

# Projecte GLOBAL: Video App

7e SPRINT



# Laravel 7e Sprint

**Professor:** Jordi Vega

**Assignatures:** M7, M8 i M9

**Nom:** Marc

**Cognoms:** Pallares Pino

# INDEX

7e Sprint  
Link al Github

3  
18



## 7e Sprint

- Corregir els errors del 6e sprint.

```
© MultimediaController.php  index.blade.php  show.blade.php  navbar.blade.php x
1  <header>
2      <div class="logo">Marc Video</div>
3      <button class="navbar-toggler" onclick="toggleNavbar()">≡</button>
4      <nav class="navbar">
5          <ul id="navbar-menu">
6              <li><a href="{{ route('home') }}">Home</a></li>
7              <li><a href="{{ route('videos.index') }}">Videos</a></li>
8              <li><a href="{{ route('users.index') }}">Usuaris</a></li>
9              @auth
10                 <li><a href="{{ route('dashboard') }}">Dashboard</a></li>
11                 @can('manage-videos')
12                     <li><a href="{{ route('videos.manage.index') }}">Administrar Videos</a></li>
13                 @endcan
14                 @can('admmistradorUsuaris')
15                     <li><a href="{{ route('users.manage.index') }}">Administrar Usuaris</a></li>
16                 @endcan
17                 <li>
18                     <form action="{{ route('logout') }}" method="POST">
19                         @csrf
20                         <button type="submit">Logout</button>
21                     </form>
22                 </li>
23             @else
24                 <li><a href="{{ route('login') }}">Login</a></li>
25                 <li><a href="{{ route('register') }}">Register</a></li>
26             @endauth

```

```

<div class="video-page">
  <div class="video-info-section">
    <div class="video-metadata">
      <div class="metadata-item">
        <p> Creat: <span class="metadata-date">{{ $video->created_at->format('F d, Y') }}</span></p>
        <p> Actualitzat: <span class="metadata-date">{{ $video->updated_at->format('F d, Y') }}</span></p>
      </div>
      <div class="metadata-actions">
        <a href="{{ $video->url }}" target="_blank" class="action-link">
          <span class="action-icon"></span>
          <span>Original Source</span>
        </a>
        @if(auth()->check() && auth()->id() === $video->user_id)
          <a href="{{ route('videos.edit', $video->id) }}" class="action-link">
            <span class="action-icon"></span>
            <span>Edit</span>
          </a>
          <form action="{{ route('videos.destroy', $video->id) }}" method="POST" style="display:inline;">
            @csrf
            @method('DELETE')
            <button type="submit" class="action-link" style="background:none; border:none; color:inherit; cursor:pointer;">
              <span class="action-icon"></span>
              <span>Delete</span>
            </button>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>

