Projecte GLOBAL: VideosApp

5e SPRINT

Professor: Jordi Vega
Assignatures: M7, M8 i M9
Nom: Marc
Cognoms: Pallares Pino

INDEX

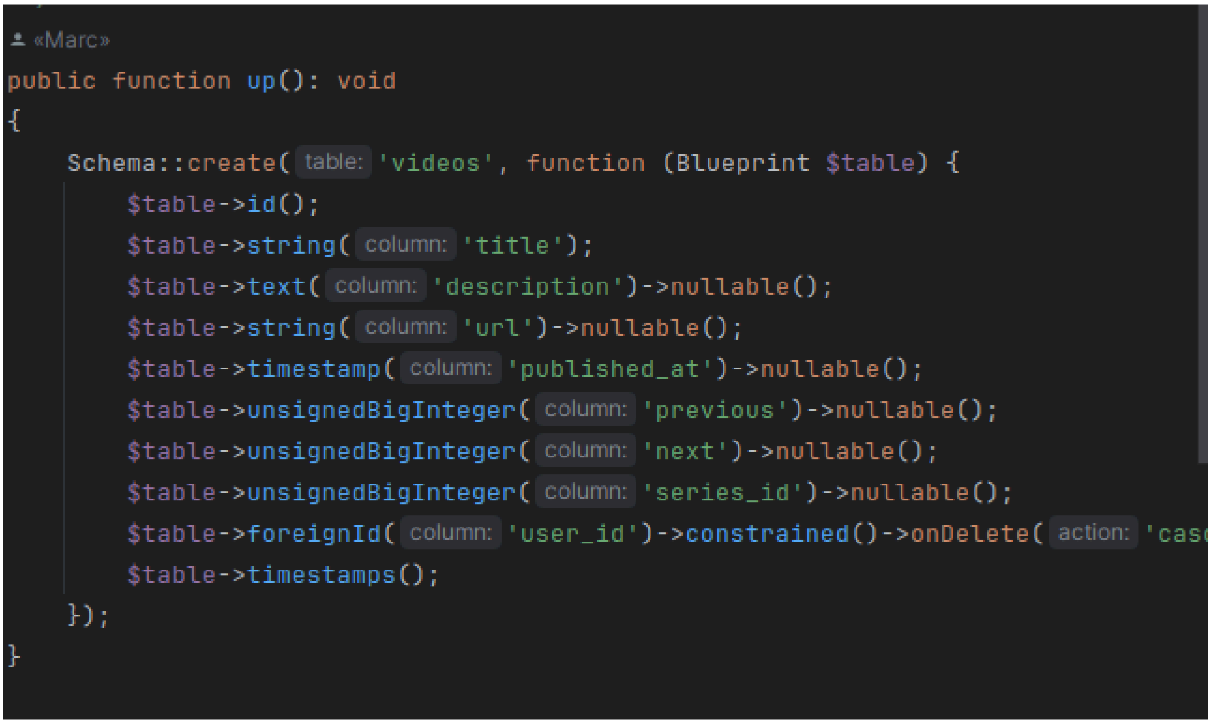
INDEX

5e Sprint

Descripció del 5e Sprint:

- Afegir el camp user_id a la taula de videos. Per a que al crear un video es guardi l'usuari que l'ha afegit. Per tant, s'haurà de modificar el controller, model, helpers…

  Fico el camp al final del create



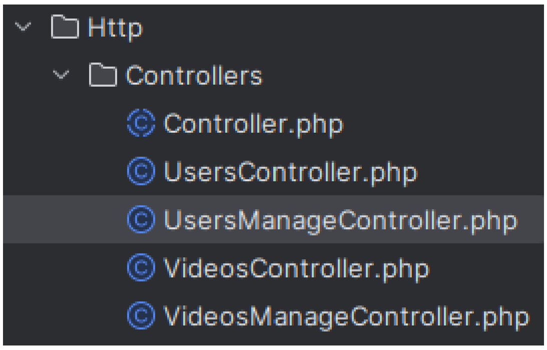
```
«Marc»
public function up(): void
{
    Schema::create( table: 'videos', function (Blueprint $table) {
        $table->id();
        $table->string( column: 'title');
        $table->text( column: 'description')->nullable();
        $table->string( column: 'url')->nullable();
        $table->timestamp( column: 'published_at')->nullable();
        $table->unsignedBigInteger( column: 'previous')->nullable();
        $table->unsignedBigInteger( column: 'next')->nullable();
        $table->unsignedBigInteger( column: 'series_id')->nullable();
        $table->foreignId( column: 'user_id')->constrained()->onDelete( action: 'casc
        $table->timestamps();
    });
}
```

- En cas que al modificar el codi falla algun test d'un sprint anterior, s'han d'arreglar.

FET

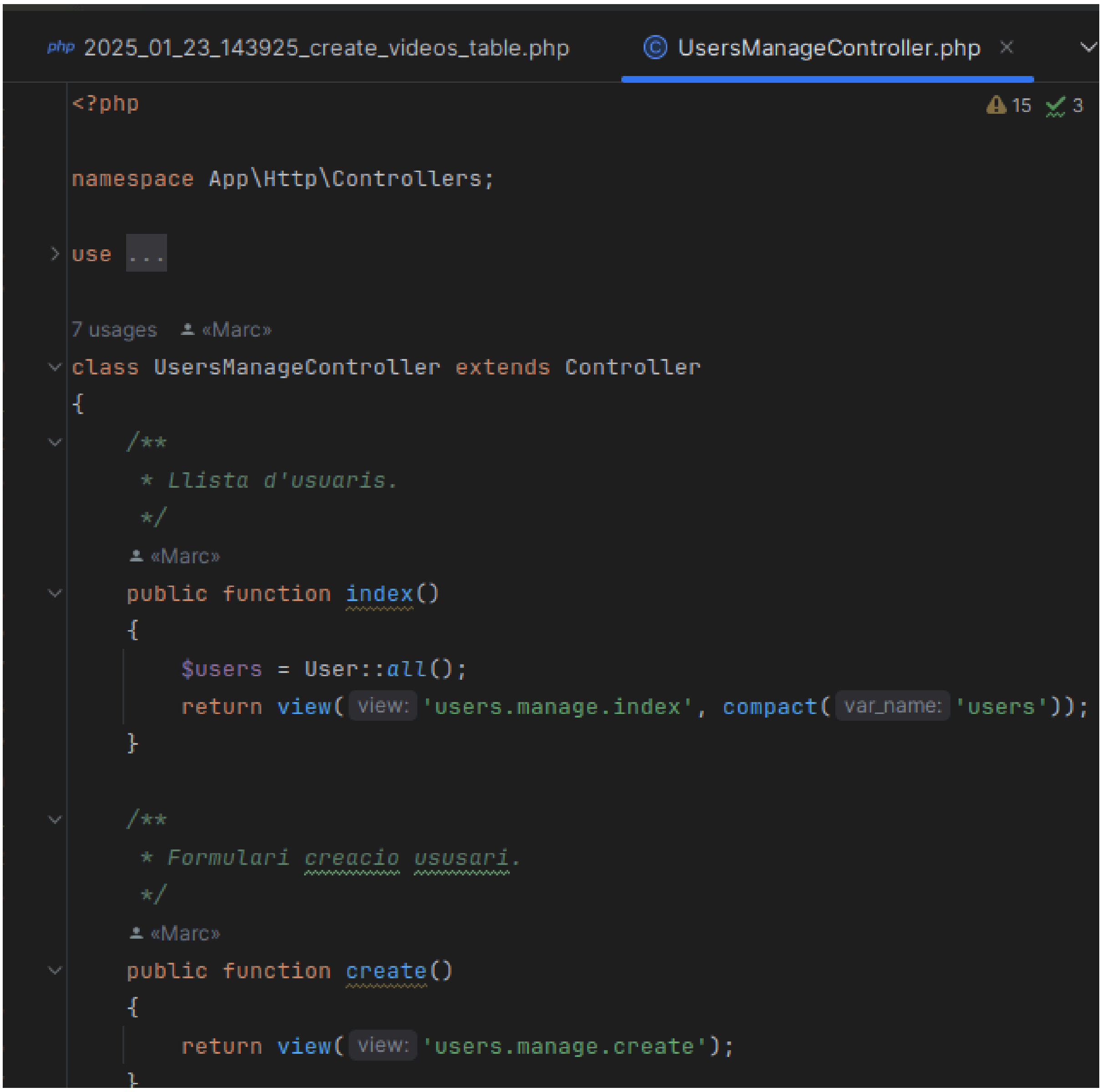
- Crear usersManageController en les funcions testedby, index, store, edit, update, delete i destroy.

