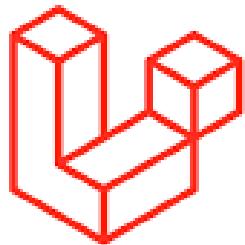


Projecte GLOBAL: VideosApp

7e SPRINT



Professor: Jordi Vega
Assignatures: M7, M8 i M9
Nom: Harvey
Cognoms: John Glover



INDEX:

Corregir els errors del 6e sprint.	3
Arreglar test dels sprints anteriors	7
Usuaris regulars creen series	8
Video Created.php	12
SendVideoCreatedNotification	13
VideoCreatedNotification.php	15
EventServiceProvider.php	16
VideosManageController.php	17
Registrar-se a Mailtrap i pusher	18
Configurar el .env	18
broadcasting.php	19
bootstrap.js	20
Rutes i vistes	21
Comprovació del app	23
videoNotificationsTest.php	28
Cicle TDD essencial	29
Terms.md	30
Comprovar errors Larastan	31

Corregir els errors del 6e sprint.

Retroacció

Qualificació	10,00 / 10,00
Qualificat el	dijous, 17 d'abril 2025, 13:20
Qualificat per	JV Jordi Vega Tomàs [PROF]

Molt bé molt bon treball. Estaria bé posar els botons d'editar i eliminar al show dels videos de l'usuari. Això per als regular user.

Canvis:

Ajusto les funcions de update i delete per redireccionar correctament.

```
$video->update($data);

// Decide redirect: admins back to manage, regular back to public show
if (auth()->user()->can(abilities: 'manage-videos')) {
    $redirect = $request->query(key: 'redirect', route(name: 'videos.manage.index'));
} else {
    $redirect = $request->query(key: 'redirect', route(name: 'video.show', $video->id));
}
```

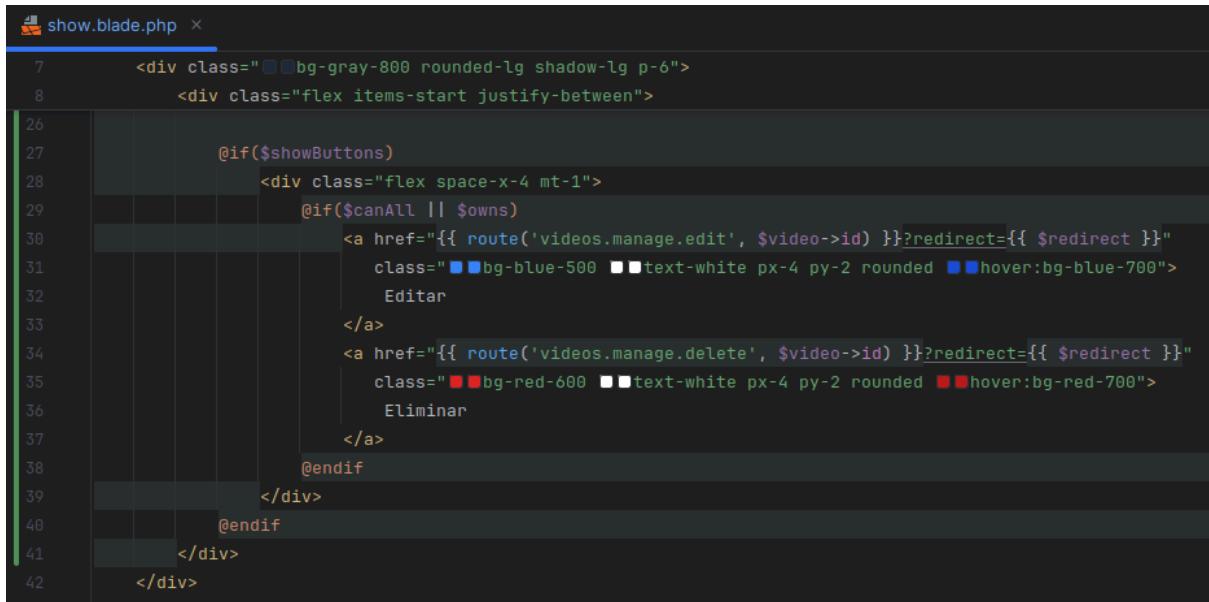
```
public function delete($id): View|Factory|Application
{
    $video = Video::findOrFail($id);
    abort_unless(
        boolean: auth()->user()->can(abilities: 'manage-videos') || $video->user_id === auth()->id(),
        code: 403
    );
    return view(view: 'videos.manage.delete', compact(var_name: 'video'));
}
```

A les vistes de edit i delete procuro saber d'on ve i qui l'usuari és per accedir a la vista i redireccionar una vegada completada l'acció.

```
edit.blade.php ×  
1 @extends('layouts.VideosAppLayout')  
2  
3 @section('content')  
4 @php  
5     // Capture where we came from  
6     $redirect = request()->query('redirect');  
7     // Decide cancel/back URL  
8     $cancelUrl = $redirect  
9     ?? (auth()->user()->can('manage-videos'))  
10        ? route('videos.manage.index')  
11        : route('video.show', $video->id));  
12 @endphp
```

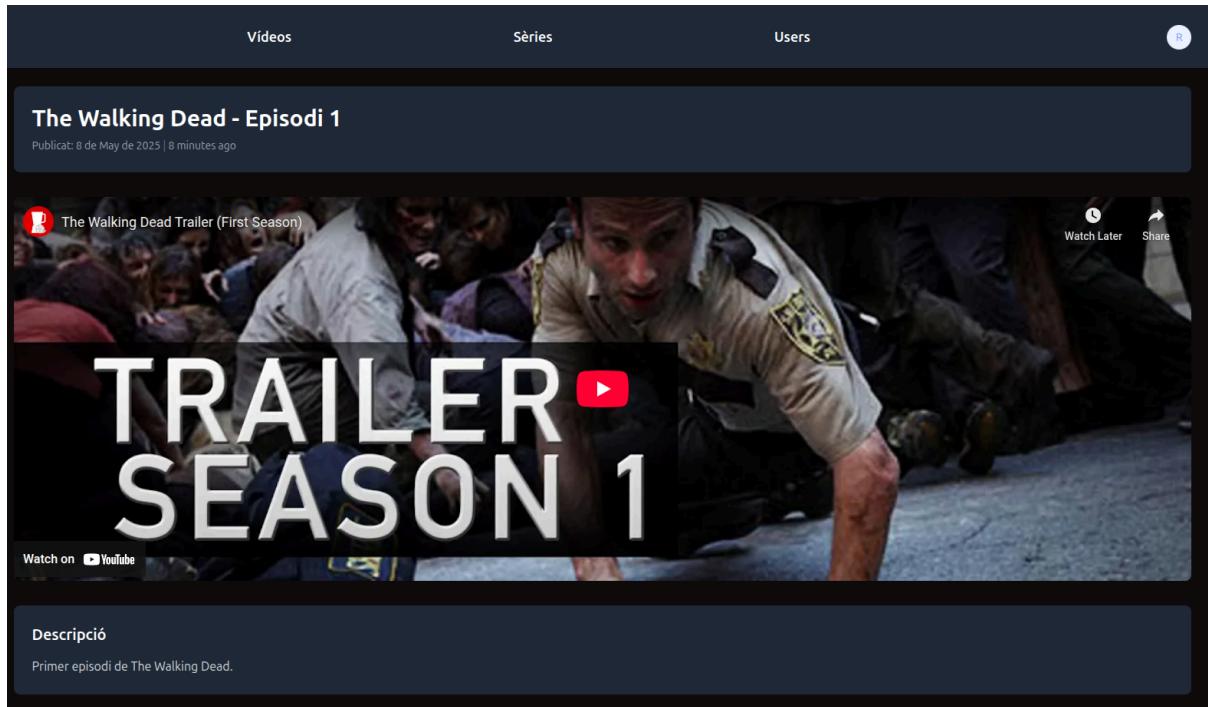
```
delete.blade.php ×  
1 @extends('layouts.VideosAppLayout')  
2  
3 @section('content')  
4 @php  
5     $redirect = request()->query('redirect');  
6     // If cancelling, go back to public show or manage index:  
7     $cancelUrl = $redirect  
8     ?? (auth()->user()->can('manage-videos'))  
9         ? route('videos.manage.index')  
10         : route('video.show', $video->id));  
11 @endphp
```

Al show on és mostren els botons, procurarem sol mostrar als usuaris admin o que tinguin permisos sempre, i per als regular, solament els seus vídeos.