```

- En cas que al modificar el codi falla algun test d'un sprint anterior, s'han d'arreglar.
- Afegir si no s'ha fet al sprint anterior que els regulars users puguin crear series i afegir videos a la serie.

```

Route::middleware('auth')->group(function () {
    Route::get('series/create', [SeriesController::class, 'create'])->name('series.create');
    Route::post('series', [SeriesController::class, 'store'])->name('series.store');
    Route::post('series/{series}/videos', [SeriesController::class, 'addVideo'])->name('series.addVideo');
});
Route::get('series', [SeriesController::class, 'index'])->name('series.index');

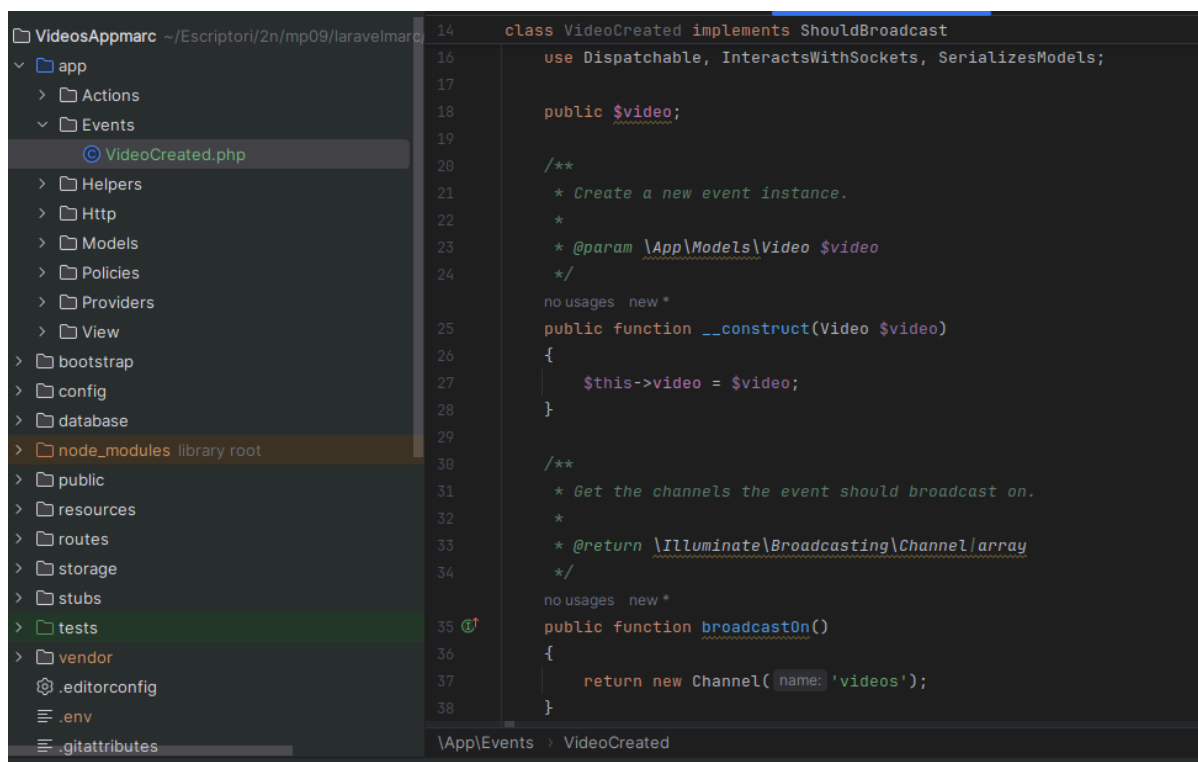
```

```
}

/**
 * Agregar un video a una serie existente.
 */
!usage new *
public function addVideo(Request $request, Series $series)
{
    $request->validate([
        'video_id' => 'required|exists:videos,id',
    ]);

    // Ensure the series belongs to the authenticated user
    if ($series->user_id !== auth()->id()) {
        return redirect()->back()->with('error', 'Unauthorized action.');
```

- Crear l'event de VideoCreated amb el constructor i BroadcastOn.



```
14 class VideoCreated implements ShouldBroadcast
15
16     use Dispatchable, InteractsWithSockets, SerializesModels;
17
18     public $video;
19
20     /**
21      * Create a new event instance.
22      *
23      * @param \App\Models\Video $video
24      */
25     no usages new *
26     public function __construct(Video $video)
27     {
28         $this->video = $video;
29     }
30
31     /**
32      * Get the channels the event should broadcast on.
33      *
34      * @return \Illuminate\Broadcasting\Channel|array
35      */
36     no usages new *
37     public function broadcastOn()
38     {
39         return new Channel( name: 'videos');
```

- Al controller s'ha de disparar l'event al crear el video.

```

php web.php index.blade.php VideoCreated.php VideosController.php x SeriesController.php Series.php
10 class VideosController extends Controller
34 public function store(Request $request)
42     'series_id' => 'nullable|exists:series,id',
43 ];
44
45 $video = new Video([
46     'title' => $request->input( key: 'title'),
47     'description' => $request->input( key: 'description'),
48     'url' => $request->input( key: 'url'),
49     'user_id' => Auth::id(),
50     'series_id' => $request->input( key: 'series_id'),
51 ]);
52
53 if ($video->save()) {
54     // Disparar el evento VideoCreated
55     event(new VideoCreated($video));
56
57     return redirect()->route( route: 'videos.index')->with('success', 'Video created successfully.');
```

- S'ha de crear el listener de la notificació SendVideoCreatedNotification i s'ha de crear 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.