Esquema per veure el arxiu creat



```
Http
  Controllers
    Controller.php
    UsersController.php
    UsersManageController.php
    VideosController.php
    VideosManageController.php
```

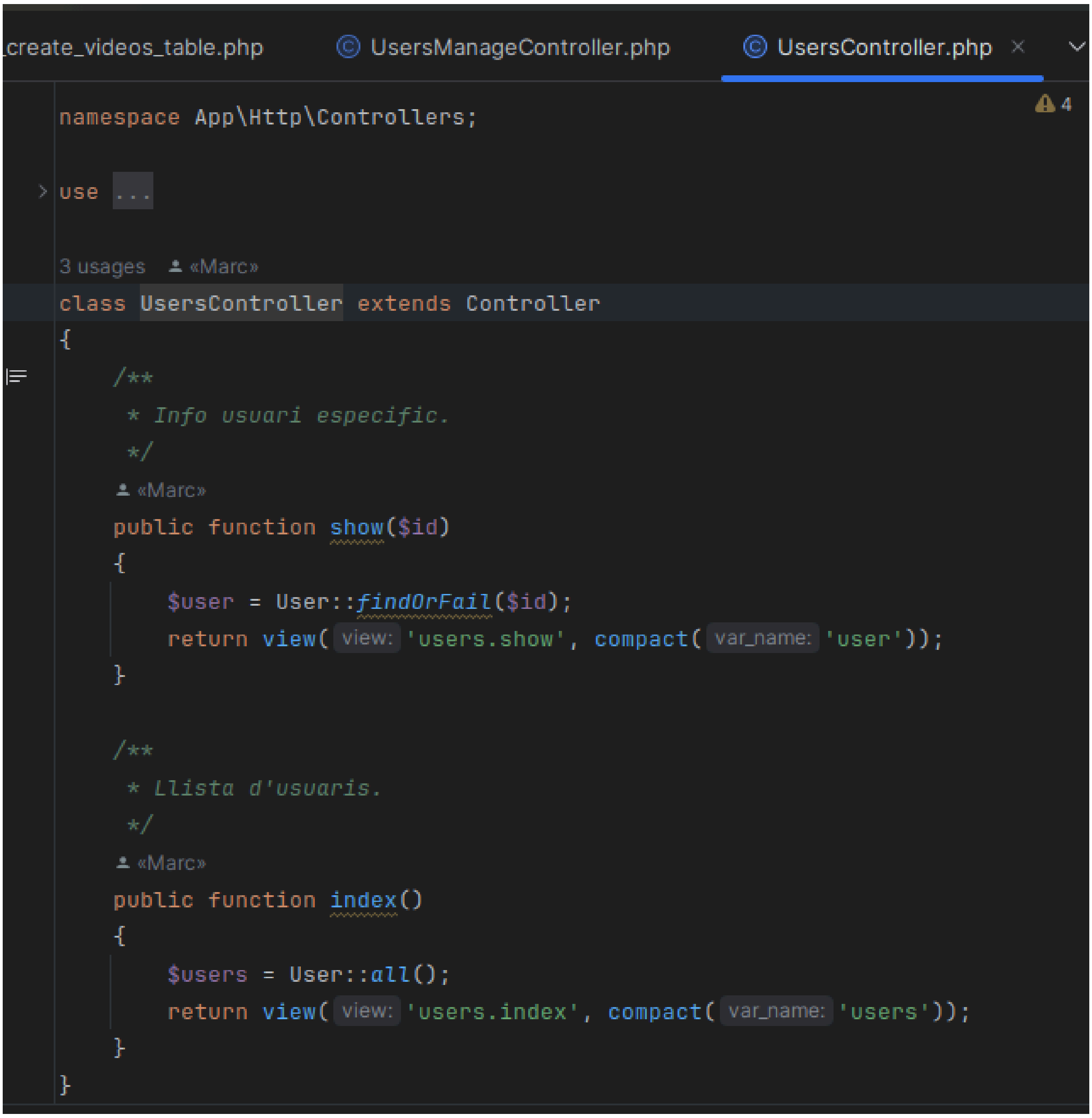
Les funcions testedby, index, store, edit, update, delete i destroy.


2025_01_23_143925_create_videos_table.php
UsersManageController.php
15
3

```
<?php

namespace App\Http\Controllers;

use ...

7 usages  «Marc»
class UsersManageController extends Controller
{
    /**
     * Llista d'usuaris.
     */
    «Marc»
    public function index()
    {
        $users = User::all();
        return view( view: 'users.manage.index', compact( var_name: 'users'));
    }

    /**
     * Formulari creacio ususari.
     */
    «Marc»
    public function create()
    {
        return view( view: 'users.manage.create');
    }
```

- Crear la funció index i show a usersController.

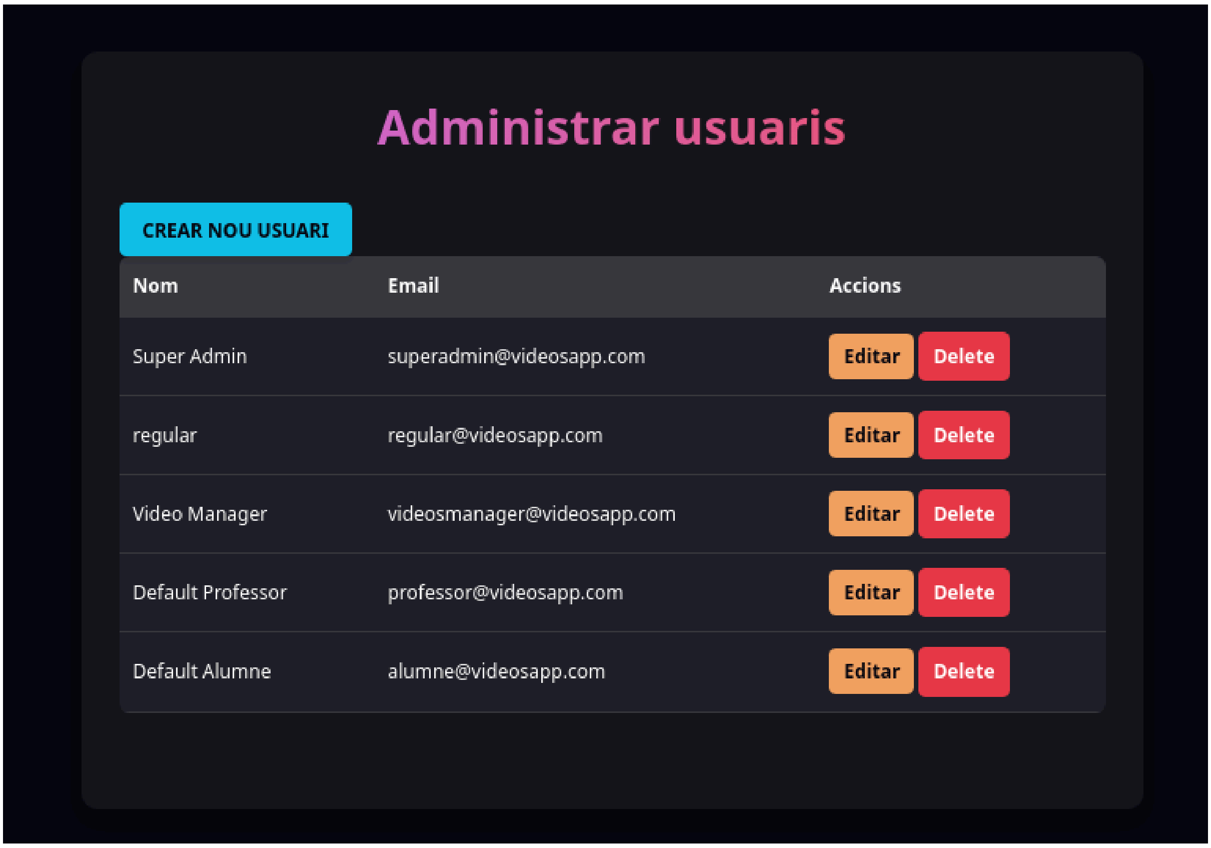
Les funcions:



```php
namespace App\Http\Controllers;

use ...

class UsersController extends Controller
{
    /**
     * Info usuari especific.
     */
    public function show($id)
    {
        $user = User::findOrFail($id);
        return view( view: 'users.show', compact( var_name: 'user'));
    }

    /**
     * Llista d'usuaris.
     */
    public function index()
    {
        $users = User::all();
        return view( view: 'users.index', compact( var_name: 'users'));
    }
}
```

