

6 DE DICIEMBRE DE 2024



ME ESTÁN FILMANDO

MEMORIA DEL PROYECTO CFGS - DESARROLLO DE APLICACIÓN WEB

JESÚS SORRIBAS ÁLVAREZ – CURSO 2024/25

IES Bernaldo de Quirós

Contenido

Introducción.....	2
Tecnologías.....	3
Análisis.....	4
Vistas	5
Lógica.....	12
Implementación.....	13
Despliegue	19
Conclusión	28
Fuentes bibliográficas	28

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Introducción

Me están filmando es una aplicación que recomienda películas a sus clientes en función de sus gustos y que también les ofrece otros filmes si prefieren otras cosas en algún momento determinado y les indica donde poder encontrar la película en cuestión de forma legal. El funcionamiento es sencillo, introduciendo unos pocos datos personales, la aplicación se pone a funcionar y muestra por pantalla las películas que le pueden interesar al usuario. Si el cliente ya ha visto la película recomendada, la puede marcar como vista, de este modo no se le recomendará más.

Según datos del segundo trimestre de 2023, en España hay más de 25 millones de suscriptores a plataformas de video bajo demanda. Por cuota de mercado, algunas plataformas son Prime Video, Netflix, Disney+, HBO Max, Filmin, Apple tv o Movistar+.

Cuando alguien quiere ver una película y no sabe cuál, tiende a ver lo que está en primera plana, cosa que no es sinónimo de calidad; o se deja recomendar películas típicas que no le tienen por qué gustar. Otra opción con la que cuentan los suscriptores es con el pobre filtrado de las propias plataformas de streaming. Estas recomiendan películas que le puedan gustar al cliente, pero para ello ya tendría que haber visto películas antes en la plataforma. También puede ocurrir que el usuario se coma mucho la cabeza para escoger una película y pierda mucho tiempo decidiendo. Todos estos casos tendrán una solución en común, “Me están filmando”.

La gente, hoy en día, quiere entretenimiento, y con mi aplicación podrán encontrar lo que quieren y lo que les gusta perdiendo el menor tiempo posible a la hora de escoger una película de calidad que ver, en un mundo donde todos queremos el contenido cada vez más rápido.

El objetivo de la aplicación es ayudar a la gente a encontrar contenido que les gusta de forma fácil y rápida, haciendo que pierdan el menor tiempo posible en encontrarlo. Además, se da a conocer el cine no comercial a más población, que puede ver algo que les haga pensar y que realmente merezca la pena.

La página está dirigida a personas de todas las edades, solo se necesita acceso a internet. Cualquiera puede entrar y dejarse recomendar, si bien es cierto que cuanto más joven, más fácil resultará desenvolverse, la aplicación tendrá una interfaz lo suficientemente intuitiva como para que gente de mayor edad pueda interactuar. En cuanto al perfil del usuario, se trata de alguien al que le guste el cine y pase tiempo viendo películas por norma general, pudiendo visitar la página gente menos habituada a ver películas de vez en cuando.

Tecnologías

Para la realización del proyecto, se optó por la utilización de Laravel. Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP.

Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades.

Lo primero que necesitamos es Composer, que es un sistema de gestión de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías de PHP.

También requerimos node.js. Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono.

Además, se utiliza la ayuda Laravel Breeze, que agiliza la tarea de crear logins. Esto es, crea automáticamente la lógica e incluso las vistas del registro e inicio de sesión de usuarios. Laravel Breeze requiere Node para su funcionamiento.