Project
VideosAppmarc
app
Actions
Events
Helpers
Http
Listeners
SendVideoCreatedNotification.php
Models
Notifications
VideoCreatedNotification.php
Policies
Providers
View
bootstrap
config
database
node\_modules library root
public
resources
routes
storage
tests

```

<?php
namespace App\Notifications;

use Illuminate\Bus\Queueable;
use Illuminate\Notifications\Notification;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Notifications\Messages\MailMessage;
use App\Models\Video;

class VideoCreatedNotification extends Notification implements ShouldQueue
{
    use Queueable;

    public $video;

    /**
     * Create a new notification instance.
     *
     * @param \App\Models\Video $video
     */
    public function __construct(Video $video)
    {

```

```

10 class SendVideoCreatedNotification
11
12 /**
13  * Handle the event.
14  *
15  * @param \App\Events\VideoCreated $event
16  * @return void
17  */
18 new *
19 public function handle(VideoCreated $event)
20 {
21     $video = $event->video;
22
23     // Enviar correu als administradors
24     $admins = User::where('role', 'admin')->get();
25     foreach ($admins as $admin) {
26         Mail::to($admin->email)->send(new \App\Mail\VideoCreatedMail($video));
27     }
28
29     // Enviar notificació VideoCreated
30     foreach ($admins as $admin) {
31         $admin->notify(new VideoCreatedNotification($video));
32     }
33 }

```

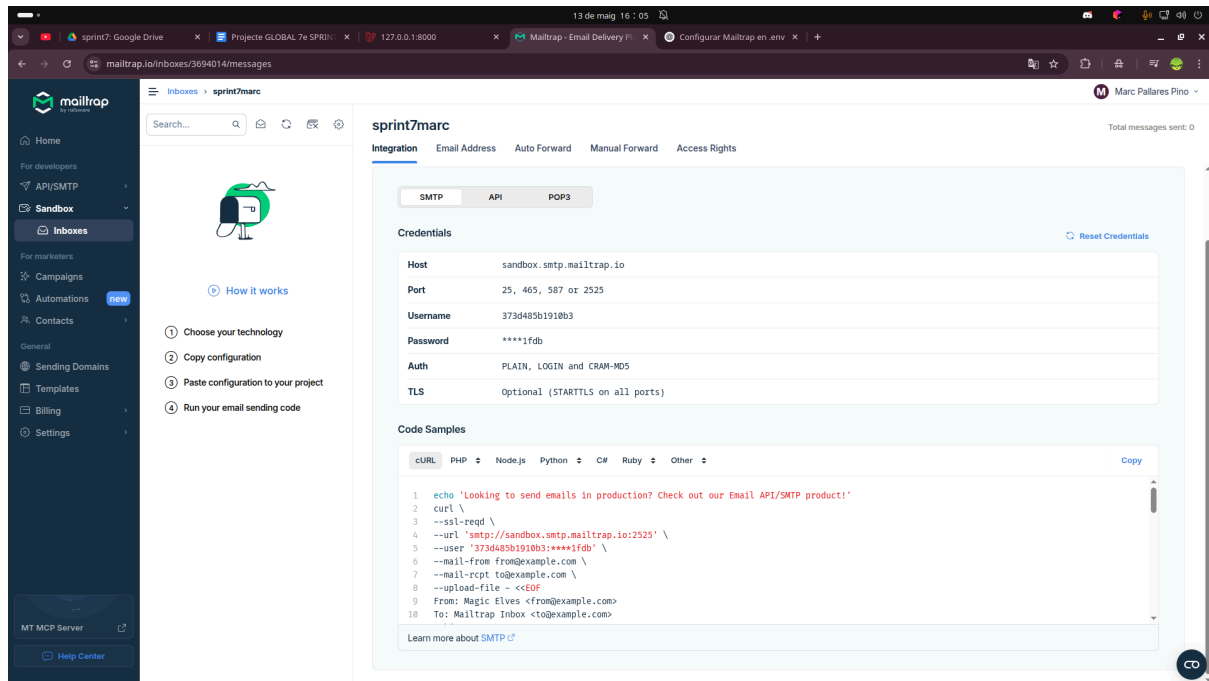
- S'ha de crear app/Providers/EventServiceProvider.php i afegir la notificació de la creació del video.

```

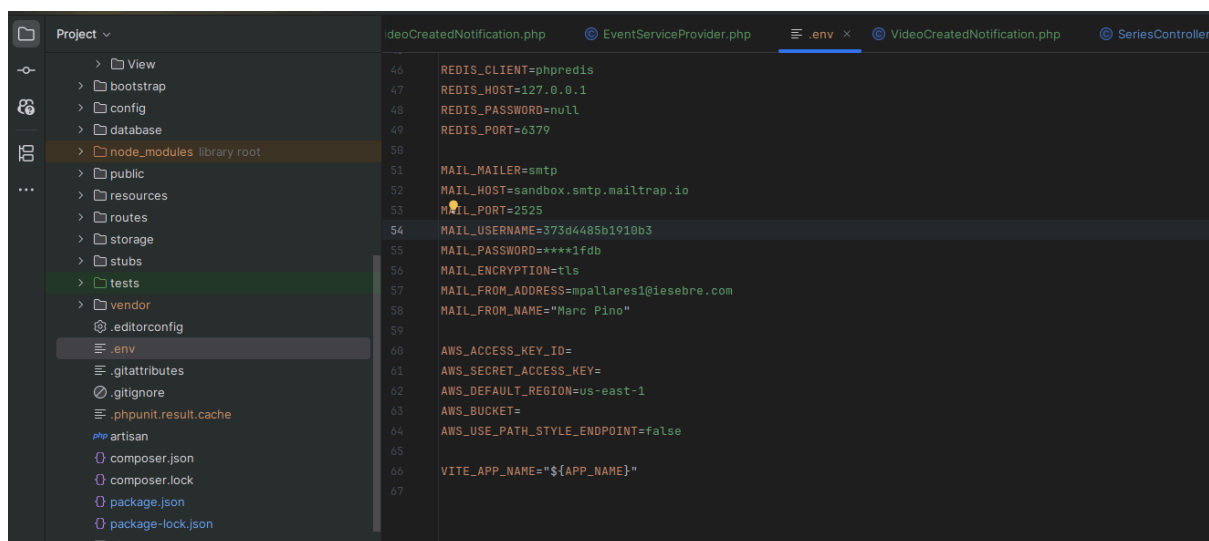
10 class EventServiceProvider extends ServiceProvider
11
12 /**
13  * The event to listener mappings for the application.
14  *
15  * @var array
16  */
17 no usages
18 protected $listen = [
19     VideoCreated::class => [
20         SendVideoCreatedNotification::class,
21     ],
22 ];
23
24 /**
25  * Register any events for your application.
26  *
27  * @return void
28  */
29 no usages new *
30 public function boot()
31 {
32     parent::boot();
33 }