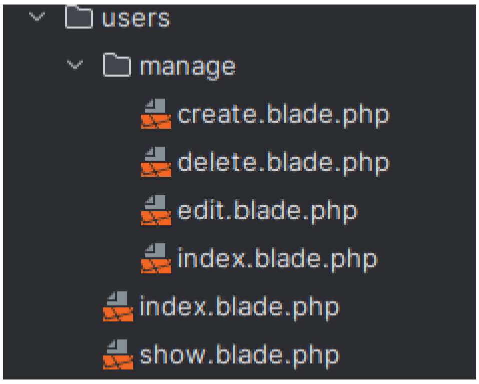
- Crear les vistes per al CRUD que només poden veure-ho els que tinguin els permisos adients: resources/views/users/manage/index.blade.php, resources/views/users/manage/create.blade.php, resources/views/users/manage/edit.blade.php, resources/views/users/manage/delete.blade.php.

la vista:



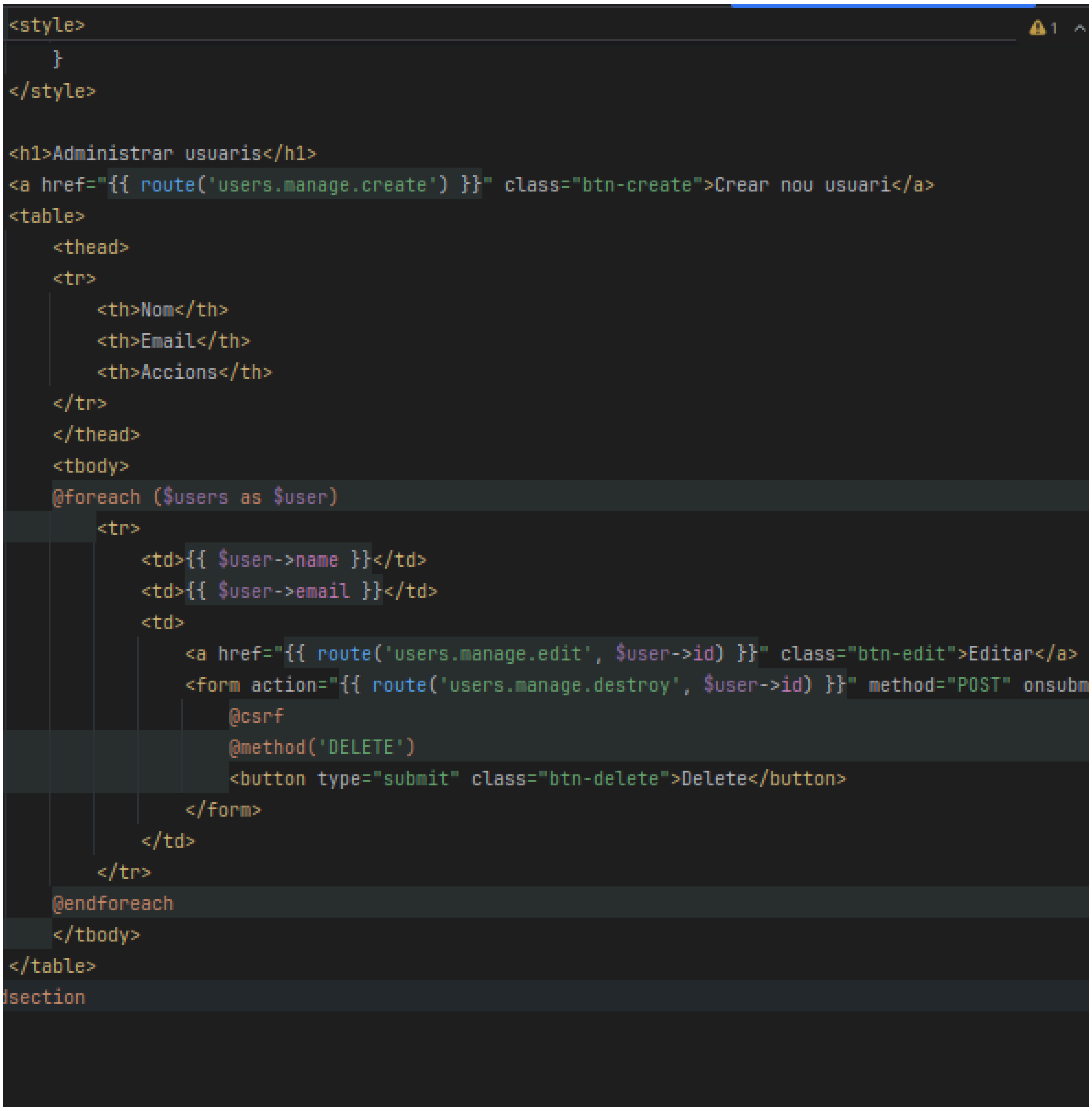
Administrar usuaris

Nom	Email	Accions
Super Admin	superadmin@videosapp.com	Editar Delete
regular	regular@videosapp.com	Editar Delete
Video Manager	videosmanager@videosapp.com	Editar Delete
Default Professor	professor@videosapp.com	Editar Delete
Default Alumne	alumne@videosapp.com	Editar Delete



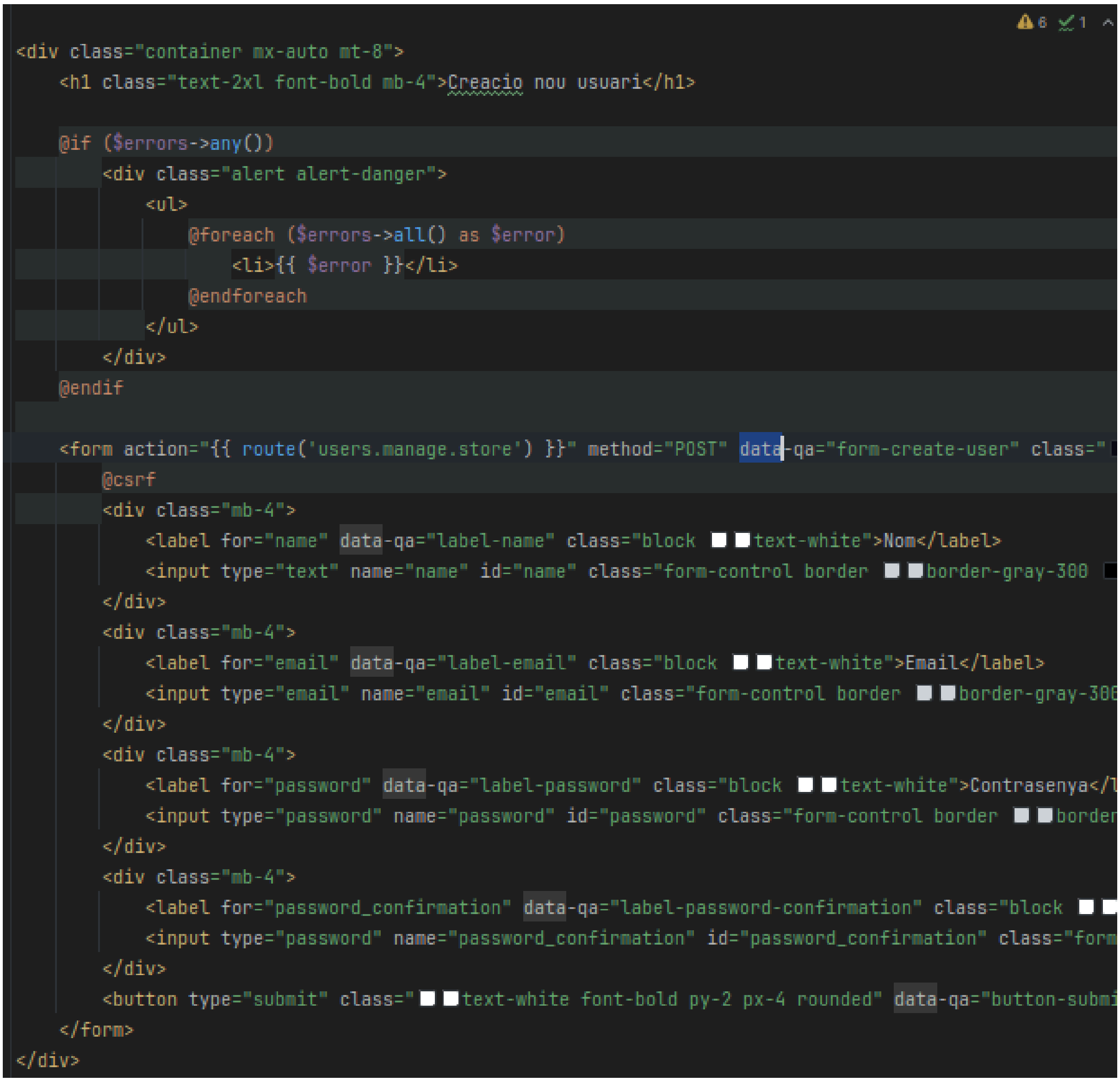
```
users
  manage
    create.blade.php
    delete.blade.php
    edit.blade.php
    index.blade.php
  index.blade.php
  show.blade.php
```

- A la vista index.blade.php, afegir la taula del CRUD d'usuaris.