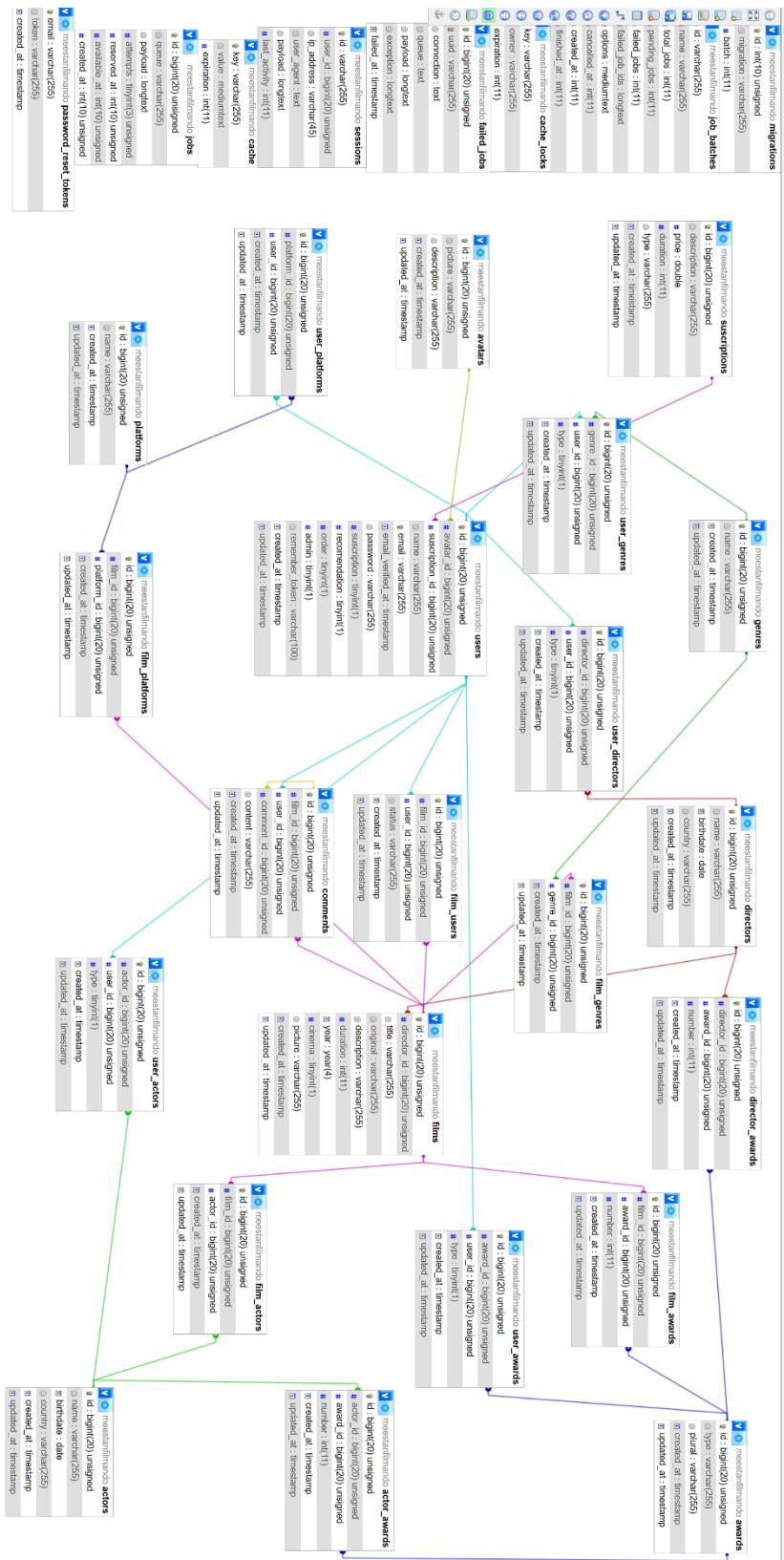
Para la parte gráfica, se utiliza Blade. Blade es un motor de plantillas que se incluye con Laravel. Blade no restringe el uso código PHP en las plantillas. Todas las plantillas de Blade se compilan en código PHP puro y se almacenan en caché hasta que sean modificadas. Además, se puede implementar código PHP sin necesidad de utilizar sus etiquetas. Gracias a Blade, se puede insertar lógica simple directamente sobre HTML.

Además, para el front-end, se utiliza JavaScript puro, que permite al cliente realizar algunos cambios referentes a las plantillas; y CSS puro, que permite dar estilos personalizados a nuestra aplicación. También se utiliza Bootstrap, que es un framework multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web; y Tailwind, que es un framework de CSS de código abierto para el diseño de páginas web.

Para la base de datos, se utiliza MySQL, que es un sistema relacional de gestión de bbdd de código abierto.

También se utiliza xampp para proporcionar un entorno servidor local para el desarrollo.