```


- Registrar-se a Mailtrap o Mailchimp. Per utilitzar el servidor de correus.



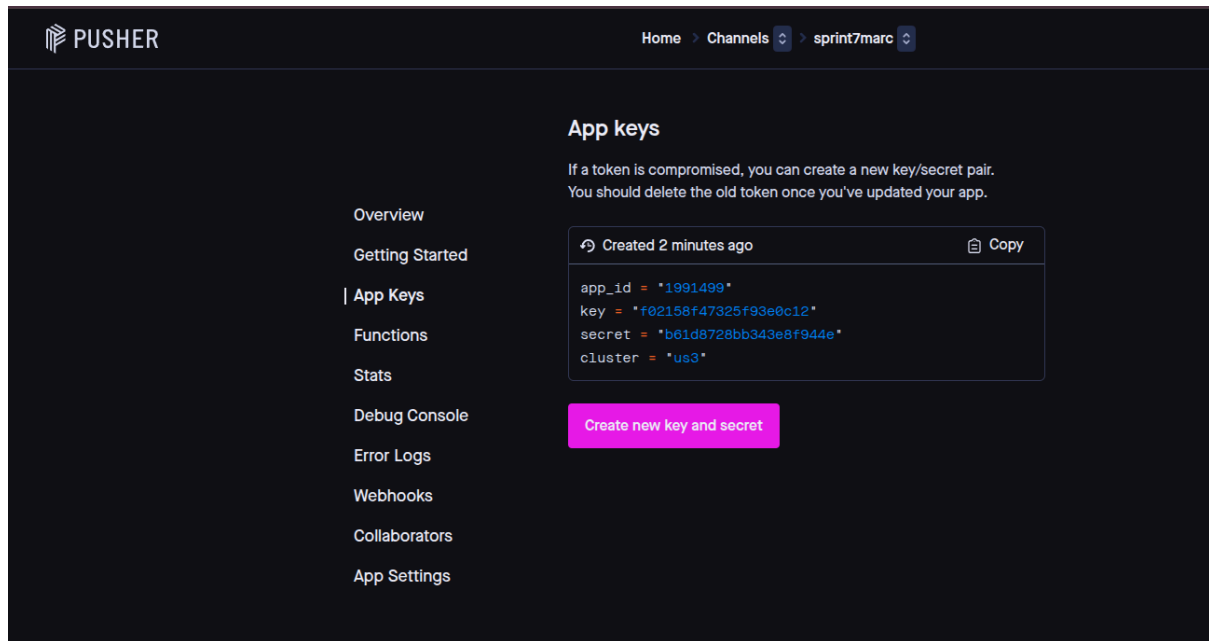
- Configurar el .env per tal d'utilitzar les credencials de Mailtrap/Mailchimp/EmailJs.