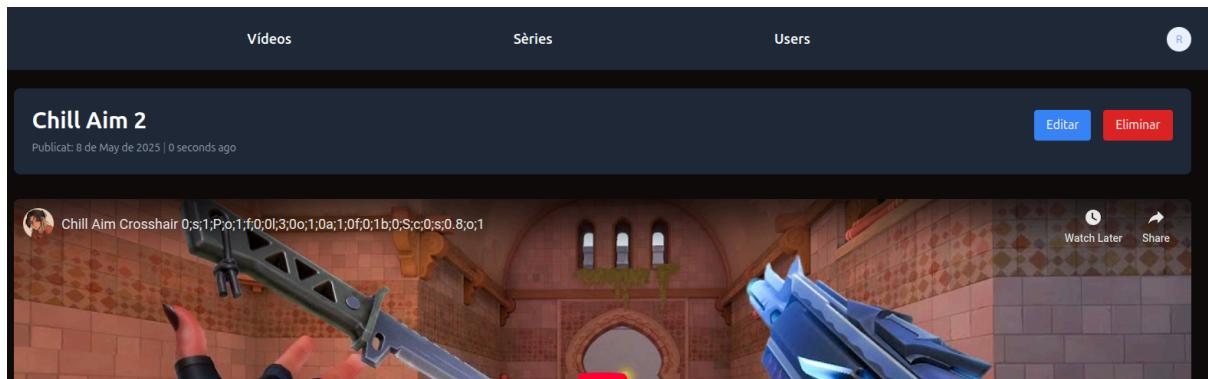


```
show.blade.php
7     <div class="bg-gray-800 rounded-lg shadow-lg p-6">
8         <div class="flex items-start justify-between">
26             @if($showButtons)
27                 <div class="flex space-x-4 mt-1">
28                     @if($canAll || $owns)
29                         <a href="{{ route('videos.manage.edit', $video->id) }}?redirect={{ $redirect }}"
30                             class="bg-blue-500 text-white px-4 py-2 rounded hover:bg-blue-700">
31                             Editar
32                         </a>
33                         <a href="{{ route('videos.manage.delete', $video->id) }}?redirect={{ $redirect }}"
34                             class="bg-red-600 text-white px-4 py-2 rounded hover:bg-red-700">
35                             Eliminar
36                         </a>
37                     @endif
38                 </div>
39             @endif
40         </div>
41     </div>
42 
```

Amb el regular, no veurem el botó a tots els vídeos.



Una vegada creem un vídeo amb l'usuari, ens apareixen els botons al seu vídeo,



Ens permet editar i eliminar.

Arreglar test dels sprints anteriors

Elimino els test de usuari regular no pot guardar vídeos.

Ja tindre les rutes arreglades.

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan test --filter=SeriesManageControllerTest

  PASS  Tests\Feature\Series\SeriesManageControllerTest
    ✓ user with permissions can see add series
    ✓ user with permissions can store series
    ✓ user with permissions can destroy series
    ✓ user without permissions cannot destroy series
    ✓ user with permissions can see edit series
    ✓ user without permissions cannot see edit series
    ✓ user with permissions can update series
    ✓ user without permissions cannot update series
    ✓ user with permissions can manage series
    ✓ regular users cannot manage series
    ✓ guest users cannot manage series
    ✓ videomanagers can manage series
    ✓ superadmins can manage series

  Tests:    13 passed (26 assertions)
  Duration: 0.51s
```

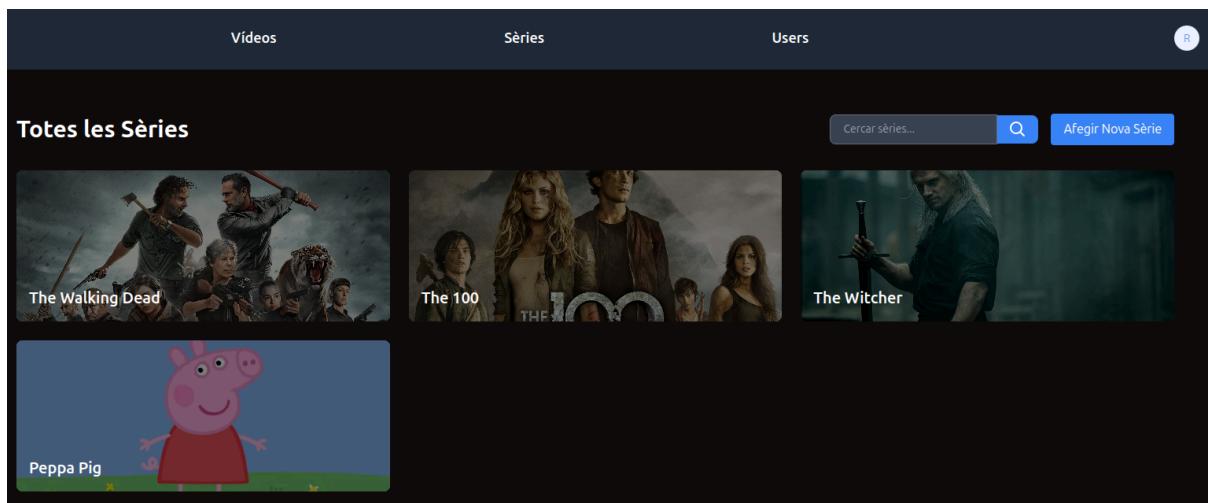
Usuaris regulars creen series

Afegir si no s'ha fet al sprint anterior que els regulars users puguin crear series i afegir videos a la serie.

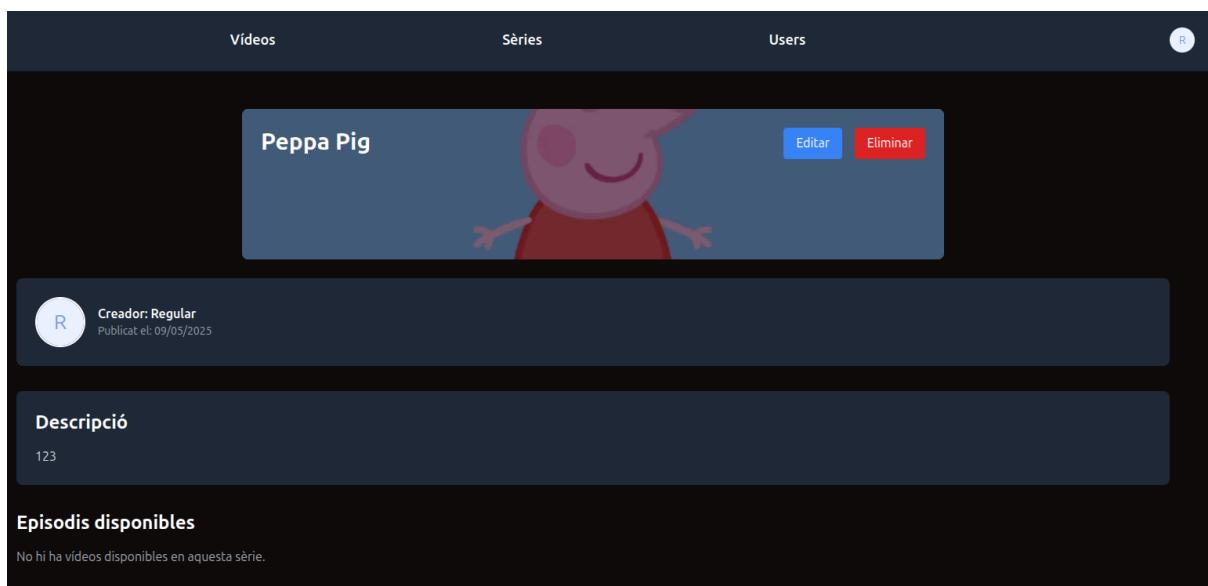
```
delete.blade.php ×  
1 @extends('layouts.VideosAppLayout')  
2  
3 @section('content')  
4     @php  
5         $redirect = request('redirect');  
6         $cancelUrl = $redirect  
7             ?? (auth()->user()->can('manage-series'))  
8                 ? route('series.manage.index')  
9                 : route('series.show', $serie->id));  
10    @endphp
```