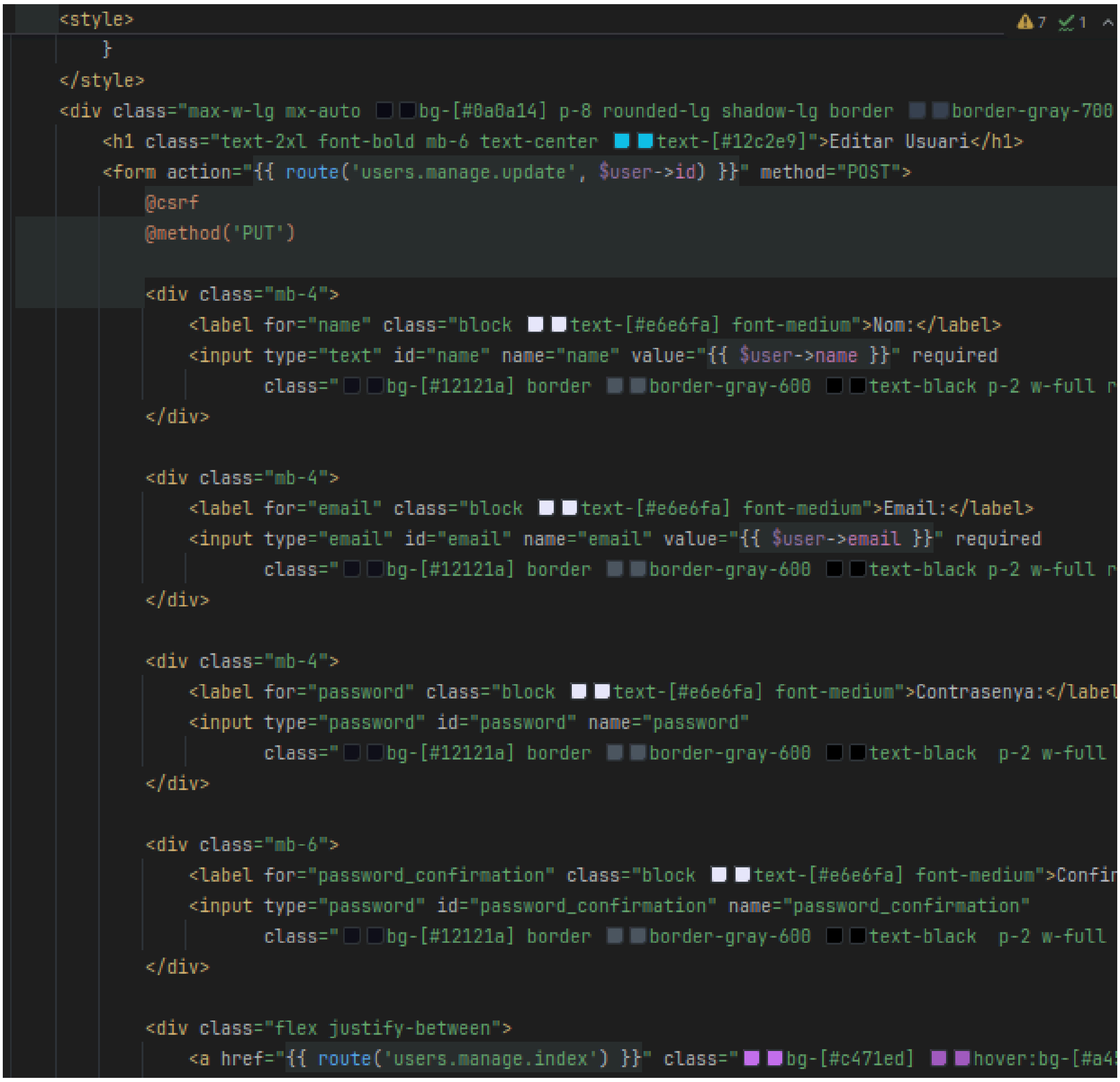
```
<style>
    }
</style>

<h1>Administrar usuaris</h1>
<a href="{{ route('users.manage.create') }}" class="btn-create">Crear nou usuari</a>
<table>
    <thead>
    <tr>
        <th>Nom</th>
        <th>Email</th>
        <th>Accions</th>
    </tr>
    </thead>
    <tbody>
    @foreach ($users as $user)
        <tr>
            <td>{{ $user->name }}</td>
            <td>{{ $user->email }}</td>
            <td>
                <a href="{{ route('users.manage.edit', $user->id) }}" class="btn-edit">Editar</a>
                <form action="{{ route('users.manage.destroy', $user->id) }}" method="POST" onsubm
                    @csrf
                    @method('DELETE')
                    <button type="submit" class="btn-delete">Delete</button>
                </form>
            </td>
        </tr>
    @endforeach
    </tbody>
</table>
dsection
```

- A la vista create.blade.php, afegir el formulari per posar els usuaris, s’ha d’utilitzar l’atribut data-qa per a que sigui més fàcil identificar per als testos.



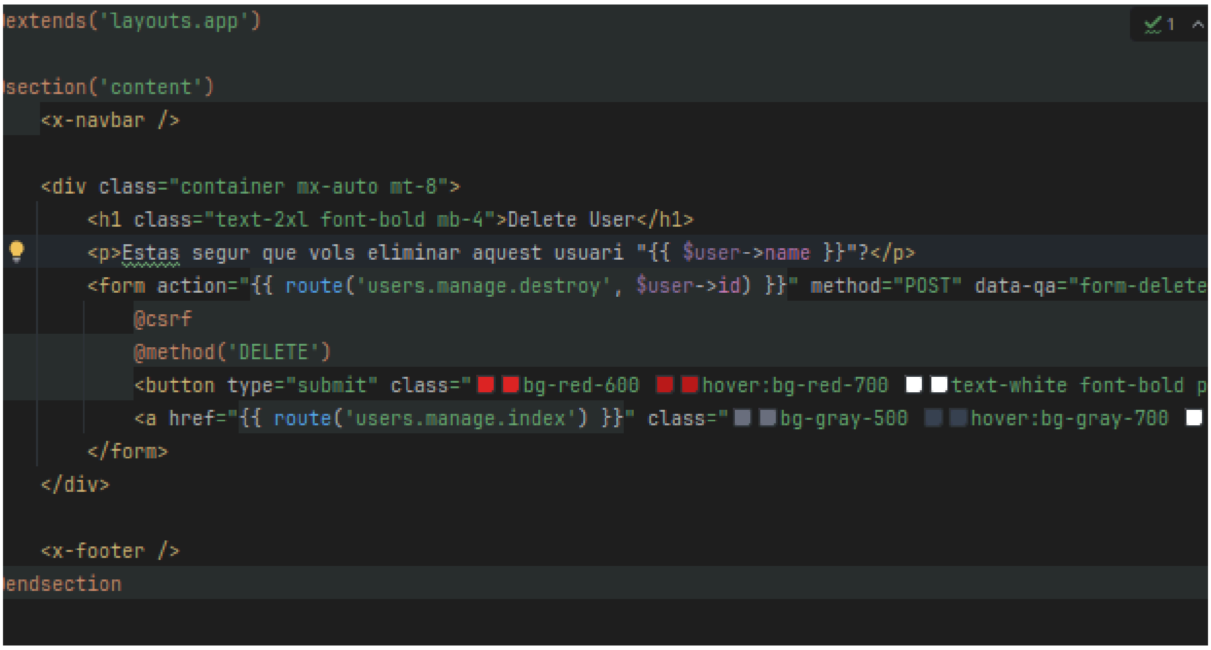
```
<div class="container mx-auto mt-8">
    <h1 class="text-2xl font-bold mb-4">Creacio nou usuari</h1>

    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('users.manage.store') }}" method="POST" data-qa="form-create-user" class="
        @csrf
        <div class="mb-4">
            <label for="name" data-qa="label-name" class="block text-white">Nom</label>
            <input type="text" name="name" id="name" class="form-control border border-gray-300
        </div>
        <div class="mb-4">
            <label for="email" data-qa="label-email" class="block text-white">Email</label>
            <input type="email" name="email" id="email" class="form-control border border-gray-30
        </div>
        <div class="mb-4">
            <label for="password" data-qa="label-password" class="block text-white">Contrasenya</l
            <input type="password" name="password" id="password" class="form-control border border
        </div>
        <div class="mb-4">
            <label for="password_confirmation" data-qa="label-password-confirmation" class="block
            <input type="password" name="password_confirmation" id="password_confirmation" class="form
        </div>
        <button type="submit" class="text-white font-bold py-2 px-4 rounded" data-qa="button-submi
    </form>
</div>
```

- A la vista edit.blade.php, afegir la taula del CRUD d'usuaris.



