larastan is abandoned, you should avoid using it. Use larastan/larastan instead.
Generating optimized autoload files
```

- Després configurem l'arxiu phpsstan.neon



- Executem la següent comanda per a analitzar el codi:

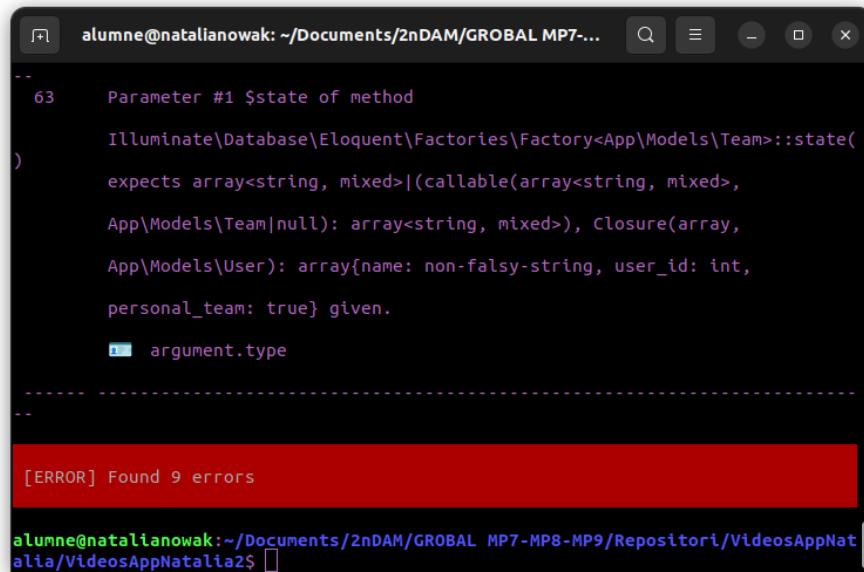


```
alumne@natalianowak: ~/Documents/2nDAM/GROBAL MP7-MP8-MP9/Repositori/VideosAppNatalia/VideosAppNatalia2$ vendor/bin/phpstan analyse --memory-limit=1G
Note: Using configuration file /home/alumne/Documents/2nDAM/GROBAL MP7-MP8-MP9/Repositori/VideosAppNatalia/VideosAppNatalia2/phpstan.neon.
31/31 [██████████] 100%

-----
Line  app/Actions/Fortify/CreateNewUser.php
34   Call to an undefined static method App\Models\User::create().
    staticMethod.notFound
49   Call to an undefined static method App\Models\Team::forceCreate().
    staticMethod.notFound
-----

-----
Line  app/Actions/Jetstream/CreateTeam.php
35   Method App\Actions\Jetstream\CreateTeam::create() should return
    App\Models\Team but returns Illuminate\Database\Eloquent\Model.
    return.type
```

Durant el desenvolupament del projecte, s'han detectat errors en Larastan associats al codi autogenerat per **Jetstream** i **Fortify**. Aquests errors, derivats de funcionalitats d'autenticació i interfícies d'usuari creades automàticament, no han estat escrits manualment per l'autor del projecte i no afecten el funcionament en temps d'execució. Per tant, es consideren propis dels paquets esmentats i no del desenvolupament específic del projecte.



The screenshot shows a terminal window with the following content:

```
alumne@natalianowak: ~/Documents/2nDAM/GROBAL MP7-...
--> 63     Parameter #1 $state of method
        Illuminate\Database\Eloquent\Factories\Factory<App\Models\Team>::state(
)
    expects array<string, mixed>|(callable(array<string, mixed>,
        App\Models\Team|null): array<string, mixed>), Closure(array,
        App\Models\User): array{name: non-falsy-string, user_id: int,
        personal_team: true} given.
        argument.type
-----
-->
[ERROR] Found 9 errors
alumne@natalianowak: ~/Documents/2nDAM/GROBAL_MP7-MP8-MP9/Repositori/VideosAppNatalia/VideosAppNatalia2$
```

- Exemple de codis corregits:

The screenshot shows a code editor window with the title "Video.php - VideosAppNatalia2 - Cursor". The file path is "app > Models > Video.php > Video > \$series\_id". The code editor displays the following PHP code:

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Carbon\Carbon;

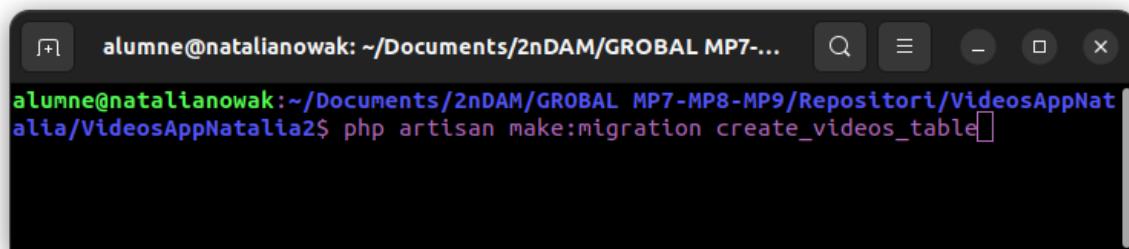
/**
 * App\Models\Video
 *
 * @property string|null $title
 * @property string|null $description
 * @property string|null $url
 * @property \Carbon\Carbon|null $published_at
 * @property string|null $previous
 * @property string|null $next
 * @property int|null $series_id
*/
class Video extends Model
{
    use HasFactory;

    protected $fillable = [
        'title', 'description', 'url', 'published_at', 'previous', 'n
    ];
}
```

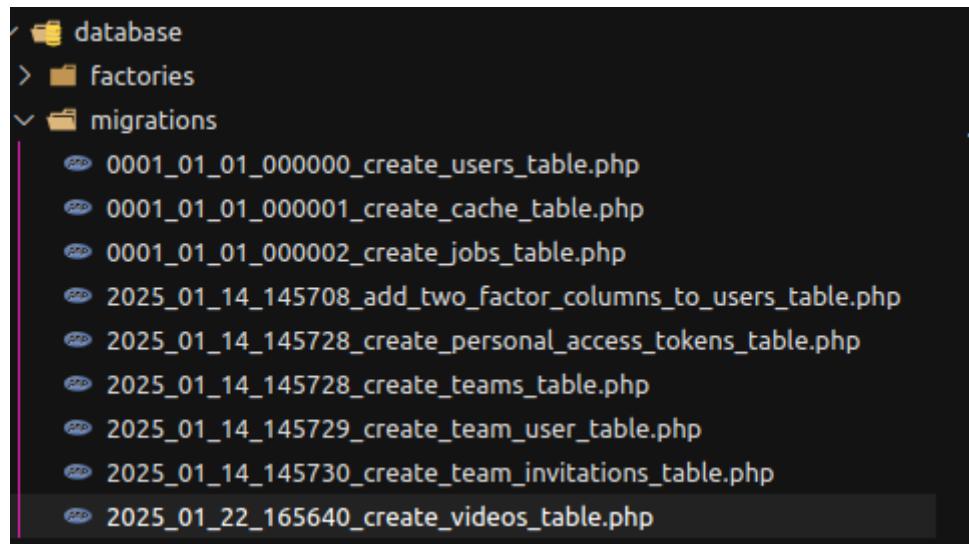
## Migracions i models

Crea la migració de video

- Creem les migracracions i les configurem en els camps corresponents



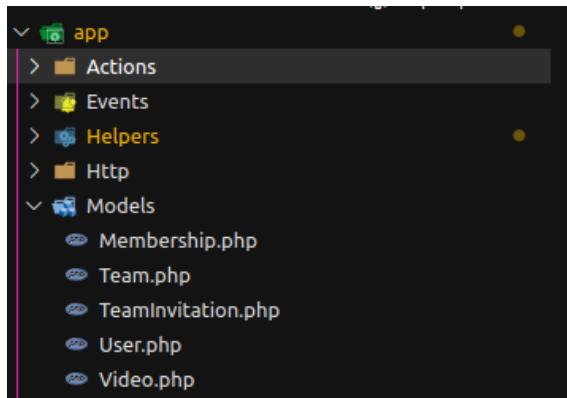
```
alumne@natalianowak: ~/Documents/2nDAM/GROBAL MP7-... 
alumne@natalianowak:~/Documents/2nDAM/GROBAL MP7-MP8-MP9/Repositori/VideosAppNatalia/VideosAppNatalia2$ php artisan make:migration create_videos_table
```



```
public function up()
{
    Schema::create('videos', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->text('description')->nullable();
        $table->string('url');
        $table->timestamp('published_at')->nullable();
        $table->unsignedBigInteger('previous')->nullable();
        $table->unsignedBigInteger('next')->nullable();
        $table->unsignedBigInteger('series_id')->nullable();
        $table->timestamps();
    });
}
```