```
edit.blade.php ×  
1 @extends('layouts.VideosAppLayout')  
2  
3 @section('content')  
4     @php  
5         $redirect = request('redirect');  
6         $cancelUrl = $redirect  
7             ?? (auth()->user()->can('manage-series'))  
8                 ? route('series.manage.index')  
9                 : route('series.show', $serie->id));  
10    @endphp
```

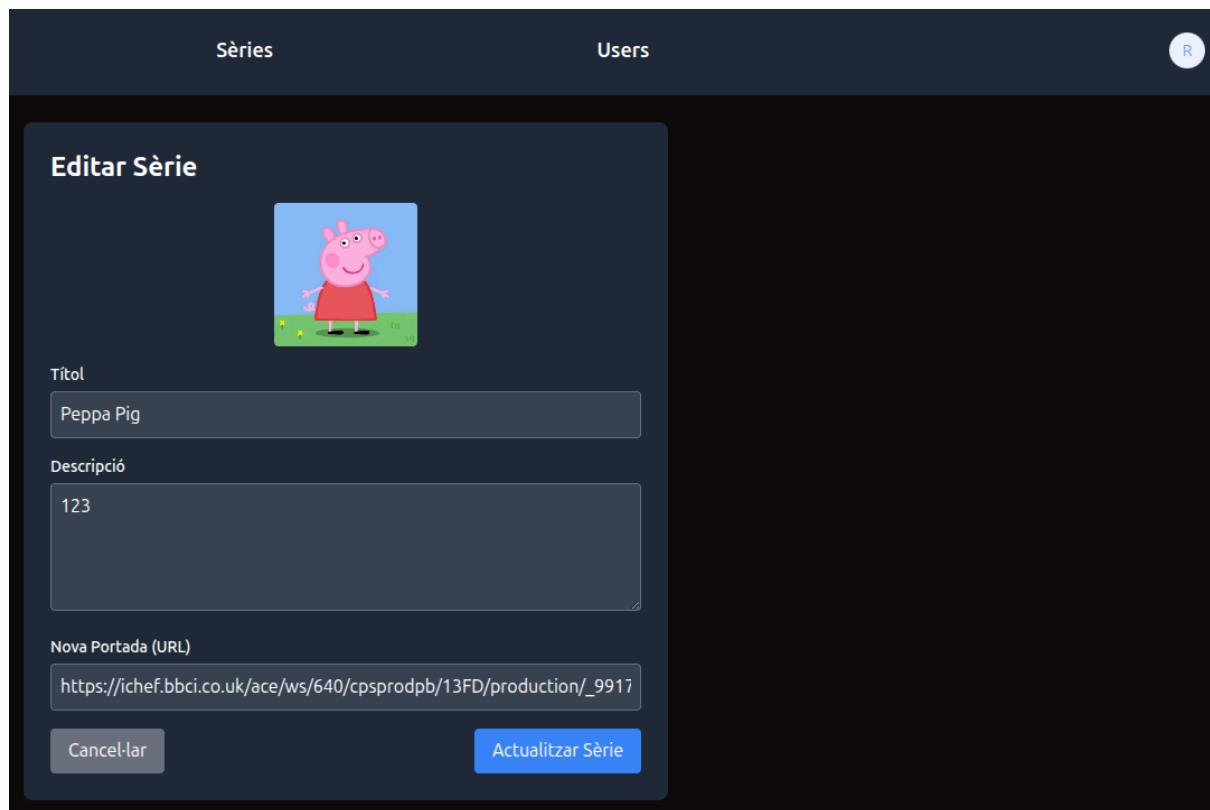
Entro amb l'usuari regular, apareix el botó d'afegir nova sèrie i afegire una.



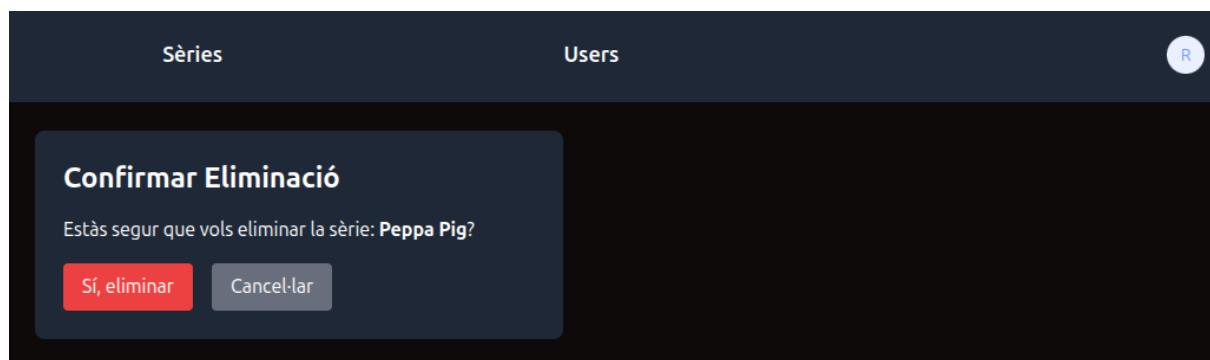
Si clico dins de la sèrie creada no té episodis de moment, però apareix el botó de editar i eliminar ja que és del propi usuari.



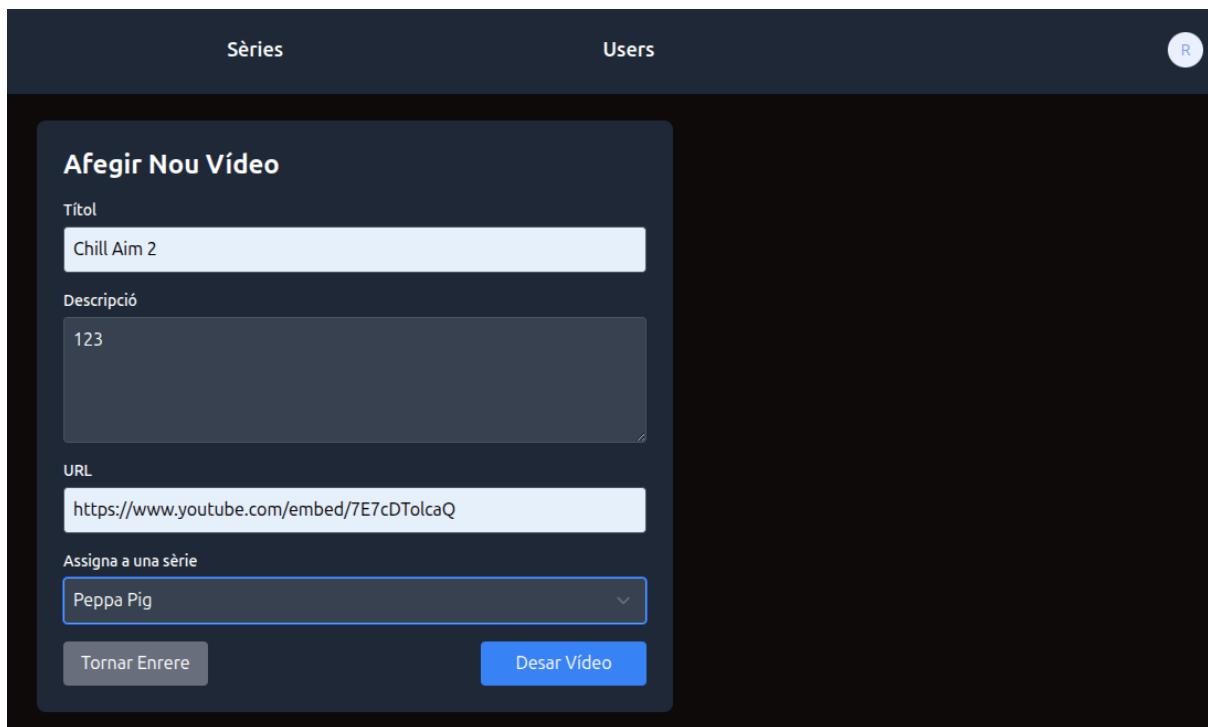
Em permet editar.



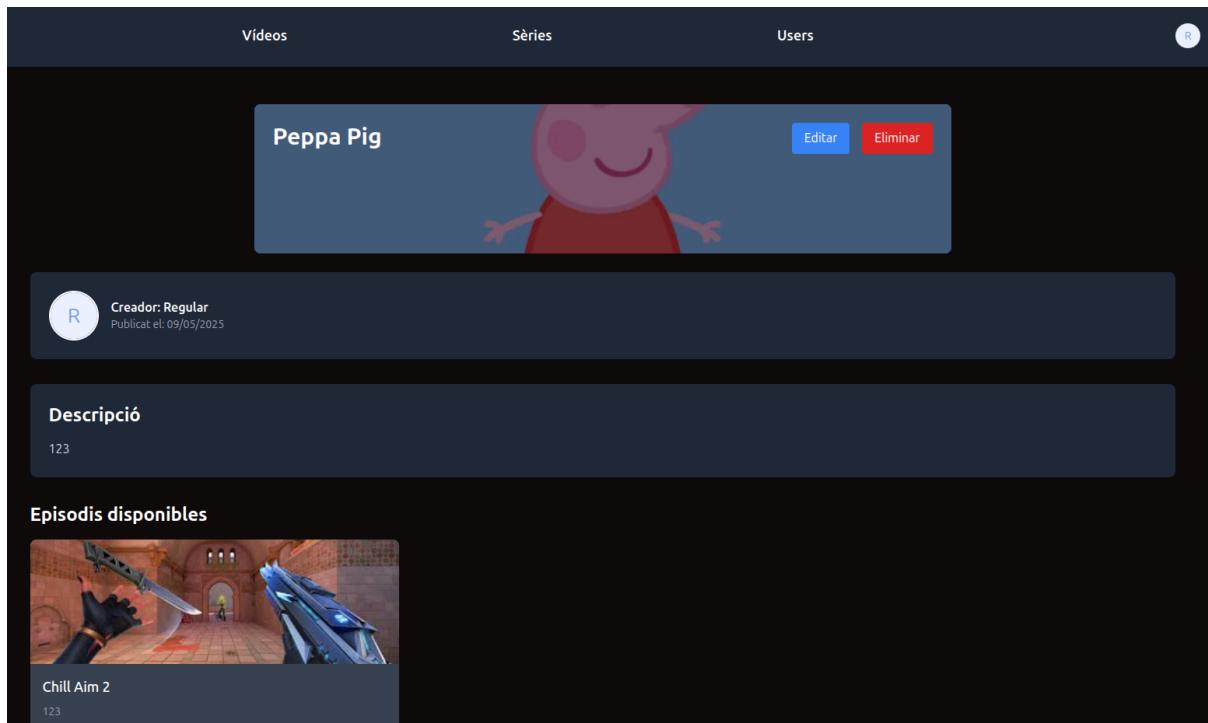
I també em permet eliminar.



Si afegeix un vídeo a la sèrie.



M'apareix el vídeo que està creat i afegit per el mateix usuari regular.

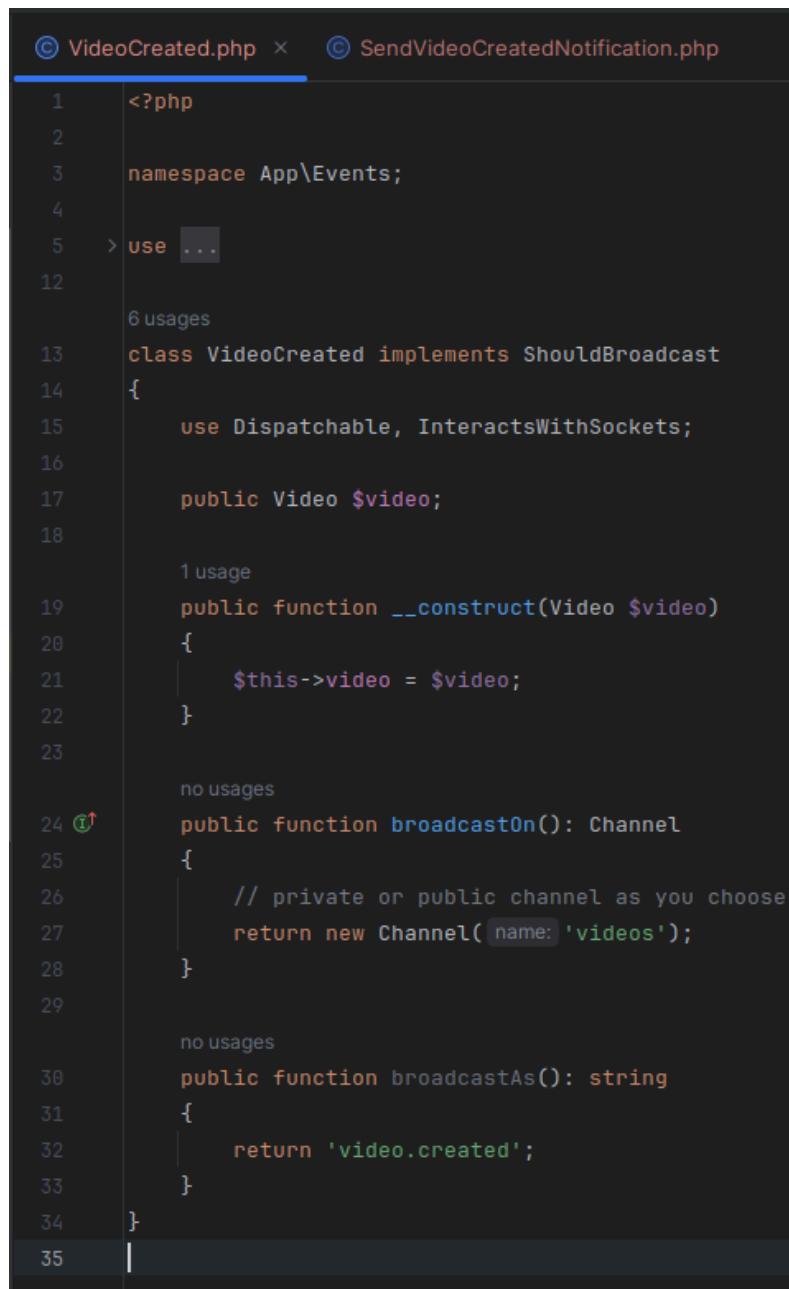


Video Created.php

Creo el document, i dins creo el broadcast on i as per a que quan és crei el vídeo active el broadcast per enviar la notificació i mail.

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan make:event VideoCreated
INFO Event [app/Events/VideoCreated.php] created successfully.

alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$
```