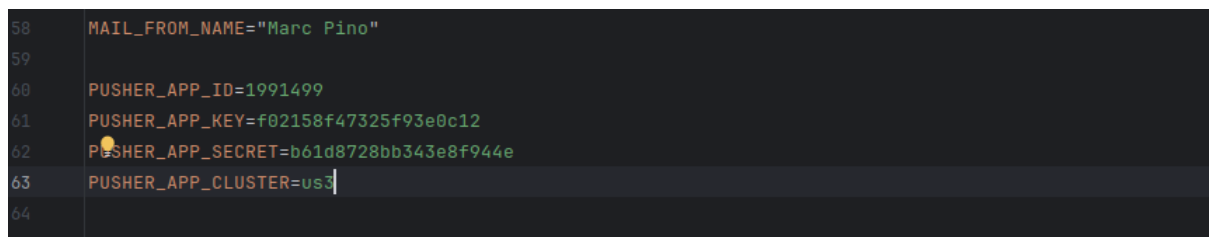


 <b>INSTITUT DE L'EBRE</b> TORTOSA	<b>GLOBAL M7, M8 i M9</b>	<b>DAM</b> <b>2n</b>
--	---------------------------	-------------------------

- Registrar-se a pusher.



- Configurar .env amb les credencials de pusher.



- A config/broadcasting.php revisar que pusher estigui configurat per defecte.

The screenshot shows an IDE with the following components:

- Project Explorer:** A sidebar on the left showing the project structure. The 'config' directory is expanded, listing various configuration files. 'broadcasting.php' is selected and highlighted.
- Code Editor:** The main area displays the 'broadcasting.php' file. It contains a PHP array configuration for Pusher. The 'default' key is set to 'pusher'. The 'connections' array includes a 'pusher' connection with the following settings:
  - 'driver' => 'pusher'
  - 'key' => env('key: 'PUSHER\_APP\_KEY')
  - 'secret' => env('key: 'PUSHER\_APP\_SECRET')
  - 'app\_id' => env('key: 'PUSHER\_APP\_ID')
  - 'options' => [
    - 'cluster' => env('key: 'PUSHER\_APP\_CLUSTER')
    - 'useTLS' => true,
- Terminal:** At the bottom, there are tabs for 'Local' and 'Local (2)', both of which are currently empty.

- A App/Events/VideoCreated.php afegir la funció broadcastAs() i mirar que implementa shouldBroadcast.