## Creació de models

- Creem els models Video i User dins de la carpeta app/models



- En el model Video implementem tots els formats de data necessaris per al camp *published at*

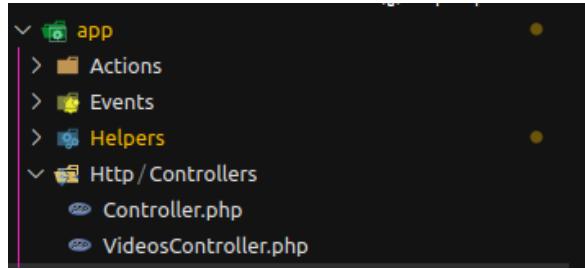
```
public function getFormattedPublishedAtAttribute()
{
    return $this->published_at ? $this->published_at->format('d \d\e F \d\e Y') : null;
}

/**
 * Obtiene la fecha de publicación en formato humano.
 *
 * @return string|null
 */
public function getFormattedForHumansPublishedAtAttribute()
{
    return $this->published_at ? $this->published_at->diffForHumans() : null;
}

/**
 * Obtiene el timestamp de la fecha de publicación.
 *
 * @return int|null
 */
public function getPublishedAtTimestampAttribute()
{
    return $this->published_at ? $this->published_at->timestamp : null;
}
```

# Controladors i helpers

- Creació de VideosController i implementació de *testedBy* i *show*

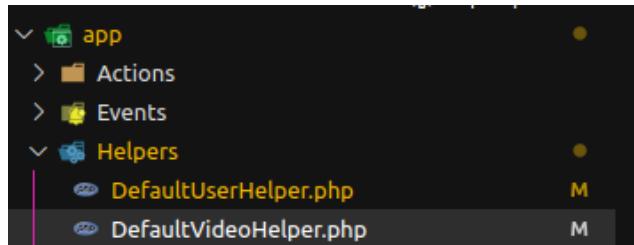


```
public function show(int $id): View
{
    $video = Video::query()->findOrFail($id);

    return view('videos.show', compact('video'));
}

/**
 * Devuelve el nombre de quien probó la funcionalidad.
 *
 * @return \Illuminate\Http\JsonResponse
 */
public function testedBy(): JsonResponse
{
    return response()->json(['tested_by' => 'Natalia Nowak']);
}
```

- Creació de UserHelper i VideoHelper per a definir els usuaris i vídeos per defecte

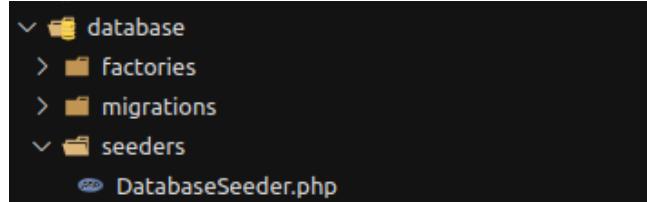


```
public function createDefaultUser(): User
{
    $credentials = config('default-users.default_user');

    return User::query()->create([
        'name' => $credentials['name'],
        'email' => $credentials['email'],
        'password' => Hash::make($credentials['password']),
    ]);
}
```

# Seeders

- Configuració de DatabaseSeeder per a la creació d'usuaris i vídeos per defecte



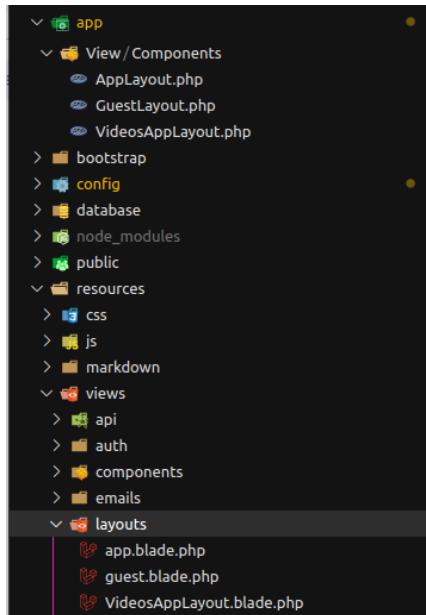
```
class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        // Crear usuarios por defecto
        User::factory()->create([
            'name' => 'Usuari per Defecte',
            'email' => 'default@example.com',
            'password' => bcrypt('password'),
        ]);

        DefaultVideoHelper::createDefaultVideos();
    }
}
```