The screenshot shows a code editor with two tabs open. The left tab is titled 'VideoCreated.php' and contains the following PHP code:

```
1 <?php
2
3 namespace App\Events;
4
5 > use ...
12
6 usages
13 class VideoCreated implements ShouldBroadcast
14 {
15     use Dispatchable, InteractsWithSockets;
16
17     public Video $video;
18
19     1 usage
20     public function __construct(Video $video)
21     {
22         $this->video = $video;
23     }
24
25     no usages
26     public function broadcastOn(): Channel
27     {
28         // private or public channel as you choose
29         return new Channel(name: 'videos');
30     }
31
32     no usages
33     public function broadcastAs(): string
34     {
35         return 'video.created';
36     }
37 }
```

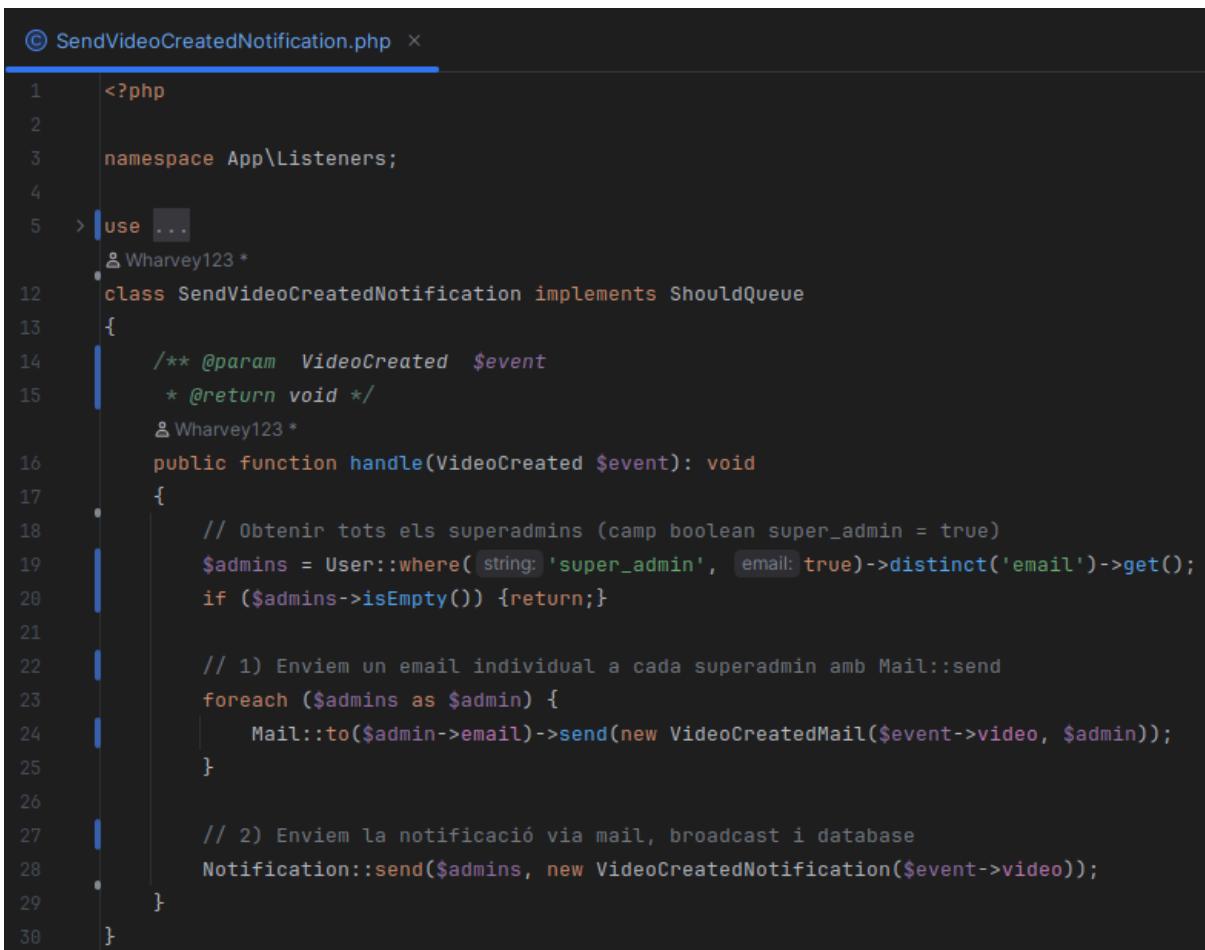
The right tab is titled 'SendVideoCreatedNotification.php' and is currently not visible.