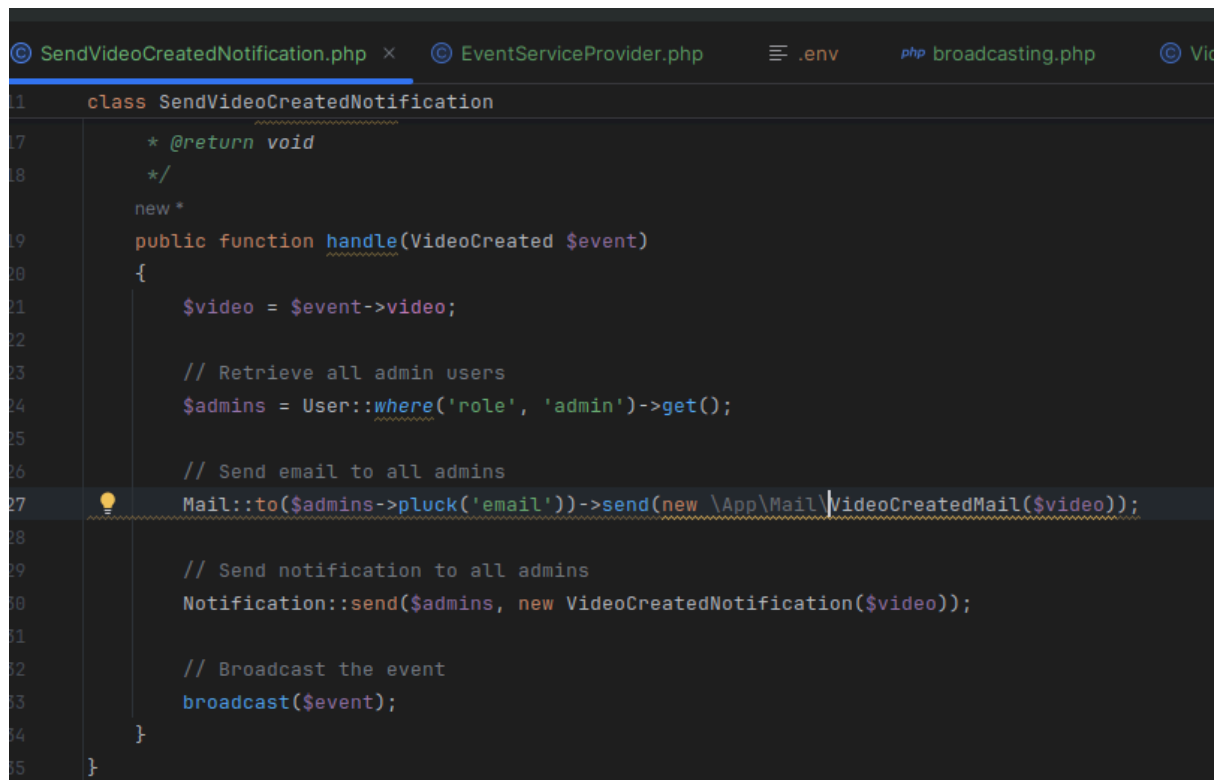
The screenshot shows an IDE with the following components:

- Project Explorer:** The 'app' directory is expanded, showing 'Events'. 'VideoCreated.php' is selected and highlighted.
- Code Editor:** The main area displays the 'VideoCreated.php' file. It shows a class definition:
 

```
class VideoCreated implements ShouldBroadcast
```

 The class has two methods:
  - `broadcastOn()`: A public function that returns a new Channel with the name 'videos'.
  - `broadcastAs()`: A public function that returns the string 'video.created'.
- Terminal:** At the bottom, there are tabs for 'Local' and 'Local (2)', both of which are currently empty.

- A SendVideoCreatedNotification.php assegurar-se que pusher transmet l'event.



```
11 class SendVideoCreatedNotification
12
13     * @return void
14     */
15     new *
16     public function handle(VideoCreated $event)
17     {
18         $video = $event->video;
19
20         // Retrieve all admin users
21         $admins = User::where('role', 'admin')->get();
22
23         // Send email to all admins
24         Mail::to($admins->pluck('email'))->send(new \App\Mail\VideoCreatedMail($video));
25
26         // Send notification to all admins
27         Notification::send($admins, new VideoCreatedNotification($video));
28
29         // Broadcast the event
30         broadcast($event);
31     }
32 }
```

- Registrar event a app/Providers/EventServiceProvider.php.

```
26      * @return void
27      */
28      no usages new *
29      public function boot()
30      {
31          // Call the parent boot method to ensure proper functionality
32          parent::boot();
33
34          // Additional event registration logic can be added here if needed
35      }
36
37      /**
38       * Determine if events and listeners should be automatically discovered.
39       *
40       * @return bool
41       */
42      no usages new *
43      public function shouldDiscoverEvents()
44      {
45          // Enable automatic event discovery if required
46          return false;
47      }
```