# Disseny i vistes

## Layouts i rutes

- Creació del layout en la carpeta resources i en la carpeta:



- Construcció del mètode per a que retorna la vista corresponent

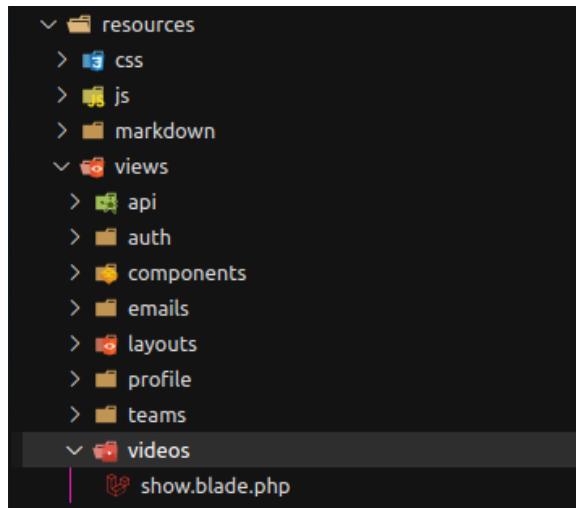
```
class VideosAppLayout extends Component
{
    /**
     * Create a new component instance.
     */
    public function __construct()
    {
        //
    }

    /**
     * Get the view / contents that represent the component.
     */
    public function render(): View|Closure|string
    {
        return view('layouts.VideosAppLayout');
    }
}
```

- Configuració de la ruta corresponent a vídeos

```
Route::get('videos/{video}', [VideosController::class, 'show'])->name('videos.show');
```

- Creació de videos.show per a mostrar el vídeo amb el seu títol i descripció

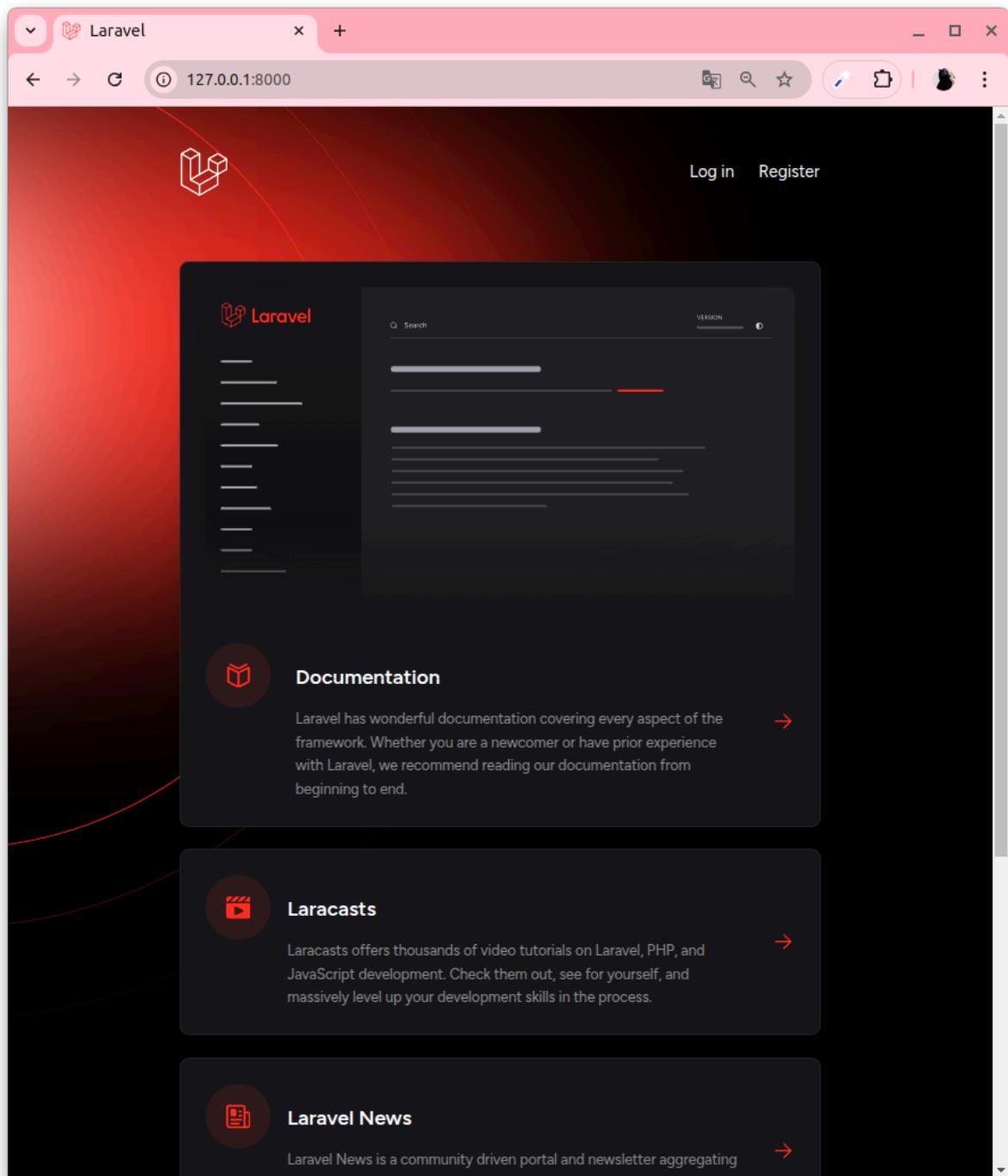


```
@extends('layouts.VideosAppLayout')

@section('content')
    <h2 class="text-2xl font-bold">{{ $video->title }}</h2>
    <p>{{ $video->description }}</p>
    <a href="{{ $video->url }}" target="_blank">Ver video</a>
@endsection
```