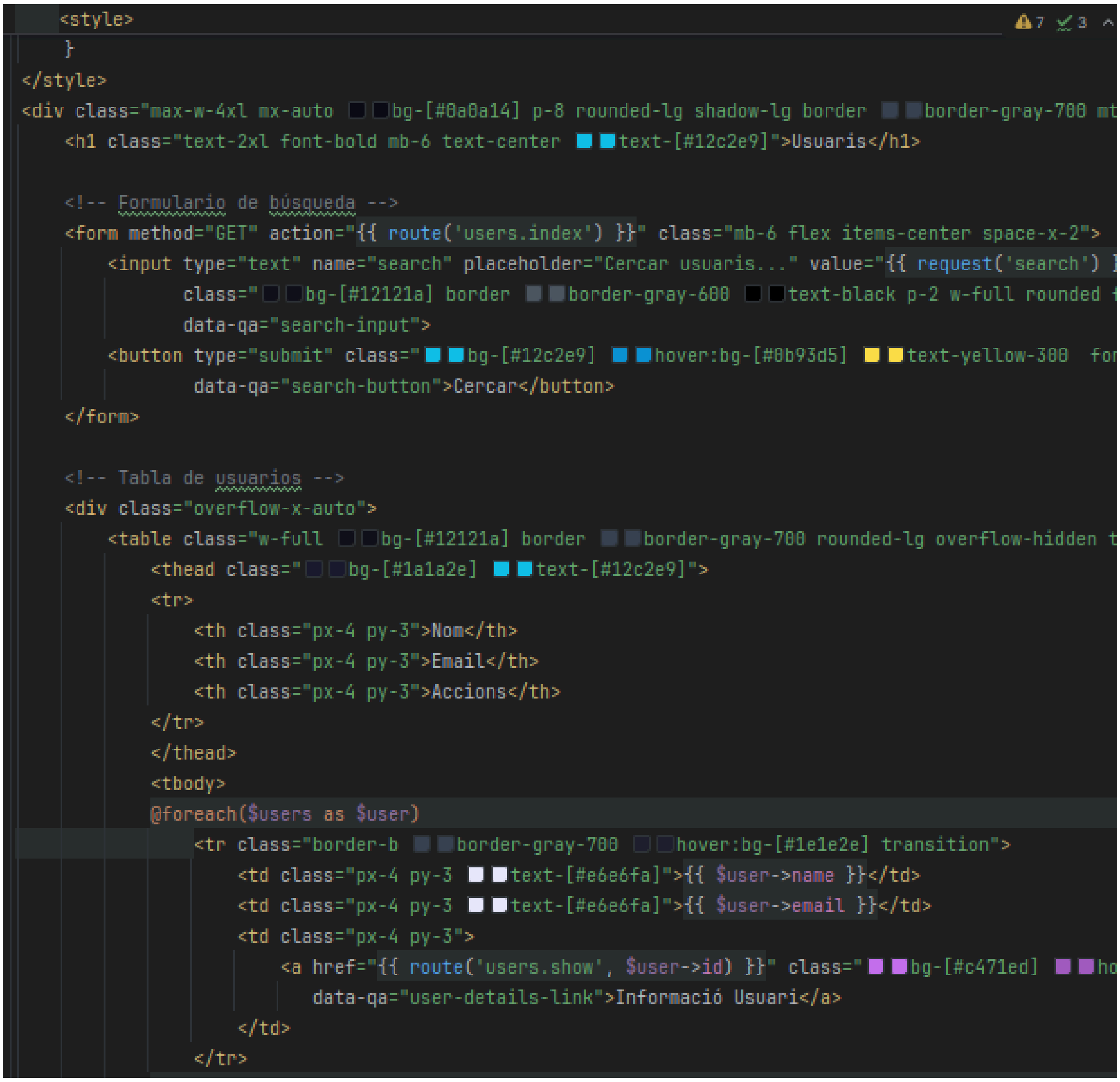
```
<style>
    }
</style>
<div class="max-w-lg mx-auto bg-[#0a0a14] p-8 rounded-lg shadow-lg border border-gray-700
    <h1 class="text-2xl font-bold mb-6 text-center text-[#12c2e9]">Editar Usuari</h1>
    <form action="{{ route('users.manage.update', $user->id) }}" method="POST">
        @csrf
        @method('PUT')

        <div class="mb-4">
            <label for="name" class="block text-[#e6e6fa] font-medium">Nom:</label>
            <input type="text" id="name" name="name" value="{{ $user->name }}" required
                   class="bg-[#12121a] border border-gray-600 text-black p-2 w-full r
        </div>

        <div class="mb-4">
            <label for="email" class="block text-[#e6e6fa] font-medium">Email:</label>
            <input type="email" id="email" name="email" value="{{ $user->email }}" required
                   class="bg-[#12121a] border border-gray-600 text-black p-2 w-full r
        </div>

        <div class="mb-4">
            <label for="password" class="block text-[#e6e6fa] font-medium">Contrasenya:</label
            <input type="password" id="password" name="password"
                   class="bg-[#12121a] border border-gray-600 text-black  p-2 w-full
        </div>

        <div class="mb-6">
            <label for="password_confirmation" class="block text-[#e6e6fa] font-medium">Confir
            <input type="password" id="password_confirmation" name="password_confirmation"
                   class="bg-[#12121a] border border-gray-600 text-black  p-2 w-full
        </div>

        <div class="flex justify-between">
            <a href="{{ route('users.manage.index') }}" class="bg-[#c471ed] hover:bg-[#a4
```

- A la vista delete.blade.php, afegir la confirmació de l'eliminació de l'usuari.



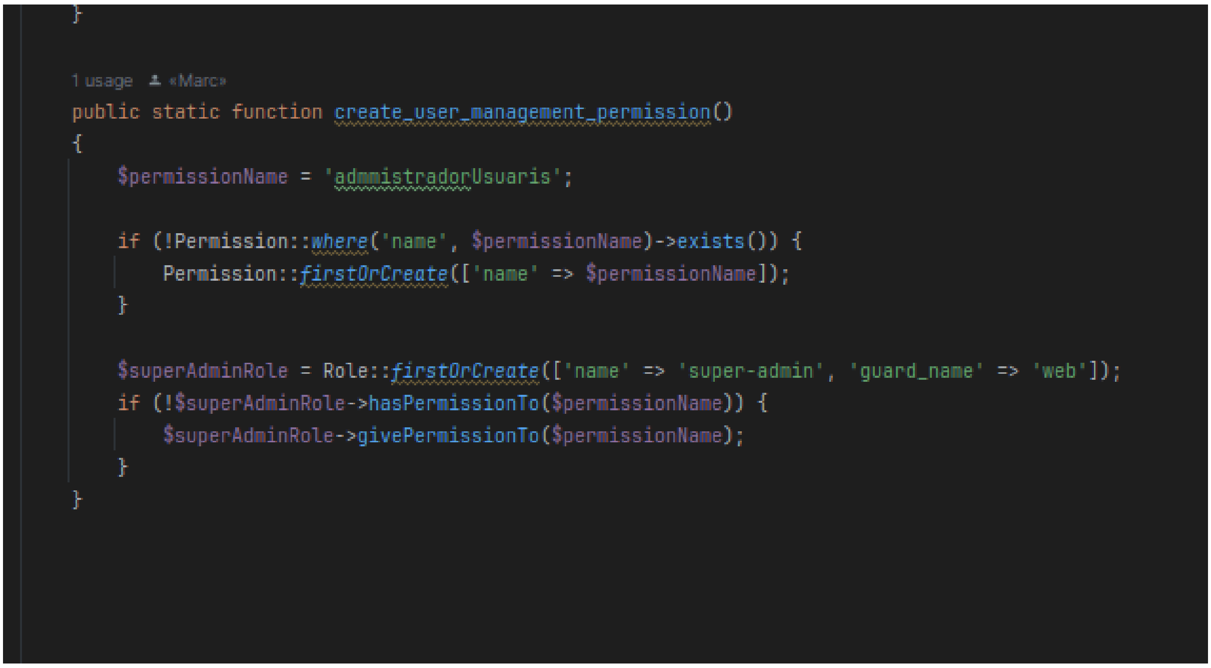
```
extends('layouts.app')

section('content')
    <x-navbar />

    <div class="container mx-auto mt-8">
        <h1 class="text-2xl font-bold mb-4">Delete User</h1>
        <p>Estas segur que vols eliminar aquest usuari "{{ $user->name }}"?</p>
        <form action="{{ route('users.manage.destroy', $user->id) }}" method="POST" data-qa="form-delete
            @csrf
            @method('DELETE')
            <button type="submit" class="bg-red-600 hover:bg-red-700 text-white font-bold p
            <a href="{{ route('users.manage.index') }}" class="bg-gray-500 hover:bg-gray-700
        </form>
    </div>

    <x-footer />
endsection
```

- Crear la vista de resources/views/users/index.blade.php on es vegin tots els usuaris i es puguin buscar i al clicar a l'usuari que porti al detall de l'usuari i els seus videos.