A continuación, se muestra el diagrama de clases:



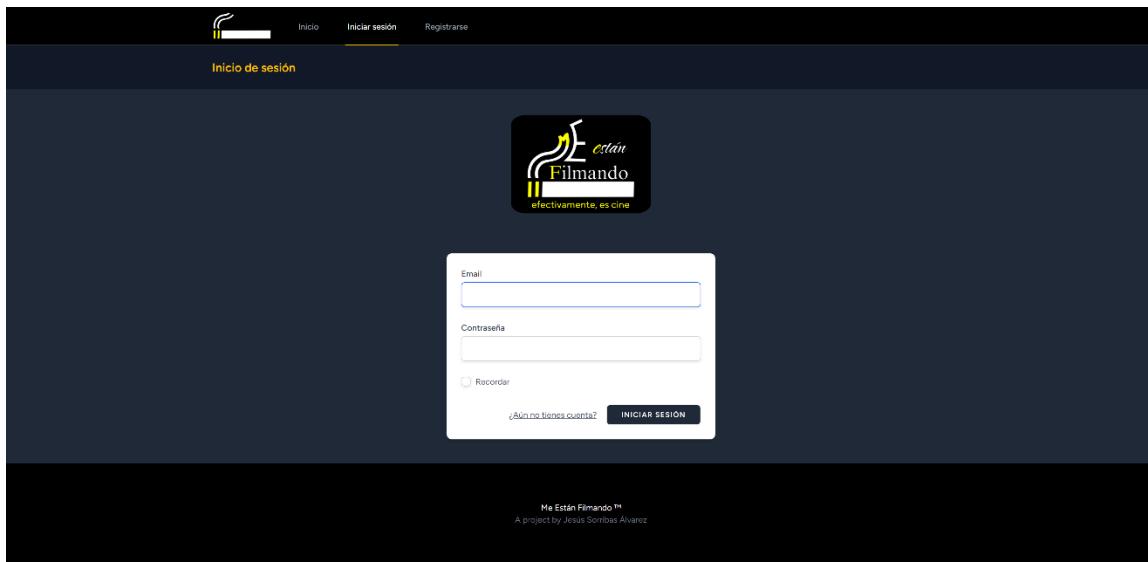
Si bien es cierto que hay tablas que se acaban descartando (véase la tabla “suscriptions”, la tabla “user_platforms” o las tablas intermedias con “awards”), la gran dimensión del entramado de clases nos da mucha variedad de interacción entre tablas una vez dentro de la aplicación.

Vistas

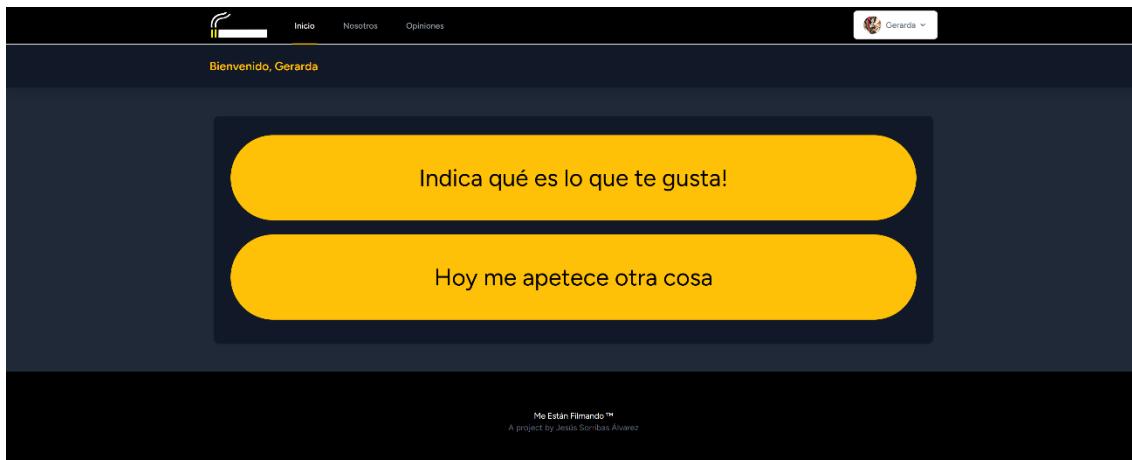
Bienvenida:

Registro:

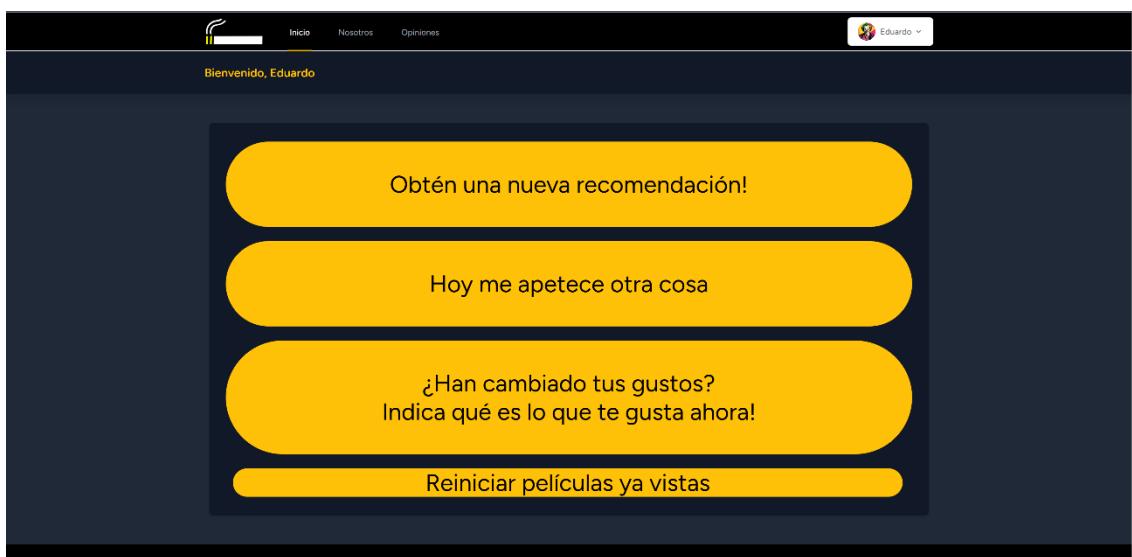
Jesús Sorribas Álvarez
CURSO: 2024-2025
PROYECTO
Inicio de sesión:



Inicio de alguien que todavía no ha indicado sus gustos:



Inicio de alguien que ya ha indicado que es lo que le gusta:



Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Vista de administrador, con una sub barra de navegación:

The screenshot shows a dark-themed web application interface. At the top, there's a navigation bar with links like 'Inicio', 'Nosotros', 'Opiniones', 'Usuarios', 'Películas', 'Directores', 'Actores', 'Géneros', 'Plataformas', 'Avatares', and 'Comentarios'. A user profile 'El Admin' is visible. Below the navigation, a yellow callout box says 'Obtén una nueva recomendación!'. Another yellow callout box below it says 'Hoy me apetece otra cosa'. A third yellow callout box further down says '¿Han cambiado tus gustos? Indica qué es lo que te gusta ahora!'. At the bottom of the main content area, there's a yellow bar with the text 'Reiniciar películas ya vistas'.

Vista sobre ‘nosotros’:

The screenshot shows the 'Sobre nosotros' (About Us) page. The top navigation bar includes 'Inicio', 'Nosotros', and 'Opiniones', with a user profile 'Gerardo'. The main content area contains three large text blocks, each starting with 'Lorem ipsum dolor sit amet...'. Below these blocks is a link 'Inicio' and at the bottom, a footer note 'Me Están Filmando™' followed by 'A project by Jesús Sorribas Álvarez'.

Vistas del formulario:

The screenshot shows a survey form titled 'Indica qué es lo que te gusta!'. It contains several questions with multiple choice answers:

- ¿Te gusta reírte cuando ves una película?
 - Sí
 - No
- ¿Te gusta que una película te convenga?
 - Sí
 - No
- ¿Te gustan las historias de acción?
 - Sí
 - No
- ¿Te gustan las historias de amor?
 - Sí
 - No
- ¿Te gustan las aventuras?
 - Sí
 - No
- ¿Te gusta adentrarte en mundos de fantasía?
 - Sí
 - No

 There are also 'Cancelar' and 'Siguiente' buttons at the bottom.