SendVideoCreatedNotification

Al listener de la notificació SendVideoCreatedNotification i creo la funció handle(VideoCreated \$event) que envia un correu als admins i s'envia la notificació VideoCreated i se li passa informació del video relacionada en l'event.

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan make:listener SendVideoCreatedNotification --event=VideoCreated
INFO Listener [app/Listeners/SendVideoCreatedNotification.php] created successfully.

alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$
```



The screenshot shows a code editor window with the file `SendVideoCreatedNotification.php` open. The code is a PHP class named `SendVideoCreatedNotification` that implements the `ShouldQueue` interface. It has a single method `handle` that sends emails to superadmins and sends notifications via mail, broadcast, and database. The code uses Laravel's Eloquent ORM and Mail facade.

```
⑥ SendVideoCreatedNotification.php ×

1 <?php
2
3 namespace App\Listeners;
4
5 > use ... Wharvey123 *
6
7 class SendVideoCreatedNotification implements ShouldQueue
8 {
9
10     /** @param VideoCreated $event
11      * @return void */
12     & Wharvey123 *
13
14     public function handle(VideoCreated $event): void
15     {
16
17         // Obtenir tots els superadmins (camp boolean super_admin = true)
18         $admins = User::where('super_admin', true)->distinct('email')->get();
19         if ($admins->isEmpty()) {return;}
20
21
22         // 1) Enviem un email individual a cada superadmin amb Mail::send
23         foreach ($admins as $admin) {
24             Mail::to($admin->email)->send(new VideoCreatedMail($event->video, $admin));
25         }
26
27         // 2) Enviem la notificació via mail, broadcast i database
28         Notification::send($admins, new VideoCreatedNotification($event->video));
29     }
30 }
```

Creo el videocreatedmail, i posaré un build que li posa subject al correu i el construeix el model.

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan make:mail VideoCreatedMail
INFO  Mailable [app/Mail/VideoCreatedMail.php] created successfully.

alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$
```

```
© VideoCreatedMail.php ×
1  <?php
2
3  namespace App\Mail;
4
5  use App\Models\Video;
6  use Illuminate\Bus\Queueable;
7  use Illuminate\Mail\Mailable;
8  use Illuminate\Queue\SerializesModels;
9
10 class VideoCreatedMail extends Mailable
11 {
12     use Queueable, SerializesModels;
13
14     public Video $video;
15     public $user;
16
17     1 usage
18     public function __construct(Video $video, $user)
19     {
20         $this->video = $video;
21         $this->user = $user;
22     }
23
24     public function build()
25     {
26         return $this
27             ->subject('Nou video creat: ' . $this->video->title)
28             ->view('emails.video_created')
29             ->with([
30                 'video' => $this->video,
31                 'user'  => $this->user,
32             ]);
33     }
34 }
```

VideoCreatedNotification.php

Creo el videocreatednotification per mostrar el que enviar al correu i fer el procés.

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan make:notification VideoCreatedNotification
INFO  Notification [app/Notifications/VideoCreatedNotification.php] created successfully.

alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$
```

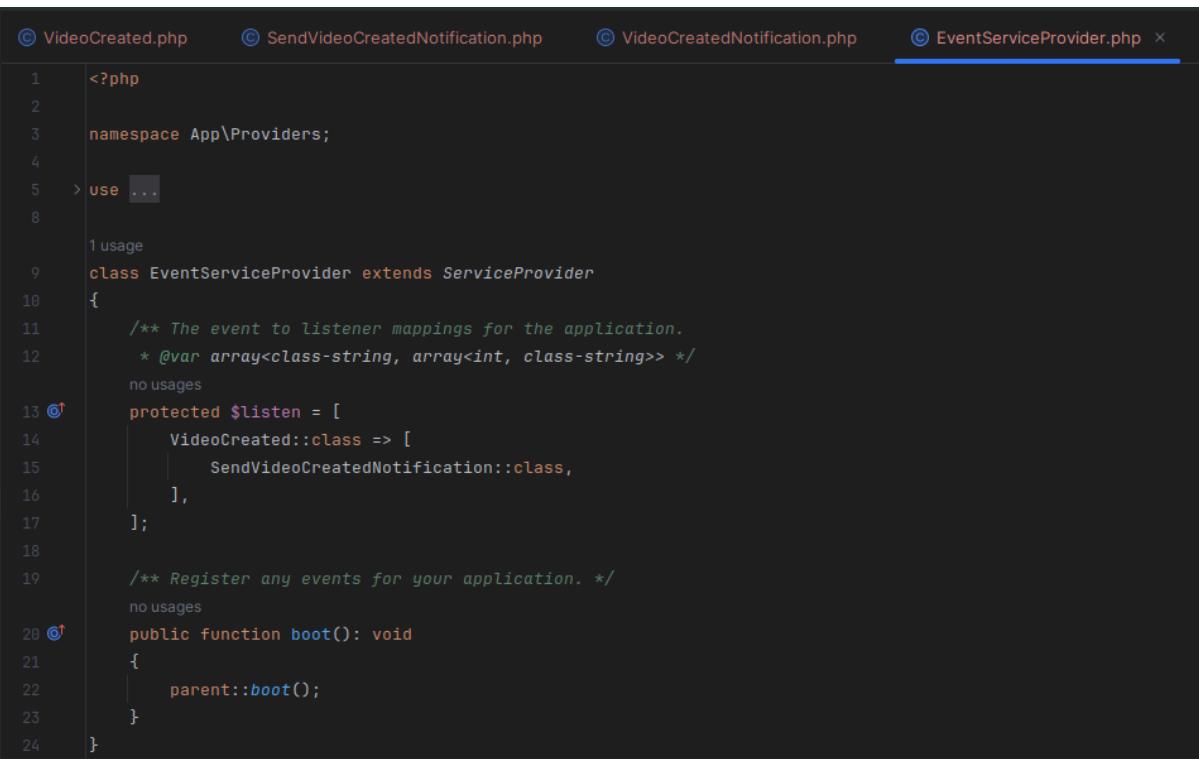
```
© VideoCreatedNotification.php ×
1  <?php
2
3  namespace App\Notifications;
4  use App\Models\Video;
5  use Illuminate\Bus\Queueable;
6  use Illuminate\Contracts\Queue\ShouldQueue;
7  use Illuminate\Notifications\Notification;
8  use Illuminate\Notifications\Messages\BroadcastMessage;
9
10
11
12  use Queueable;
13  public Video $video;
14
15
16  public function __construct(Video $video)
17  {
18      $this->video = $video;
19
20
21  public function via($notifiable): array
22  {
23      return ['broadcast', 'database'];
24
25
26  public function toBroadcast($notifiable): BroadcastMessage
27  {
28      return new BroadcastMessage([
29          'video_id'    => $this->video->id,
30          'title'       => $this->video->title,
31          'created_at'  => $this->video->created_at->toIso8601String(),
32      ]);
33
34
35  public function toArray($notifiable): array
36  {
37      return [
38          'video_id'  => $this->video->id,
39          'title'     => $this->video->title,
40          'url'        => url( path: "/video/{$this->video->id}" ),
41      ];
42
43
44
```

EventServiceProvider.php

Creo el event service provider per executar la funció i cridar el boot.