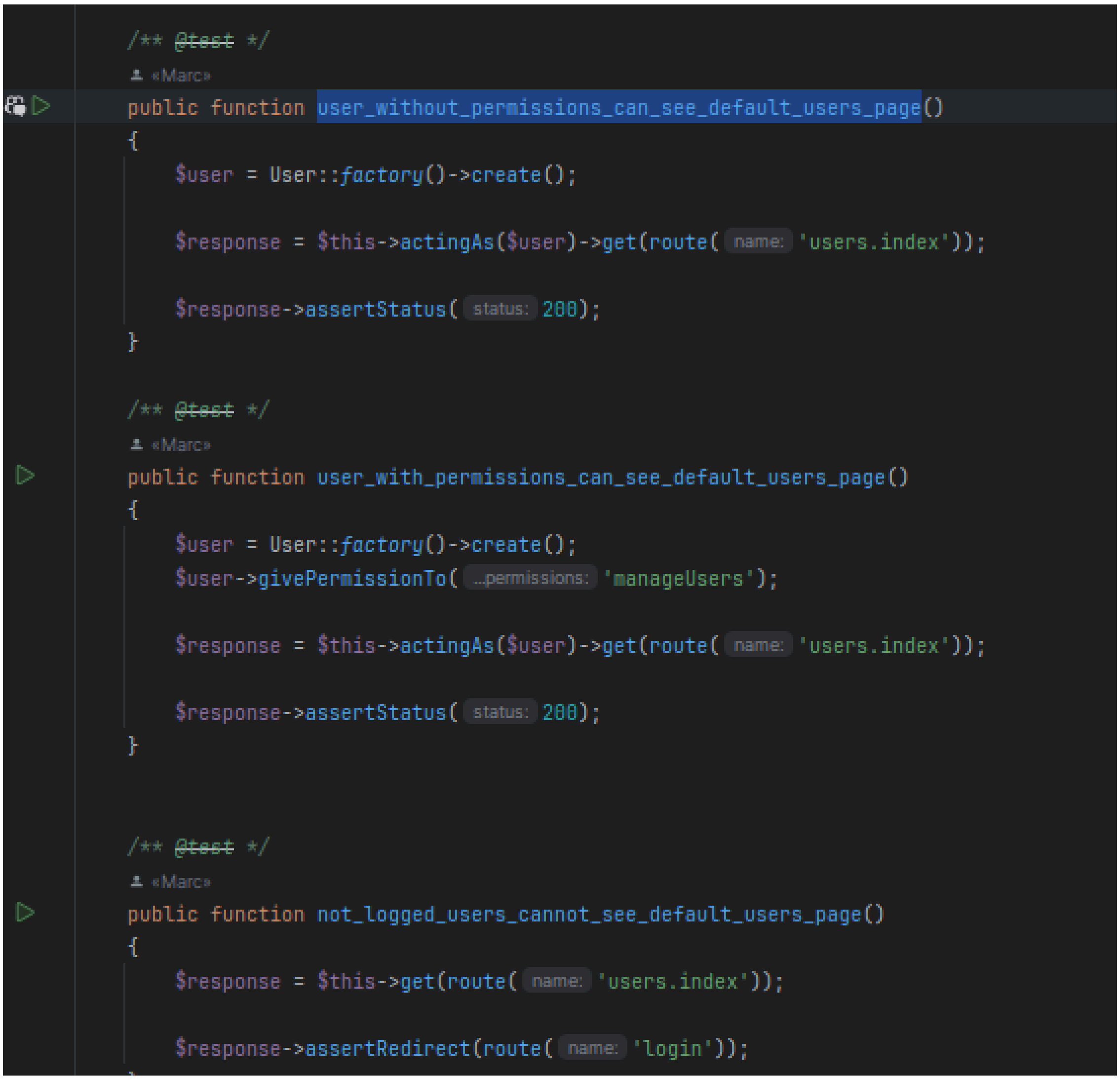
```
<style>
    }
</style>
<div class="max-w-4xl mx-auto bg-[#0a0a14] p-8 rounded-lg shadow-lg border border-gray-700 mt
    <h1 class="text-2xl font-bold mb-6 text-center text-[#12c2e9]">Usuaris</h1>

    <!-- Formulario de búsqueda -->
    <form method="GET" action="{{ route('users.index') }}" class="mb-6 flex items-center space-x-2">
        <input type="text" name="search" placeholder="Cercar usuaris..." value="{{ request('search') }
               class="bg-[#12121a] border border-gray-600 text-black p-2 w-full rounded f
               data-qa="search-input">
        <button type="submit" class="bg-[#12c2e9] hover:bg-[#0b93d5] text-yellow-300  for
                data-qa="search-button">Cercar</button>
    </form>

    <!-- Tabla de usuarios -->
    <div class="overflow-x-auto">
        <table class="w-full bg-[#12121a] border border-gray-700 rounded-lg overflow-hidden t
            <thead class="bg-[#1a1a2e] text-[#12c2e9]">
            <tr>
                <th class="px-4 py-3">Nom</th>
                <th class="px-4 py-3">Email</th>
                <th class="px-4 py-3">Accions</th>
            </tr>
            </thead>
            <tbody>
            @foreach($users as $user)
                <tr class="border-b border-gray-700 hover:bg-[#1e1e2e] transition">
                    <td class="px-4 py-3 text-[#e6e6fa]">{{ $user->name }}</td>
                    <td class="px-4 py-3 text-[#e6e6fa]">{{ $user->email }}</td>
                    <td class="px-4 py-3">
                        <a href="{{ route('users.show', $user->id) }}" class="bg-[#c471ed] ho
                           data-qa="user-details-link">Informació Usuari</a>
                    </td>
                </tr>
```

- A helpers crear els permisos de gestió dels usuaris per al crud i assignar-los als usuaris superadmin.



```
}

1 usage  «Marc»
public static function create_user_management_permission()
{
    $permissionName = 'admmistradorUsuaris';

    if (!Permission::where('name', $permissionName)->exists()) {
        Permission::firstOrCreate(['name' => $permissionName]);
    }

    $superAdminRole = Role::firstOrCreate(['name' => 'super-admin', 'guard_name' => 'web']);
    if (!$superAdminRole->hasPermissionTo($permissionName)) {
        $superAdminRole->givePermissionTo($permissionName);
    }
}
```