## Vista localhost

- Pàgina principal amb el *Log in i Register*:



- Prova de registre i inici de sessió en els usuaris per defecte

Laravel

127.0.0.1:8000/register

Name  
Usuari per Defecte

Email  
default@example.com

Password  
.....

Confirm Password  
.....

Already registered? [REGISTER](#)

Laravel

127.0.0.1:8000/login

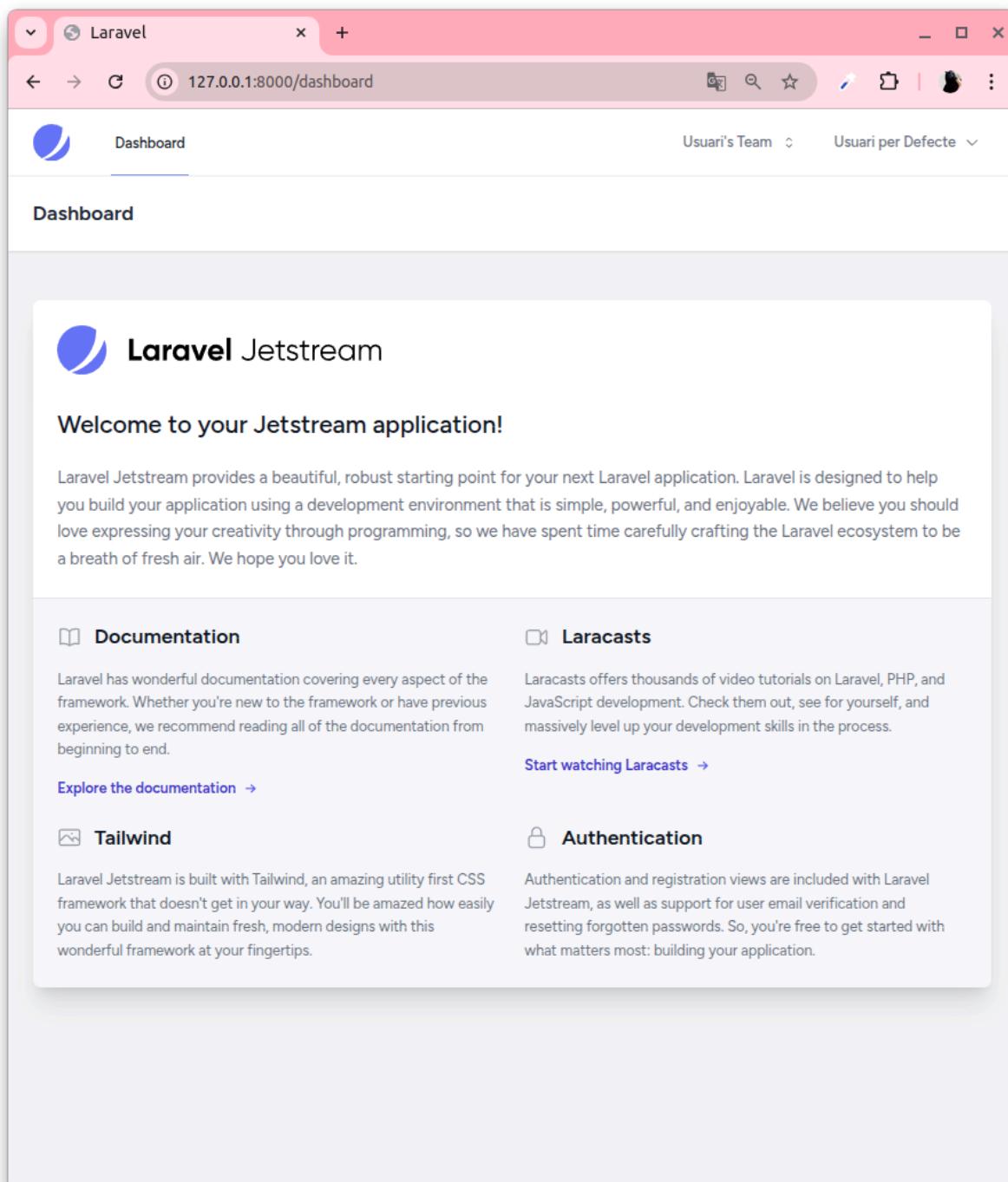
Email  
default@example.com

Password  
.....

Remember me

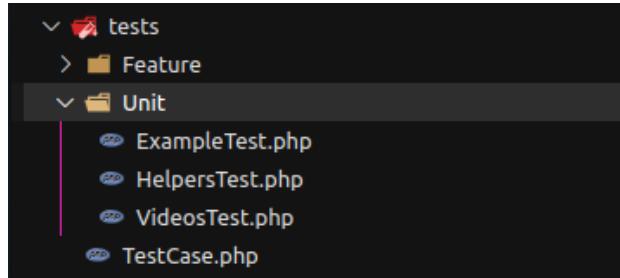
[Forgot your password?](#) [LOG IN](#)

- Vista una vegada registrat



# Proves unitàries i funcionals

- Creació dels respectius testos en la carpeta unit



- Crearem HelpersTest per a verificar la creació dels usuaris per defecte

```
class HelpersTest extends TestCase
{
    use RefreshDatabase;

    public function test_default_user_creation(): void
    {
        $user = User::factory()->create([
            'name' => 'Taco',
            'email' => 'taco12@gmail.com',
            'password' => Hash::make('password'),
        ]);

        $this->assertNotNull($user);
        $this->assertEquals('Taco', $user->name);
        $this->assertEquals('taco12@gmail.com', $user->email);
        $this->assertTrue(Hash::check('password', $user->password));
    }

    public function test_default_teacher_creation(): void {
        $profe = User::factory()->create([
            'name' => 'Kumy',