```
aalumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan make:provider EventServiceProvider
INFO Provider [app/Providers/EventServiceProvider.php] created successfully.

aalumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$
```



```
⑤ VideoCreated.php      ⑤ SendVideoCreatedNotification.php      ⑤ VideoCreatedNotification.php      ⑤ EventServiceProvider.php ×
1 <?php
2
3 namespace App\Providers;
4
5 > use ...
6
7 1 usage
8
9 class EventServiceProvider extends ServiceProvider
10 {
11
12     /** The event to listener mappings for the application.
13      * @var array<class-string, array<int, class-string>> */
14     no usages
15
16     protected $listen = [
17         VideoCreated::class => [
18             SendVideoCreatedNotification::class,
19         ],
20     ];
21
22     /** Register any events for your application. */
23     no usages
24     public function boot(): void
25     {
26         parent::boot();
27     }
28 }
```

VideosManageController.php

Modificar la funció de store per llençar el event una vegada creat el vídeo i iniciar el procés d'enviament de correu i notificació.

```
public function store(Request $request): RedirectResponse
{
    abort_unless(auth()->user()->can('videos.create'), 403);

    $data = $request->validate([
        'title'      => 'required|string|max:255',
        'description' => 'required|string',
        'url'        => 'required|url',
        'series_id'   => 'nullable|integer|exists:series,id',
    ]);

    $data['published_at'] = now();
    $data['user_id']      = auth()->id();

    $video = Video::create($data);

    // Llançar l'esdeveniment
    event(new VideoCreated($video));

    // Redirigir a la URL anterior o a la pàgina del vídeo
    $redirect = $request->input('redirect', route('video.show', $video->id));

    return redirect($redirect)
        ->with('success', 'Video creat correctament.');
}
```

No olvidare crear la taula de notifications.

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan notifications:table

INFO Migration created successfully.
```

Registrar-se a Mailtrap i pusher

Em registro al mailtrap i pusher per copiar les credencials necessàries.

Code Samples

curl PHP: Laravel 9+ Node.js Python C# Ruby Other

```

1 # Looking to send emails in production? Check out our Email API/SMTP product!
2 MAIL_MAILER=smtp
3 MAIL_HOST=sandbox.smtp.mailtrap.io
4 MAIL_PORT=2525
5 MAIL_USERNAME=9384682c71d42a
6 MAIL_PASSWORD=****4ebd

```

PUSHER Home > Channels > VideosAppHarvey

App keys

If a token is compromised, you can create a new key/secret pair. You should delete the old token once you've updated your app.

Overview Getting Started | App Keys Functions Stats

Created less than a minute ago	Copy
app_id = "1989557" key = "d25103bceb4bce78b4ee" secret = "b9ac72c33824352b3295" cluster = "eu"	

Configurar el .env

Al env posaré la informació que he obtingut prèviament.

```
# Mailtrap SMTP settings
MAIL_MAILER=smtp
MAIL_HOST=sandbox.smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=9384682c71d42a
MAIL_PASSWORD=bfe1e43eec4ebd
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"
```

```
# Pusher broadcast settings
PUSHER_APP_ID=1989557
PUSHER_APP_KEY=d25103bceb4bce78b4ee
PUSHER_APP_SECRET=b9ac72c33824352b3295
PUSHER_APP_CLUSTER=eu
```

broadcasting.php

Al broadcast he de configurar les claus per cridarles correctament al push.

```
php broadcasting.php ×  
1 <?php  
2  
3 return [  
4     'default' => env( key: 'BROADCAST_DRIVER', default: 'pusher'),  
5  
6     'connections' => [  
7  
8         'pusher' => [  
9             'driver' => 'pusher',  
10            'key'    => env( key: 'PUSHER_APP_KEY'),  
11            'secret'  => env( key: 'PUSHER_APP_SECRET'),  
12            'app_id'   => env( key: 'PUSHER_APP_ID'),  
13            'options'  => [  
14                'cluster'  => env( key: 'PUSHER_APP_CLUSTER'),  
15                'useTLS'    => true,  
16                // 'encrypted' => true, // legacy option  
17            ],  
18        ],  
19  
20        'redis' => [  
21            'driver'      => 'redis',  
22            'connection'  => env( key: 'BROADCAST_REDIS_CONNECTION', default: 'default'),  
23        ],  
24  
25        'log' => ['driver' => 'log',],  
26        'null' => ['driver' => 'null',],  
27    ],  
28];|
```

Refresco amb clear.

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan config:clear  
  
INFO Configuration cache cleared successfully.  
  
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan cache:clear  
  
INFO Application cache cleared successfully.  
  
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$
```

bootstrap.js

Instalo laravel-echo pusher-js en npm.

```
aIumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ npm install --save laravel-echo pusher-js
added 12 packages, and audited 234 packages in 2s

48 packages are looking for funding
  run `npm fund` for details
```

Després, configuro el bootstrap per importarlo.

```
JS bootstrap.js ×
1 import Echo from 'laravel-echo';
2 window.Pusher = require('pusher-js');
3
4 window.Echo = new Echo({ options: {
5   broadcaster: 'pusher',
6   key: import.meta.env.VITE_PUSHER_APP_KEY,
7   cluster: import.meta.env.VITE_PUSHER_APP_CLUSTER,
8   forceTLS: true
9 }});
```

Rutes i vistes

Creo un document anomenat **video_created.blade.php** que és mostrarà la info correctament al correu per enviarlo d'una forma més adequada.

```
<!DOCTYPE html>
<html lang="ca">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <title>Nou Video Creat</title>
</head>
<body style="...">
<div style="...">
    <div style="...">
        <h1 style="...">Nou Video Creat</h1>
        @isset($user)
            <p>Hola {{ $user->name }},</p>
        @endisset
        <p>S'ha publicat un nou video amb el títol:</p>
        <p style="...">{{ $video->title }}</p>
        <p>Si vols veure'n el detall o revisar-lo, fes clic al botó següent:</p>
        <p style="...">
            <a href="{{ url('/video/' . $video->id) }}" style="...">
                Veure Video
            </a>
        </p>
        <p>Gràcies per formar part de la nostra comunitat!</p>
    </div>
    <div style="...">
        <p>Si tens cap dubte, contacta amb nosaltres a <a href="mailto:support@example.com" style="...">support@example.com</a>.</p>
    </div>
</div>
</body>
</html>
```

Crear la ruta de notificacions en **web.php**.

```
<?php
use ...
// Ruta per a la pàgina d'inici (redirigeix a vídeos)
Route::get( uri: '/', fn() => redirect()->route( route: 'videos.index') );
// Mostrar notificacions al panel d'usuari
Route::middleware(['auth'])->get( uri: '/notifications', function () {
    return view( view: 'notifications.index');
})->name( name: 'notifications.index');
```

També creare una carpeta anomenada **notifications** i dintre un índex amb la vista de les notificacions

```

index.blade.php x
1 @extends('Layouts.VideosAppLayout')
2
3 @section('content')
4     <div class="max-w-md mx-auto p-4 bg-gray-900 text-gray-100 min-h-screen">
5         <h2 class="text-2xl font-semibold mb-4 text-white text-center">Les teves notificacions</h2>
6
7         @if (Auth::user()->notifications->isEmpty())
8             <p class="text-gray-400 text-center">No tens notificacions pendents.</p>
9         @else
10            <ul id="notification-list" class="space-y-3">
11                @foreach (Auth::user()->notifications()->latest()->get() as $notification)
12                    <li class="p-3 rounded-md {{ $notification->read_at ? 'bg-gray-800' : 'bg-gray-700' }} transition duration-200 text-gray-200" data-id="{{ $notification->id }}">
13                        <div class="flex justify-between items-center">
14                            <div>
15                                <strong class="text-base text-white">Nou video:</strong>
16                                <span class="block text-sm">{{ $notification->data['title'] ?? '-' }}</span>
17                                <small class="text-xs text-gray-400">{{ $notification->created_at->diffForHumans() }}</small>
18                            </div>
19                            <a href="{{ route('video.show', ['id' => $notification->data['video_id']]) }}" class="px-3 py-1 bg-blue-600 text-white text-xs rounded-md hover:bg-blue-700 transition duration-200" style="color: inherit; text-decoration: none;">
20                                Veure video
21                            </a>
22                        </div>
23                    </li>
24                @endforeach
25            </ul>
26        @endif
27    </div>
28
29
30
31
32
33
34
35
36

```

Per clicar el botó de veure creare la seguent funció:

```

<script>
window.Echo.private('App.Models.User.' + {{ auth()->id() }})
.notification((notification) => {
    const list = document.getElementById('notification-list');
    const item = document.createElement('li');

    item.className = 'p-3 bg-gray-700 rounded-md transition duration-200 text-gray-200';
    item.setAttribute('data-id', notification.id);

    item.innerHTML = `
        <div class="flex justify-between items-center">
            <div>
                <strong class="text-base text-white">Nou video:</strong>
                <span class="block text-sm">{{notification.title ?? '-' }}</span>
                <small class="text-xs text-gray-400">ara mateix</small>
            </div>
            <a href="/video/{{$notification.data.video_id}}"
                class="px-3 py-1 bg-blue-600 text-white text-xs rounded-md hover:bg-blue-700 transition duration-200"
            >
                Veure video
            </a>
        </div>
    `;

    list.prepend(item);
});
</script>

```

Al videosaplayout afegire la ruta de notificacions per a que sigui accessible però solament per els admins.



