

Projecte IonicVideosApp



Professor: Jordi Vega

Assignatura: M8 Programació multimèdia i dispositius mòbils

Nom: Marc


Cognoms: Pallares Pino

INDEX

Que s'ha de fer?	3
Backend (Laravel - VideosApp):	3
Frontend (Ionic en vue):	8
Que s'ha d'entregar?	15



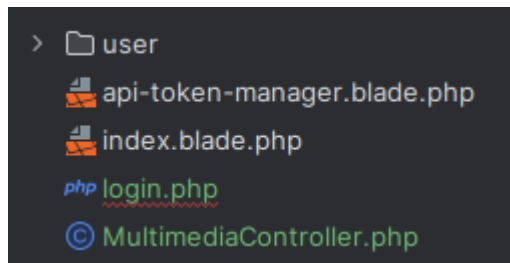
INDEX
better.parts.faster.

 INSTITUT DE L'EBRE TORTOSA	M8 Programació multimèdia i dispositius mòbils	DAM 2n
--	--	------------------

Que s'ha de fer?

Backend (Laravel - VideosApp):

- Crear el localStorage.
- Crear el controller ApiMultimediaController (Per tal de guardar arxius de vídeo i fotos)



```
© MultimediaController.php x  © ApiMultimediaControllerTest.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Storage;
7  use App\Models\Multimedia;
8
9  6 usages new *
10 class MediaController extends Controller
11 {
12     2 usages new *
13     public function upload(Request $request)
14     {
15         $request->validate([
16             'file' => 'required|file|mimes:jpg,jpeg,png,mp4|max:10240', // Supports images and videos, max 10MB
17         ]);
18
19         $path = $request->file('file')->store('uploads', ['options' => 'public']);
20
21         $user = $request->user();
22         $media = $user->media()->create([
23             'name' => $request->file('file')->getClientOriginalName(),
24             'url' => Storage::url($path),
25         ]);
26
27         return response()->json([
28             'message' => 'File uploaded successfully.',
29             'media' => $media,
30         ], status: 201);
31     }
32
33     2 usages new *
34     public function getUserMedia(Request $request)
35     {
36         $user = $request->user();
37         $media = $user->media()->get();
38
39         return response()->json([
40             'message' => 'User media retrieved successfully.',
41             'media' => $media,
42         ]);
43     }
44 }
```

App\Http\Controllers > MediaController

```

1 <?php
2
3 namespace Tests\Feature;
4
5 use Illuminate\Foundation\Testing\RefreshDatabase;
6 use Illuminate\Http\UploadedFile;
7 use Illuminate\Support\Facades\Storage;
8 use Tests\TestCase;
9 use App\Models\Multimedia;
10
11 class ApiMultimediaControllerTest extends TestCase
12 {
13     use RefreshDatabase;
14
15     public function test_can_upload_multimedia()
16     {
17         Storage::fake('local');
18
19         $response = $this->postJson('api/multimedia', [
20             'file' => UploadedFile::fake()->image('photo.jpg'),
21         ]);
22
23         $response->assertStatus(201)
24             ->assertJsonStructure(['id', 'path', 'name', 'type', 'created_at', 'updated_at']);
25
26         Storage::disk('local')->assertExists('path');
27     }
28
29     public function test_can_list_multimedia()
30     {
31         Multimedia::factory()->count(3)->create();
32
33         $response = $this->getJson('api/multimedia');
34
35         $response->assertStatus(200)
36             ->assertJsonStructure([
37                 '*' => ['id', 'name', 'path', 'type', 'created_at', 'updated_at'],
38             ])
39             ->assertJsonCount(3);
40     }
41
42     public function test_can_show_multimedia()
43     {
44     }
45 }

```

Feature
\$multimedia: Collection|Model
Source: ...
/tests/Feature/ApiMultimediaControllerTest.php

- Crear el Model

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Multimedia extends Model
9 {
10     use HasFactory;
11
12     no usages
13     protected $fillable = ['name', 'url'];
14
15     public function user()
16     {
17         return $this->belongsTo('related: User::class');
18     }
19 }
```

- Crear la api.