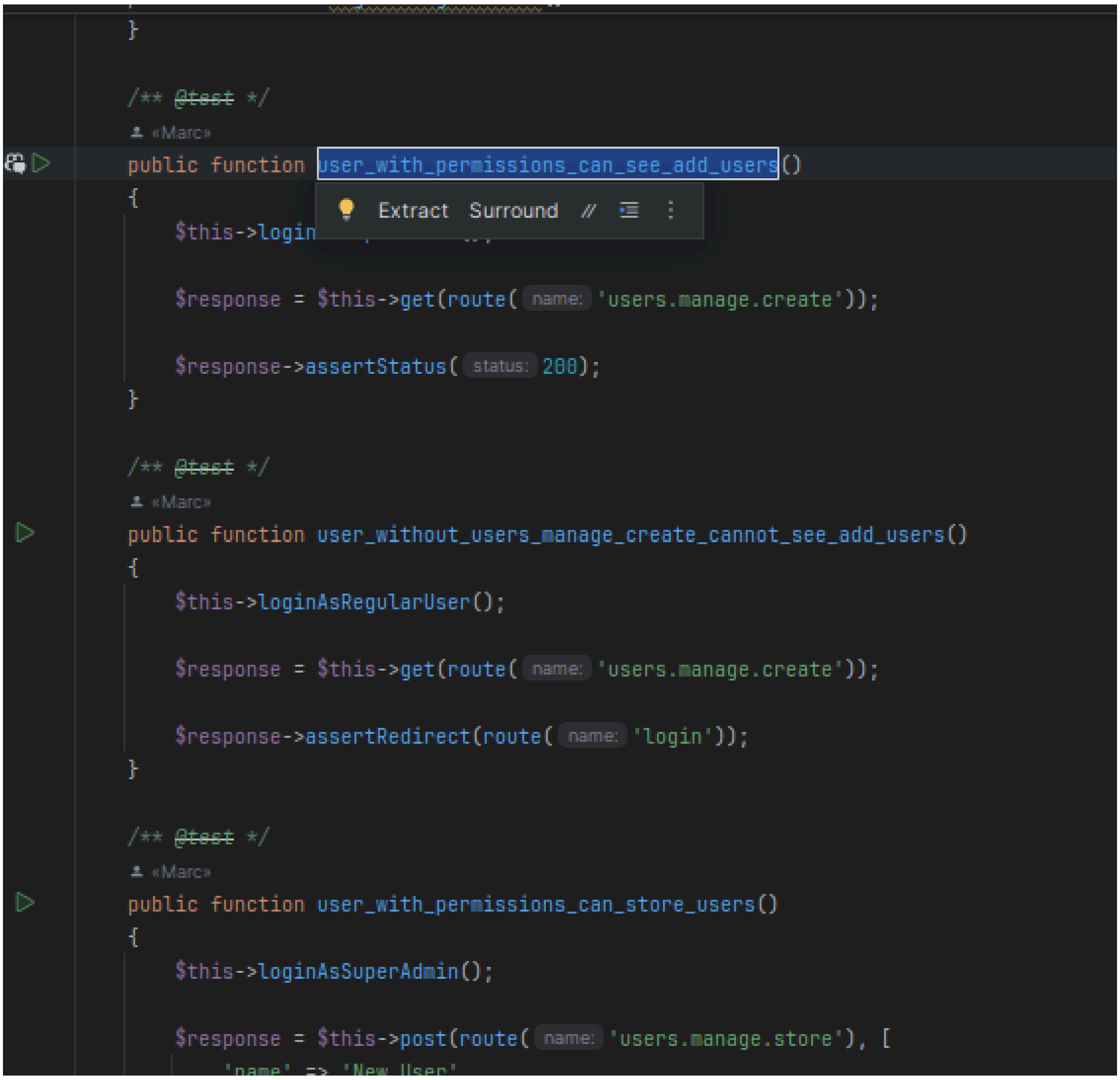
- A UserTest crear les funcions:
  user_without_permissions_can_see_default_users_page,
  user_with_permissions_can_see_default_users_page,
  not_logged_users_cannot_see_default_users_page,
  user_without_permissions_can_see_user_show_page,
  user_with_permissions_can_see_user_show_page,
  not_logged_users_cannot_see_user_show_page



```
/** @test */
«Marc»
public function user_without_permissions_can_see_default_users_page()
{
    $user = User::factory()->create();

    $response = $this->actingAs($user)->get(route( name: 'users.index'));

    $response->assertStatus( status: 200);
}

/** @test */
«Marc»
public function user_with_permissions_can_see_default_users_page()
{
    $user = User::factory()->create();
    $user->givePermissionTo( ...permissions: 'manageUsers');

    $response = $this->actingAs($user)->get(route( name: 'users.index'));

    $response->assertStatus( status: 200);
}


/** @test */
«Marc»
public function not_logged_users_cannot_see_default_users_page()
{
    $response = $this->get(route( name: 'users.index'));

    $response->assertRedirect(route( name: 'login'));
```

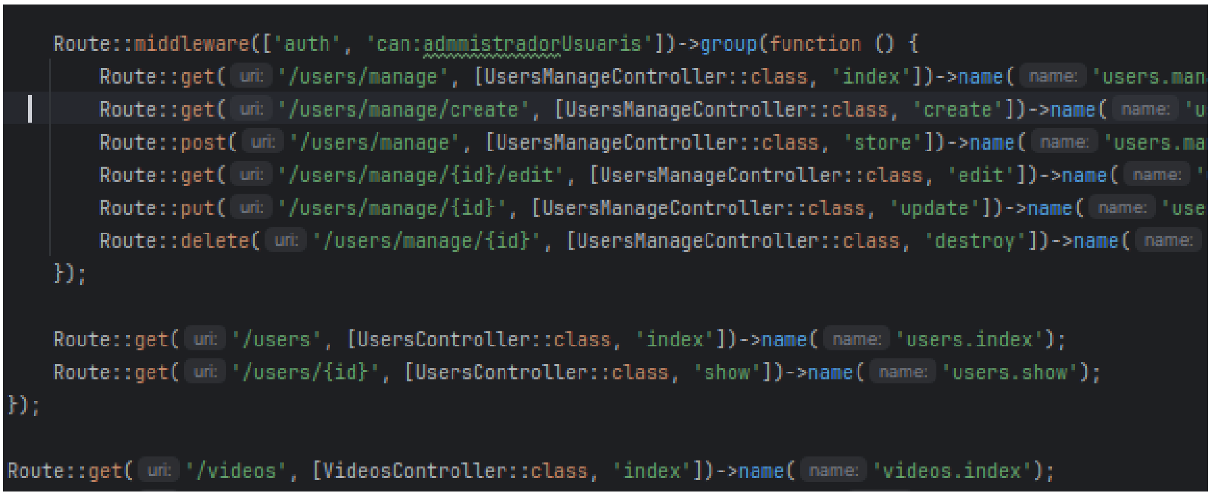
- A UsersManageControllerTest crear les funcions:
  loginAsVideoManager,
  loginAsSuperAdmin,
  loginAsRegularUser,
  user_with_permissions_can_see_add_users,
  user_without_users_manage_create_cannot_see_add_users,
  user_with_permissions_can_store_users,
  user_without_permissions_cannot_store_users,
  user_with_permissions_can_destroy_users,
  user_without_permissions_cannot_destroy_users,
  user_with_permissions_can_see_edit_users,
  user_without_permissions_cannot_see_edit_users,
  user_with_permissions_can_update_users,
  user_without_permissions_cannot_update_users

user_with_permissions_can_manage_users,
regular_users_cannot_manage_users,
guest_users_cannot_manage_users,
superadmins_can_manage_users



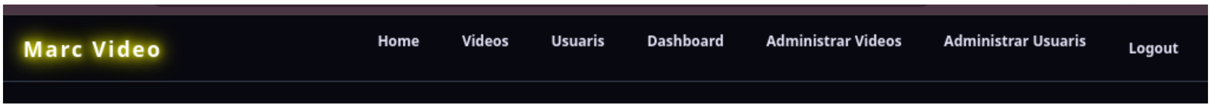
```
    }

    /** @test */
    «Marc»
    public function user_with_permissions_can_see_add_users()
    {
        $this->login

        $response = $this->get(route( name: 'users.manage.create'));

        $response->assertStatus( status: 200);
    }

    /** @test */
    «Marc»
    public function user_without_users_manage_create_cannot_see_add_users()
    {
        $this->loginAsRegularUser();

        $response = $this->get(route( name: 'users.manage.create'));

        $response->assertRedirect(route( name: 'login'));
    }

    /** @test */
    «Marc»
    public function user_with_permissions_can_store_users()
    {
        $this->loginAsSuperAdmin();

        $response = $this->post(route( name: 'users.manage.store'), [
            'name' => 'New User'
```

- Crear les rutes de users/manage per al CRUD d'usuaris amb el seu middleware correspondent i la ruta de l'índex i el show d'usuaris. Les rutes del CRUD i les de l'índex i show han d'aparèixer només quan estàs logejat.