```
 2   <html lang="{{ str_replace('_', '-', app()->getLocale()) }}>
 3   <body class="flex flex-col min-h-screen bg-[#100c6c]">
 4     <nav class="bg-gray-800 p-6 shadow-lg" x-data="{ menuOpen: false }">
 5       <div x-show="menuOpen" @click.away="menuOpen = false">
 6         <div class="fixed inset-y-0 left-0 z-30 w-64 bg-gray-800 p-6">
 7           <ul class="list-none p-0 m-0">
 8             <li>
 9               <a href="#" class="block text-white text-lg font-semibold hover:text-gray-300" Notificaciones</a>
10             </li>
11           </ul>
12         </div>
13       </div>
14     </nav>
15     <div class="flex-grow flex flex-col justify-between">
16       <div class="flex flex-grow flex-wrap gap-4 p-6">
17         <div class="w-1/2">
18           <div class="flex flex-col gap-2" style="background-color: #f0f0f0; padding: 10px; border-radius: 10px">
19             <h3>Vídeos</h3>
20             <ul class="list-none p-0 m-0">
21               <li><a href="#">Video 1</a></li>
22               <li><a href="#">Video 2</a></li>
23             </ul>
24           </div>
25         </div>
26         <div class="w-1/2">
27           <div class="flex flex-col gap-2" style="background-color: #f0f0f0; padding: 10px; border-radius: 10px">
28             <h3>Sèries</h3>
29             <ul class="list-none p-0 m-0">
30               <li><a href="#">Sèrie 1</a></li>
31               <li><a href="#">Sèrie 2</a></li>
32             </ul>
33           </div>
34         </div>
35       </div>
36       <div class="flex flex-col gap-4 p-6" style="background-color: #f0f0f0; padding: 10px; border-radius: 10px">
37         <h3>Users</h3>
38         <ul class="list-none p-0 m-0">
39           <li><a href="#">User 1</a></li>
40           <li><a href="#">User 2</a></li>
41         </ul>
42       </div>
43     </div>
44   </div>
45 </body>
```

Regular:



SuperAdmin:



Comprovació del app

El primer pas és executar artisan queue:work

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan queue:work
INFO Processing jobs from the [default] queue.
```

Entro amb superadmin.

The image shows a login form for a video application. It consists of several input fields and buttons. At the top left is a placeholder 'Email' with the value 'superadmin@videosapp.com'. Below it is a placeholder 'Password' with several dots. To the left of the password field is a checkbox labeled 'Remember me'. At the bottom left is a link 'Forgot your password?'. On the right side is a large, dark blue 'LOG IN' button.

Email

superadmin@videosapp.com

Password

.....

Remember me

[Forgot your password?](#)

LOG IN

Creo un vídeo.

Afegir Nou Vídeo

Títol
Chill Aim 2

Descripció
123

URL
<https://www.youtube.com/embed/7E7cDTolcaQ>

Assigna a una sèrie
No pertany a una sèrie

[Tornar Enrere](#) [Desar Vídeo](#)

Una vegada creat, em surt les notificacions, que son 2 ja que hi han 2 superadmins.

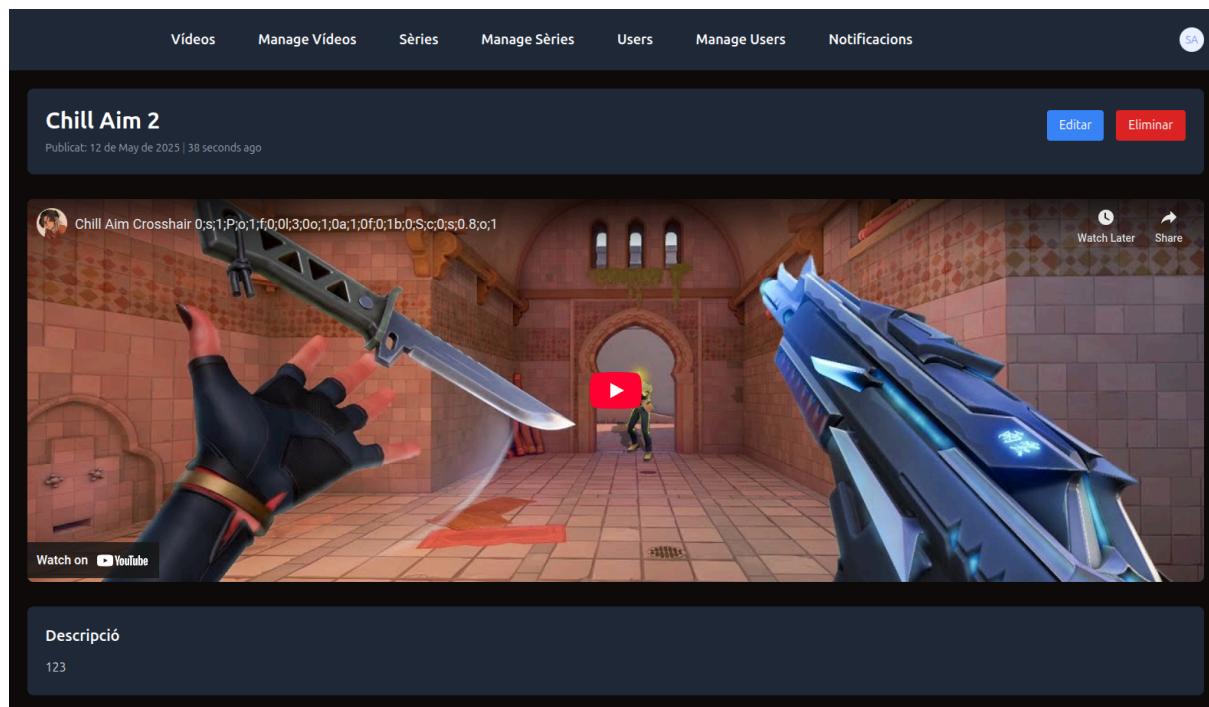
Manage Sèries Users Manage Users Notificacions

Les teves notificacions

Nou vídeo:
Chill Aim 2 [Veure vídeo](#)
1 second ago

Nou vídeo:
Chill Aim 2 [Veure vídeo](#)
1 second ago

Si cliquem veure vídeo ens surt el vídeo correctament.



Al inbox ens surten 2 correus, 1 per cada admin.

Inboxes > My Inbox

Search... ✉️ ⟳ ✉️ ⚙️

My Inbox

Integration **Email Address**

Nou vídeo creat: Chill Aim 2
to: <superadmin@videosapp.com> a few seconds ago

Nou vídeo creat: Chill Aim 2
to: <charvey@gmail.com> a few seconds ago

SMTP API

Si observo el correu veure que està tot correcte i si clico veure em porta al vídeo.

Nou vídeo creat: Chill Aim 2

From: Laravel <hello@example.com>
To: <superadmin@videosapp.com>

Show Headers

HTML HTML Source Text Raw Spam Analysis HTML Check (5) Tech Info

Mobile Desktop Computer

Nou Vídeo Creat

Hola Super Admin,

S'ha publicat un nou vídeo amb el títol:

Chill Aim 2

Si vols veure'n el detall o revisar-lo, fes clic al botó següent:

[Veure Vídeo](#)

Gràcies per formar part de la nostra comunitat!