            'email' => 'kumy12@gmail.com',
            'password' => Hash::make('passwordprofe'),
        ]);
        $this->assertNotNull($profe);
        $this->assertEquals('Kumy', $profe->name);
        $this->assertEquals('kumy12@gmail.com', $profe->email);
        $this->assertTrue(Hash::check('passwordprofe', $profe->password));
    }
}
```

- Seguidament creem VideosTest per a implementar les funcions de *published\_at*

```

public function test_can_get_formatted_published_at_date()
{
    $video = Video::factory()->create([
        'published_at' => Carbon::create(2023, 1, 1, 12, 0, 0),
    ]);

    $this->assertEquals('01 de January de 2023', $video->formatted_published_at);
}

public function test_can_get_formatted_published_at_date_when_not_published()
{
    $video = Video::factory()->create([
        'published_at' => null,
    ]);

    $this->assertNull($video->formatted_published_at);
}

```

- Comprovem que els testos funcionen correctament

```

alumne@natalianowak: ~/Documents/2nDAM/GLOBAL MP7-...
alumne@natalianowak: ~/Documents/2...

PASS Tests\Feature\VideosTest
✓ users can view videos
✓ users cannot view not existing videos
0.02s
0.01s
0.01s

Tests: 7 skipped, 46 passed (86 assertions)
Duration: 1.51s

alumne@natalianowak:~/Documents/2nDAM/GLOBAL MP7-MP8-MP9/Repositori/VideosAppNatalia/VideosAppNatalia2$ 

```