App\Providers > EventServiceProvider

- Al controller disparar l'event del push.

```
16      $users = $video->testedByUsers;
17
18      if ($users->isEmpty()) {
19          return response()->json(['message' => 'No users have tested this video.'], status: 404);
20      }
21
22      return response()->json($users);
23  }
24
25  /**
26   * Authorize the action for the given resource.
27   *
28   * @param string $action
29   * @param string $model
30   * @return void
31   */
32  private function authorize(string $action, string $model)
33  {
34      if (!Auth::user()->can($action, $model)) {
35          abort( code: 403, message: 'Unauthorized action.' );
36      }
37  }
38  }
```

- S'ha d'instal·lar laravel-echo pusher-js en npm.

```
Terminal Local x Local (2) x + v
alumnat@marc:~/Escriptori/2n/mp09/laravelmarc/VideosAppmarc$ npm install --save laravel-echo pusher-js
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'laravel-echo@2.1.0',
npm WARN EBADENGINE   required: { node: '>=20' },
npm WARN EBADENGINE   current: { node: 'v18.19.1', npm: '9.2.0' }
npm WARN EBADENGINE }
( [REDACTED] ) ⚙ reify:pusher-js: http fetch GET 200 https://registry.npmjs.org/pusher-js/-/pusher-js-8.4.0.tgz 198ms
```

- S'ha de configurar laravel echo a resources/js/[bootstrap.js](#)

```
© VideosController.php js bootstrap.js x
1 import axios from 'axios';
2 import Echo from 'laravel-echo';
3 import Pusher from 'pusher-js';
4
5 window.axios = axios;
6
7 window.axios.defaults.headers.common['X-Requested-With'] = 'XMLHttpRequest';
8
9 // Configure Pusher
10 window.Pusher = Pusher;
11
12 // Configure Laravel Echo
13 window.Echo = new Echo({ options: {
14   broadcaster: 'pusher',
15   key: process.env.MIX_PUSHER_APP_KEY,
16   cluster: process.env.MIX_PUSHER_APP_CLUSTER,
17   forceTLS: true,
18 }});
```

- Crear vista de notificaciones push, per mostrar les notificaciones, prèviament s'han d'escoltar.

```
index.blade.php  php web.php  php NotificationsController.php x
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class NotificationController extends Controller
8  {
9      public function index()
10     {
11         return view('notifications.index');
12     }
13 }
14
```

```
Route::get(uri: '/series/create', [SeriesController::class, 'create'])->name(name: 'series.create');
Route::post(uri: '/series', [SeriesController::class, 'store'])->name(name: 'series.store');
Route::post(uri: '/series/{series}/videos', [SeriesController::class, 'addVideo'])->name(name: 'series.addVideo');
});

Route::get(uri: '/notifications', [NotificationController::class, 'index'])->name(name: 'notifications.index');

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

```
index.blade.php x  php web.php  php NotificationsController.php  Js app.js
1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container">
5          <h1>Push Notifications</h1>
6          <ul id="notification-list" class="list-disc pl-5">
7              <!-- Les notificacions es mostraran aquí -->
8          </ul>
9      </div>
10 @endsection
```

```
index.blade.php  php web.php  php NotificationsController.php  Js app.js x
1  import './bootstrap';
2  new *
3  window.Echo.private(`App.Models.User.${userId}`)
4      .notification((notification) :void => {
5          console.log(notification);
6          // Aquí pots actualitzar la interfície d'usuari amb la notificació rebuda
7          const notificationList :HTMLLIElement = document.getElementById('notification-list');
8          const newNotification :HTMLLIElement = document.createElement('li');
9          newNotification.textContent = notification.message; // Assumeix que la notificació té un camp "message"
10          notificationList.appendChild(newNotification);
11      });
```

- Crear ruta notificacions.