Si tens cap dubte, contacta amb nosaltres a support@example.com.

videoNotificationsTest.php

Guia pas a pas en captures i explicacions utilitzant la metodologia TDD i AAA per als testos.

```
test_video_created_event_is_dispatched()
```

```
test_push_notification_is_sent_when_video_is_created()
```

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan make:test VideoNotificationsTest
INFO Test [tests/Feature/VideoNotificationsTest.php] created successfully.

alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$
```



```

16  class VideoNotificationsTest extends TestCase
17
18  protected function setUp(): void
19  {
20      parent::setUp();
21      // Crear permisos i rols necessaris
22      $permissions = ['videos.create', 'manage videos', 'series.create', 'manage series'];
23      foreach ($permissions as $perm) {
24          Permission::firstOrCreate(['name' => $perm]);
25      }
26      $superadminRole = Role::firstOrCreate(['name' => 'superadmin']);
27      $superadminRole->syncPermissions($permissions);
28
29      // Definir portes d'accés (com a PermissionHelper)
30      \Illuminate\Support\Facades\Gate::define('videos.create', fn($user) =>
31          $user->hasPermissionTo('videos.create') || $user->isSuperAdmin()
32      );
33
34      // Configurar el driver de broadcasting a 'null' per als tests
35      config(['broadcasting.default' => 'null']);
36  }
37
38  /**
39   * @dataProvider provideDataForTest
40   */
41  #[Test]
42  public function video_created_event_is_dispatched()
43  {
44      // Arrange
45      Event::fake([VideoCreated::class]);
46      $user = User::factory()->create(['super_admin' => true]);
47      $user->assignRole(...roles: 'superadmin');
48      $this->actingAs($user);
49
50      // Act
51      $videoData = [
52          'title' => 'Test Video',
53          'description' => 'This is a test video description',
54          'url' => 'https://example.com/video.mp4',
55      ];
56      $response = $this->post(route('videos.manage.store'), $videoData);
57
58      // Assert
59      $response->assertRedirect();
60      Event::assertDispatched( event: VideoCreated::class, function ($event) use ($videoData) {
61          $this->assertEquals($videoData['title'], $event->getTitle());
62          $this->assertEquals($videoData['description'], $event->getDescription());
63          $this->assertEquals($videoData['url'], $event->getUrl());
64      });
65  }
66
67  public function provideDataForTest(): array
68  {
69      return [
70          [
71              'superAdminRole' => true,
72              'regularUserRole' => false
73          ],
74          [
75              'superAdminRole' => false,
76              'regularUserRole' => true
77          ]
78      ];
79  }
80
81  /**
82   * @dataProvider provideDataForTest
83   */
84  public function push_notification_is_sent_when_video_is_created()
85  {
86      // Arrange
87      $user = User::factory()->create(['super_admin' => true]);
88      $user->assignRole(...roles: 'superadmin');
89      $this->actingAs($user);
90
91      // Act
92      $videoData = [
93          'title' => 'Test Video',
94          'description' => 'This is a test video description',
95          'url' => 'https://example.com/video.mp4',
96      ];
97      $response = $this->post(route('videos.manage.store'), $videoData);
98
99      // Assert
100     $this->assertEmailSent();
101     $this->assertEmailContains('Test Video');
102     $this->assertEmailContains('This is a test video description');
103     $this->assertEmailContains('https://example.com/video.mp4');
104  }
105 }
```

Cicle TDD essencial

En TDD primer escrius un test que falla (Red), després implementes just el necessari perquè passi (Green) i finalment neteges el codi sense trencar els tests (Refactor) [CodeSignal](#).

Cada test s'estructura en tres parts clares:

- **Arrange:** configura l'estat inicial i fakes necessaris. [Medium](#)
- **Act:** executa l'acció que vols provar. [Semaphore](#)
- **Assert:** verifica els resultats esperats. [Medium](#)

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ php artisan test --filter=VideoNo

    PASS Tests\Feature\VideoNotificationsTest
        ✓ video created event is dispatched
        ✓ push notification is sent when video is created

    Tests:    2 passed (6 assertions)
    Duration: 0.21s

alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ █
```

Terms.md

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

```
terms.md ×

100
101     ## Sprint 7: Correccions, notificacions i broadcast
102
103     **Correccions i millores:**
104     - Correcció d'errors detectats al **Sprint 6**.
105     - Revisió i reparació de tests d'sprints anteriors afectats per modificacions recents.
106     - Implementació de la funcionalitat perquè els **Regular Users** puguin crear sèries i afegir-hi vídeos.
107
108     **Notificacions i Broadcasting:**
109     - Creació de l'event 'VideoCreated', amb constructor, 'broadcastOn()' i 'broadcastAs()', assegurant que implementa 'ShouldBroadcast'.
110     - Disparar l'event 'VideoCreated' des del controlador en el moment de crear un vídeo.
111     - Creació del listener 'SendVideoCreatedNotification', amb el mètode 'handle(VideoCreated $event)' per:
112         - Enviar un correu electrònic als administradors.
113         - Llengar la notificació 'VideoCreated' amb informació del vídeo.
114     - Actualització del fitxer 'app/Providers/EventServiceProvider.php' per registrar l'event i el listener corresponent.
115
116     **Configuració de serveis externs:**
117     - Mailtrap/Mailchimp: Registre i configuració del '.env' per poder enviar correus des de Laravel.
118     - Pusher: Registre, configuració de credencials al '.env' i comprovació a 'config/broadcasting.php' perquè Pusher sigui el servei per defecte.
119
120     **Laravel Echo i Notificacions Push:**
121     - Instal·lació de 'laravel-echo' i 'pusher-js' via 'npm'.
122     - Configuració de Laravel Echo al fitxer 'resources/js/bootstrap.js'.
123     - Creació d'una vista per mostrar les notificacions push en temps real.
124     - Creació de la ruta per accedir a les notificacions.
125
126     **Tests:**
127     - Creació de 'VideoNotificationsTest' amb les funcions:
128         - 'test_video_created_event_is_dispatched()'
129         - 'test_push_notification_is_sent_when_video_is_created()'
130
131     **Documentació i qualitat de codi:**
132     - Actualització d'aquest document ('terms.md') amb totes les funcionalitats del Sprint 7.
133     - Verificació de qualitat i estàtica del codi amb *Larastan* per assegurar que els fitxers creats compleixen les bones pràctiques.
134
135
136
137
138
139
```

Sprint 7: Correccions, notificacions i broadcast

Correccions i millores:

- Correcció d'errors detectats al **Sprint 6**.
- Revisió i reparació de tests d'sprints anteriors afectats per modificacions recents.
- Implementació de la funcionalitat perquè els **Regular Users** puguin crear sèries i afegir-hi vídeos.

Notificacions i Broadcasting:

- Creació de l'event `'VideoCreated'`, amb constructor, `'broadcastOn()'` i `'broadcastAs()'`, assegurant que implementa `'ShouldBroadcast'`.
- Disparar l'event `'VideoCreated'` des del controlador en el moment de crear un vídeo.
- Creació del `listener 'SendVideoCreatedNotification'`, amb el mètode `'handle(VideoCreated $event)'` per:
 - Enviar un correu electrònic als administradors.
 - Llengar la notificació `'VideoCreated'` amb informació del vídeo.
- Actualització del fitxer `'app/Providers/EventServiceProvider.php'` per registrar l'event i el listener corresponent.

Configuració de serveis externs:

- `Mailtrap/Mailchimp`: Registre i configuració del `'.env'` per poder enviar correus des de Laravel.
- `Pusher`: Registre, configuració de credencials al `'.env'` i comprovació a `'config/broadcasting.php'` perquè Pusher sigui el servei per defecte.

Laravel Echo i Notificacions Push:

- Instal·lació de `'laravel-echo'` i `'pusher-js'` via `'npm'`.
- Configuració de Laravel Echo al fitxer `'resources/js/bootstrap.js'`.
- Creació d'una vista per mostrar les notificacions push en temps real.
- Creació de la ruta per accedir a les notificacions.

Tests:

- Creació de `'VideoNotificationsTest'` amb les funcions:
 - `'test_video_created_event_is_dispatched()'`
 - `'test_push_notification_is_sent_when_video_is_created()'`

Documentació i qualitat de codi:

- Actualització d'aquest document (`'terms.md'`) amb totes les funcionalitats del Sprint 7.
- Verificació de qualitat i estàtica del codi amb `Larastan` per assegurar que els fitxers creats compleixen les bones pràctiques.

Comprovar errors Larastan

No hi han errors.

```
alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$ ./vendor/bin/phpstan analyse
Note: Using configuration file /home/alumnat/Code/M7-8-9/VideoAppHarvey/phpstan.neon.
85/85 [██████████] 100%

[OK] No errors

alumnat@harvey:~/Code/M7-8-9/VideoAppHarvey$
```