```
<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AuthController;
use App\Http\Controllers\UsersController;
use App\Http\Controllers\MultimediaController;
use App\Http\Controllers\MediaController;

Route::get(uri: '/user', function (Request $request) {
    return $request->user();
})->middleware(middleware: 'auth:sanctum');

Route::apiResource(name: 'multimedia', controller: AuthController::class);
Route::post(uri: '/login', [AuthController::class, 'login']);
Route::middleware(middleware: 'auth:sanctum')->group(function () {
    Route::get(uri: '/user/profile', [UserController::class, 'profile']);
});
Route::get(uri: '/multimedia', [MultimediaController::class, 'index']);

Route::post(uri: '/api/user/upload', [MediaController::class, 'upload'])->middleware(middleware: 'auth:api');
Route::get(uri: '/api/user/media', [MediaController::class, 'getUserMedia'])->middleware(middleware: 'auth:api');

Route::middleware(middleware: 'auth:api')->group(function () {
    Route::get(uri: '/api/user/media', [MediaController::class, 'getUserMedia']);
    Route::post(uri: '/api/user/upload', [MediaController::class, 'upload']);
    Route::delete(uri: '/api/user/media/{id}', [MediaController::class, 'deleteMedia']);
});

Route::post(uri: '/api/user/upload', [MultimediaController::class, 'upload']);
Route::post(uri: '/register', [AuthController::class, 'register']);
```

- Crear els tests.

The screenshot shows a PHP IDE with a project structure on the left and a code editor on the right. The project structure includes a 'tests' directory with a 'Feature' subdirectory containing 'Videos'. The 'ApiMultimediaControllerTest.php' file is selected. The code in the editor is as follows:


```
<?php
1
2
3 namespace Tests\Feature;
4
5 use Illuminate\Foundation\Testing\RefreshDatabase;
6 use Illuminate\Http\UploadedFile;
7 use Illuminate\Support\Facades\Storage;
8 use Tests\TestCase;
9 use App\Models\Multimedia;
10
11 class ApiMultimediaControllerTest extends TestCase
12 {
13     use RefreshDatabase;
14
15     public function test_can_upload_multimedia()
16     {
17         Storage::fake('local');
18
19         $response = $this->postJson('uri: /api/multimedia', [
20             'file' => UploadedFile::fake()->image('name: photo.jpg'),
21         ]);
22
23         $response->assertStatus(status: 201)
24             ->assertJsonStructure(['id', 'path', 'name', 'type']);
25
26         Storage::disk(name: 'local')->assertExists(path: 'multimedia/' . $response->json(key: 'path'));
27     }
28
29     public function test_can_list_multimedia()
30     {
31         Multimedia::factory()->count(count: 3)->create();
32
33         $response = $this->getJson(uri: '/api/multimedia');
34
35         $response->assertStatus(status: 200)
36             ->assertJsonStructure([
37                 '*' => ['id', 'name', 'path', 'type', 'created_at', 'updated_at'],
38             ])
39             ->assertJsonCount(count: 3);
40     }
41
42     public function test_can_show_multimedia()
```

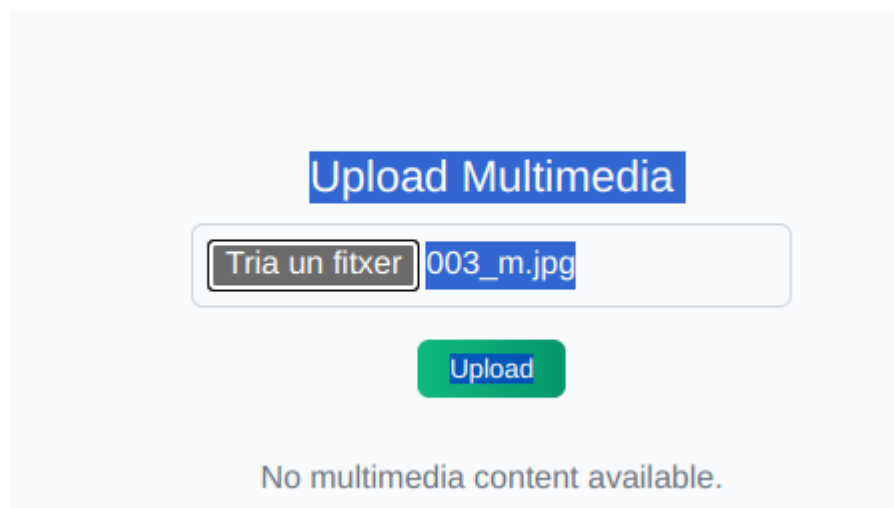
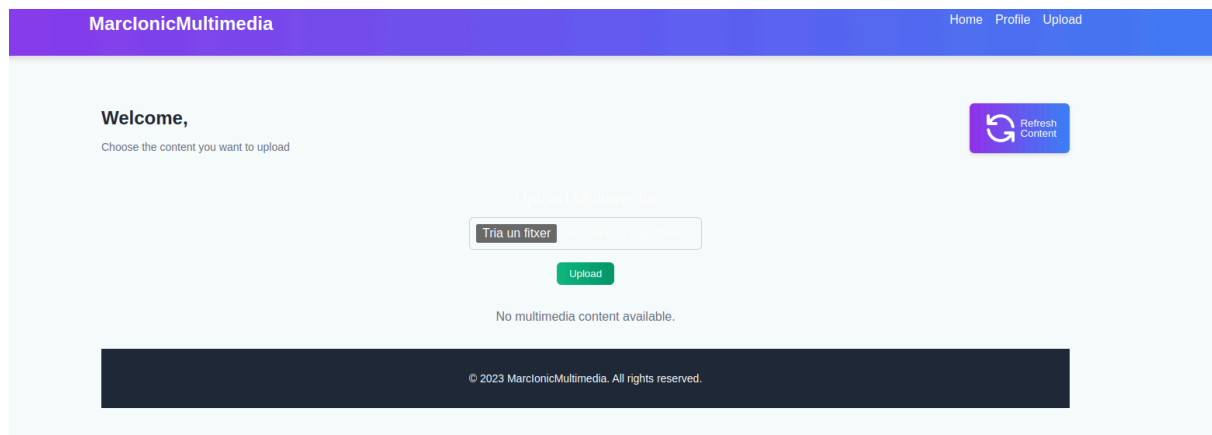
Frontend (Ionic en vue):

- Crear l'aplicació ionic en vue, s'ha de dir ionicVideosApp i el vostre nom.

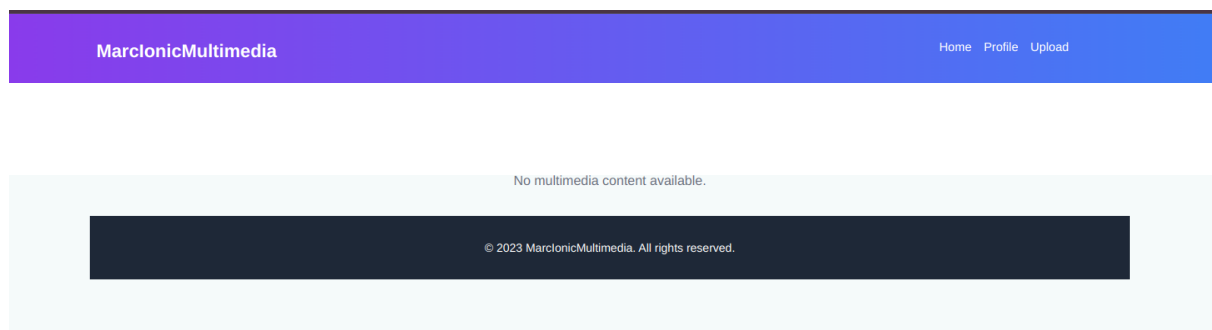