```
22
23
24  Route::get('uri: '/notifications', [NotificationsController::class, 'index'])->name('name: 'notifications.index');
25
26
27
```

- A videoNotificationsTest crear les funcions:  
test\_video\_created\_event\_is\_dispatched()  
test\_push\_notification\_is\_sent\_when\_video\_is\_created()

```
index.blade.php  php web.php  VideoNotificationsTest.php x  app.js
10 use App\Notifications\VideoCreatedNotification;
11
12 new *
12 >> class VideoNotificationsTest extends TestCase
13 {
14 new *
14 > public function test_video_created_event_is_dispatched()
15 {
16     // Arrange: Fake the event
17     Event::fake();
18
19     // Act: Create a video
20     $video = Video::factory()->create();
21
22     // Assert: Check if the VideoCreated event was dispatched
23     Event::assertDispatched(event: VideoCreated::class, function ($event) use ($video) {
24         return $event->video->id === $video->id;
25     });
26 }
27
28 new *
28 > public function test_push_notification_is_sent_when_video_is_created()
29 {
30     // Arrange: Fake the notification
31     Notification::fake();
32 }
```

```
UpdatePasswordTest.php
UpdateTeamMemberRoleTest.php
UpdateTeamNameTest.php
VideoNotificationsTest.php
```



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

```

133 ## Sprint 7
134 **Correcció d'errors**: Corregir els errors detectats en el Sprint 6.
135
136 **Notificacions**:
137 - S'ha creat la ruta '/notifications' al fitxer 'routes/web.php' per accedir a les notificacions.
138 - S'ha creat el controlador 'NotificationsController' amb la funció 'index()' per mostrar la vista de notifi
139 - S'ha creat la vista 'resources/views/notifications/index.blade.php' per mostrar les notificacions.
140
141 **Events i Notificacions**:
142 - S'ha creat l'event 'VideoCreated' per gestionar la creació de vídeos.
143 - S'ha creat la notificació 'VideoCreatedNotification' per enviar notificacions quan es crea un vídeo.
144
145 **Tests**:
146 - S'ha creat el test 'VideoNotificationsTest' a 'tests/Feature/' amb les funcions:
147   - 'test_video_created_event_is_dispatched': Verifica que l'event 'VideoCreated' es dispara quan es crea
148   - 'test_push_notification_is_sent_when_video_is_created': Verifica que es genera una notificació quan es
149
150 **Helpers**:
151 - S'han revisat i actualitzat els helpers per gestionar les notificacions.
152
153 **Routes**:
154 - S'han revisat i actualitzat les routes per incloure la gestió de notificacions.
155
156 **Afegir a resources/markdown/terms**: Afegir el que s'ha fet al Sprint 7 a 'resources/markdown/terms'.
157
  
```

**Sprint 7**  
Correcció d'errors: Corregir els errors detectats en el Sprint 6.

**Notificacions:**

- S'ha creat la ruta '/notifications' al fitxer 'routes/web.php' per accedir a les notificacions.
- S'ha creat el controlador 'NotificationsController' amb la funció 'index()' per mostrar la vista de notificacions.
- S'ha creat la vista 'resources/views/notifications/index.blade.php' per mostrar les notificacions.

**Events i Notificacions:**

- S'ha creat l'event 'VideoCreated' per gestionar la creació de vídeos.
- S'ha creat la notificació 'VideoCreatedNotification' per enviar notificacions quan es crea un vídeo.

**Tests:**

- S'ha creat el test 'VideoNotificationsTest' a 'tests/Feature/' amb les funcions:
  - 'test\_video\_created\_event\_is\_dispatched': Verifica que l'event 'VideoCreated' es dispara quan es crea un vídeo.
  - 'test\_push\_notification\_is\_sent\_when\_video\_is\_created': Verifica que es genera una notificació quan es crea un vídeo.

- Comprovar en Larastan tots els fitxers que heu creat.

```

alumnat@marc:~/Escriptori/2n/mp09/laravelmarc/VideosAppmarc$ ./vendor/bin/phpstan analyse
Note: Using configuration file /home/alumnat/Escriptori/2n/mp09/laravelmarc/VideosAppmarc/phpstan.neon.
 78/78 [████████████████████████████████████████] 100%

[OK] No errors

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



## Link al Github

<https://github.com/Mpallares1/laravelmarc>