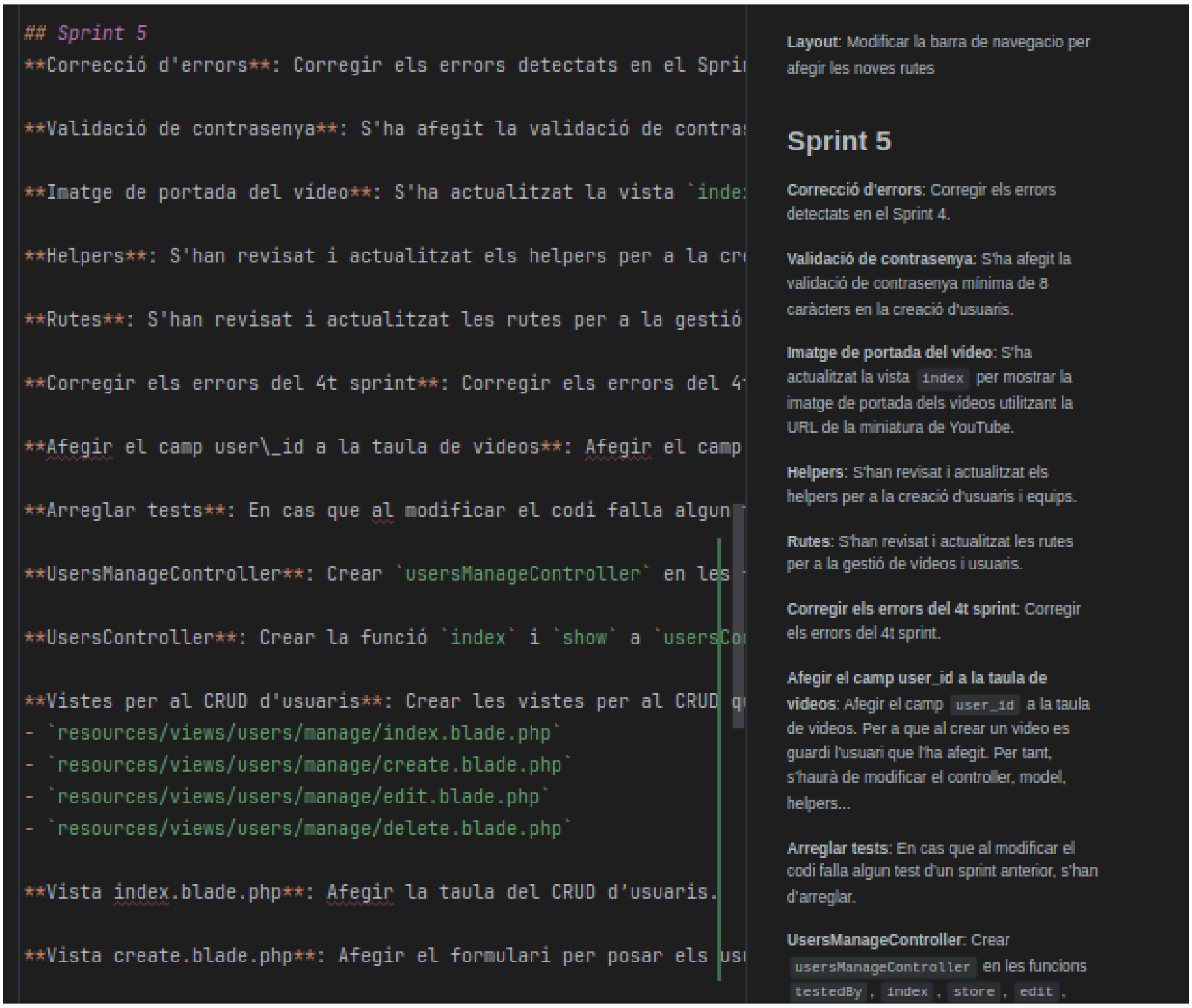


```
    Route::middleware(['auth', 'can:admmistradorUsuaris'])->group(function () {
        Route::get( uri: '/users/manage', [UsersManageController::class, 'index'])->name( name: 'users.man
        Route::get( uri: '/users/manage/create', [UsersManageController::class, 'create'])->name( name: 'u
        Route::post( uri: '/users/manage', [UsersManageController::class, 'store'])->name( name: 'users.ma
        Route::get( uri: '/users/manage/{id}/edit', [UsersManageController::class, 'edit'])->name( name: '
        Route::put( uri: '/users/manage/{id}', [UsersManageController::class, 'update'])->name( name: 'use
        Route::delete( uri: '/users/manage/{id}', [UsersManageController::class, 'destroy'])->name( name:
    });

    Route::get( uri: '/users', [UsersController::class, 'index'])->name( name: 'users.index');
    Route::get( uri: '/users/{id}', [UsersController::class, 'show'])->name( name: 'users.show');
});

Route::get( uri: '/videos', [VideosController::class, 'index'])->name( name: 'videos.index');
```

- S'ha de poder navegar entre pàgines.



- Afegir a resources/markdown/terms el que heu fet al sprint.



```
## Sprint 5
**Correcció d'errors**: Corregir els errors detectats en el Spri

**Validació de contrasenya**: S'ha afegit la validació de contra

**Imatge de portada del video**: S'ha actualitzat la vista `inde

**Helpers**: S'han revisat i actualitzat els helpers per a la cr

**Rutes**: S'han revisat i actualitzat les rutes per a la gestió

**Corregir els errors del 4t sprint**: Corregir els errors del 4

**Afegir el camp user\_id a la taula de videos**: Afegir el camp

**Arreglar tests**: En cas que al modificar el codi falla algun

**UsersManageController**: Crear `usersManageController` en les

**UsersController**: Crear la funció `index` i `show` a `usersCo

**Vistes per al CRUD d'usuaris**: Crear les vistes per al CRUD q
- `resources/views/users/manage/index.blade.php`
- `resources/views/users/manage/create.blade.php`
- `resources/views/users/manage/edit.blade.php`
- `resources/views/users/manage/delete.blade.php`

**Vista index.blade.php**: Afegir la taula del CRUD d'usuaris.

**Vista create.blade.php**: Afegir el formulari per posar els us
```

Layout: Modificar la barra de navegacio per afegir les noves rutes

Sprint 5

Correcció d'errors: Corregir els errors detectats en el Sprint 4.

Validació de contrasenya: S'ha afegit la validació de contrasenya mínima de 8 caràcters en la creació d'usuaris.

Imatge de portada del video: S'ha actualitzat la vista `index` per mostrar la imatge de portada dels videos utilitzant la URL de la miniatura de YouTube.

Helpers: S'han revisat i actualitzat els helpers per a la creació d'usuaris i equips.

Rutes: S'han revisat i actualitzat les rutes per a la gestió de videos i usuaris.

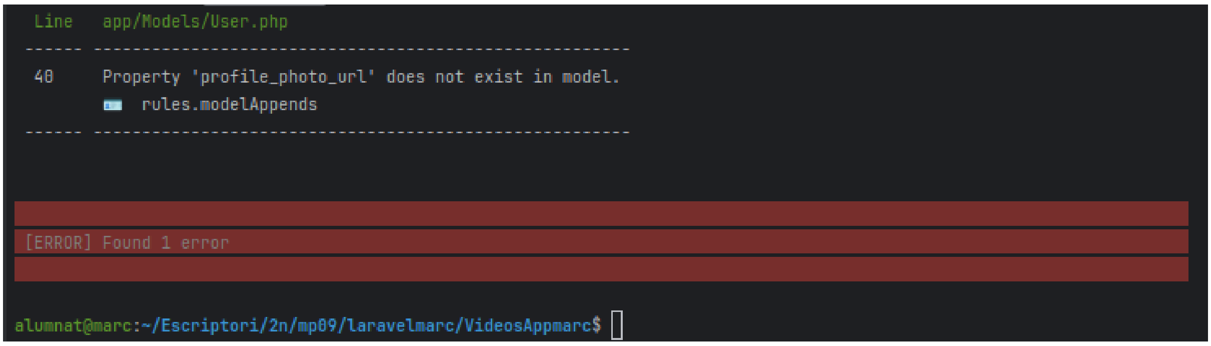
Corregir els errors del 4t sprint: Corregir els errors del 4t sprint.

Afegir el camp user_id a la taula de videos: Afegir el camp `user_id` a la taula de videos. Per a que al crear un video es guardi l'usuari que l'ha afegit. Per tant, s'haurà de modificar el controller, model, helpers...

Arreglar tests: En cas que al modificar el codi falla algun test d'un sprint anterior, s'han d'arreglar.

UsersManageController: Crear `usersManageController` en les funcions `testedBy`, `index`, `store`, `edit`,

- Comprovar en Larastan tots els fitxers que heu creat.



```
 Line   app/Models/User.php
 ------ -------------------------------------------------------
  40     Property 'profile_photo_url' does not exist in model.
         🪪  rules.modelAppends
 ------ -------------------------------------------------------


 [ERROR] Found 1 error

alumnat@marc:~/Escriptori/2n/mp09/laravelmarc/VideosAppmarc$
```

enllaç github

https://github.com/Mpallares1/laravelmarc