```
{  
  "name": "IonicVideosAppMarc",  
  "private": true,  
  "version": "0.0.1",  
  "type": "module",  
  "scripts": {  
    "dev": "vite",  
    "build": "vue-tsc && vite build",  
    "preview": "vite preview",  
    "test:e2e": "cypress run",  
    "test:unit": "vitest",  
    "lint": "eslint ."  
  },  
  "dependencies": {  
    "@capacitor/app": "7.0.0",  
    "@capacitor/core": "7.2.0",  
    "@capacitor/haptics": "7.0.0",  
    "@capacitor/keyboard": "7.0.0",  
    "@capacitor/status-bar": "7.0.0",  
    "@ionic/vue": "^8.0.0",  
    "@ionic/vue-router": "^8.0.0",  
    "axios": "^1.8.4",  
    "filepond": "^4.32.7",  
    "filepond-plugin-file-validate-type": "^1.2.9",  
    "filepond-plugin-image-preview": "^4.6.12",  
    "ionicons": "^7.0.0",  
    "pinia": "^3.0.1",  
    "vue": "^3.3.0".  
  }  
}
```

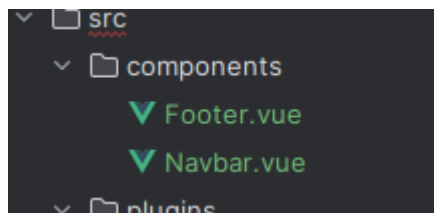
- Vista principal amb els arxius multimedia

	<p>M8</p> <p>Programació multimèdia i dispositius mòbils</p>	<p>DAM</p> <p>2n</p>
---	---	------------------------------------



- Navbar i Footer ()





footer

```
▼ HomePage.vue  ▼ Footer.vue x  {} package.json  ▼ Upload.vue
1  <template> Show component usages
2    <footer class="footer">
3      <div class="footer-content">
4        <p>© 2023 MarcIonicMultimedia. All rights reserved.</p>
5      </div>
6    </footer>
7  </template>
8
9  <script>
10   Show usages new *
11   export default {
12     name: "Footer",
13   };
14 </script>
15
16 <style scoped>
17   .footer {
18     background-color: #1f2937;
19     color: white;
20     text-align: center;
21     padding: 1rem;
22     margin-top: 2rem;
23   }
24
25   .footer-content {
26     max-width: 1280px;
27     margin: 0 auto;
28   }
```

navbar

```
▼ HomePage.vue  ▼ Footer.vue  ▼ Navbar.vue ×  package.json
1  <template> Show component usages
2    <header class="navbar">
3      <div class="navbar-content">
4        <h1 class="navbar-title">MarcIonicMultimedia</h1>
5        <nav class="navbar-links">
6          <router-link to="/" class="navbar-link">Home</router-link>
7          <router-link to="/profile" class="navbar-link">Profile</router-link>
8          <router-link to="/upload" class="navbar-link">Upload</router-link>
9        </nav>
10      </div>
11    </header>
12  </template>
13
14  <script>
15    Show usages new *
16    export default {
17      name: "Navbar",
18    };
19  </script>
20
21  <style scoped>
22  .navbar {
23    background: linear-gradient(to right, #9333ea, #3b82f6);
24    padding: 1rem;
25    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
26  }
```

- S'ha connectar el login i registre amb l'api


login


```
55 <script>
59   data() {
62     password: "",
63     error: "",
64   };
65 },
66   methods: {
67     async login() {
68       this.error = "";
69
70       try {
71         const response = await axios.post( url: "/api/login", data: {
72           email: this.email,
73           password: this.password,
74         });
75
76         localStorage.setItem("token", response.data.token);
77         this.$router.push("/"); // Redirige al HomePage
78       } catch (error) {
79         this.error = error.response?.data?.message || "Login failed. Please try again.";
80       }
81     }
82   },
83 };
84 </script>
85
```


register

Create an Account

Sign up to get started







Sign up

WHERE						ORDER BY	
	id	name	email	email_verified_at	password		
1	1	Super Admin	superadmin@videosapp.com	<null>	\$2y\$12\$6xIKKc9R8sh0n1anCiyXne7Wtry...		
2	2	regular	regular@videosapp.com	<null>	\$2y\$12\$BecR6n6MH56EVqe0o100k.lpY/j...		
3	3	Video Manager	videosmanager@videosapp.com	<null>	\$2y\$12\$ogfRMuXA0LQ6xLIxPKG70eWyhwF...		
4	4	Default Professor	professor@videosapp.com	<null>	\$2y\$12\$8DvvQZT89acXvCkocC0x0B4FGi...		
5	5	Default Alumne	alumne@videosapp.com	<null>			
6	7	prova	mpallares@iesebre.com	<null>	\$2y\$12\$fLL3CBy2qm7uAF1ncB5m2uZLoRz...		

```

1  <template> Show component usages
2  <div class="container">
3    <div class="register-card">
4      <div class="header">
5        <h2>Create an Account</h2>
6        <p>Sign up to get started</p>
7      </div>
8      <div class="form-container">
9        <form @submit.prevent="register">
10         <div class="input-group">
11           <label for="name">Full Name</label>
12           <div class="input-wrapper">
13             <svg class="icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" stroke="currentColor">
14               <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M5.121 17.804A4 4 0 112 15a4 4 0 016.879 2.804M15 11a4 4 0 10-8 0 4 4 0 008 0z" />
15             </svg>
16             <input
17               id="name"
18               type="text"
19               v-model="name"
20               required
21               placeholder="Your full name"
22             />
23           </div>
24         </div>
25
26         <div class="input-group">
27           <label for="email">Email Address</label>
28           <div class="input-wrapper">
29             <svg class="icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" stroke="currentColor">
30               <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M16 12a4 4 0 10-8 0 4 4 0 008 0z" />
31             </svg>
32             <input
33               id="email"
34               type="email"
35               v-model="email"
36               required
37               placeholder="you@example.com"
38             />
39           </div>
40         </div>
41
42         <div class="input-group">
43           <label for="password">Password</label>

```

- Una vegada loguejat, a la pàgina de l'usuari han d'aparèixer els seus arxius i ha de poder navegar a la pàgina principal on hi ha tots els arxius multimedia.

Aqui em peta tot ja , a partir de aqui no em funciona jno se perque , em deixa connectarme i tot perque em surt a la terminal del backedn que mestic connectant , que estic accedint pero no em deixa algo malament tin fet i no se el que

- S'ha d'implementar el CRUD de multimedia mitjançant l'api.
- S'ha d'utilitzar el Filepond per poder arrastrar els vídeos per a pujar-se al create.

Que s'ha d'entregar?

Github

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



INSTITUT DE L'EBRE
TORTOSA

M8

Programació multimèdia i dispositius mòbils

DAM
2n