Indica qué es lo que te gusta!

¿Te gustan las historias de crímenes?
 Sí
 No

¿Querrías ver películas bíblicas?
 Sí
 No

¿Te gustan los westerns?
 Sí
 No

¿Te gustan los musicales?
 Sí
 No

¿Te gustan las películas que plantean debates filosóficos?
 Sí
 No

¿Te gustan las películas que tratan algún tema deportivo?
 Sí
 No

Cancelar

Indica qué es lo que te gusta!

¿Hay alguna época que no te guste? No escoges ninguna si te da igual la época en la que se base el filme.
 Antigua y anterior
 Medieval
 Moderna
 Contemporánea
 Futurista
 Atemporal o incierto

¿Hay algún director al que prefieras pasar por alto?

¿Hay algún actor o alguna actriz que preferirías no ver?

Guardar

Cancelar

Vista de la recomendación en sí:

Indica qué es lo que te gusta!

Aquí tienes tu recomendación. Esperamos que te guste!

Título: Stalker

Título original: Stalker

Duración: 163 minutos

Año: 1979

Director: Andrei Tarkovsky

Actores: -

Géneros: Internacional, Ciencia Ficción, Aventuras, Filosófico, Clásico, Contemporáneo

Dónde ver? Filmin

Aceptar y volver Ver en otro momento. Siguiente Ya la he visto!

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Vista de las películas de las que se pude dejar opinión:

Malditos Bastardos
La La Land
El Gran Lebowski
Babylon
El Club de la Lucha
OldBoy
La Vida es Bella
Abre los Ojos
Saga Torrente
Amanece, que no es poco

Vista de los comentarios referidos a la película seleccionada:

Eduardo (1) Y por lo visto hay gente a la que le gusta la película. Estoy rodeado de enfermos.
05/12/2024

Gerarda (1) Me ha impactado. El final es tan sorprendente como grotesco. La sensación de estupefacción ante la maldad del villano me aterra.
05/12/2024

Eduardo (1) Menuda mierda. Me dan arcadas y todo cada vez que pienso en lo que acabo de ver. Repugnante. Sólo sé que quien haya dirigido esto es el hijo de puta más grande de la tierra.
05/12/2024

El Admin (1) Fantástica. Increíble alegoría a la venganza a través de un hombre cuya vida ha sido arrebatada e intenta recompensar pedazo a pedazo.
05/12/2024

(Disculpen el lenguaje soez, es para demostrar como filtrar estas palabras y eliminar el comentario siendo administrador)

Jesús Sorribas Álvarez
CURSO: 2024-2025
PROYECTO
Vistas de edición de usuario:

Me Están Filmando™
A project by Jesús Sorribas Álvarez

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Vista de administrador de una tabla:

Filmes

Crear nuevo filme

ID	Título	Título original	Director	Duración (min.)	Año	Acciones
1	Malditos Bastardos	Inglourious Basterds	Quentin Tarantino	153	2009	
2	La La Land	La La Land	Damien Chazelle	128	2016	
3	El Gran Lebowski	The Big Lebowski	Ethan & Joel Coen	117	1998	
4	Babylon	Babylon	Damien Chazelle	189	2022	
5	El Club de la Lucha	Fight Club	David Fincher	139	1999	
6	OldBoy	OldBoy	Park Chan-wook	119	2003	
7	La Vida es Bella	La Vita e Bella	Roberto Benigni	116	1997	
8	Abre los Ojos	Abre los Ojos	Alejandro Amenábar	117	1997	
9	Saga Torrente	Torrente	Santiago Segura	491	1998	
10	Amanece, que no es poco	Amanece, que no es poco	José Luis Cuerda	110	1989	

Showing 1 to 10 of 25 results

Vista de administrador para añadir una nueva fila a una tabla:

Filmes

Crear nuevo filme

Nuevo filme

Título

Título original

Duración (minutos)

Año

¿Se encuentra en cines?

No

Director

Géneros

Actores

Plataformas

Cancelar Guardar

Vista de administrador para editar una fila de una tabla:

Directores

Crear nuevo director

Modificar los datos del director 1

Nombre

Steven Spielberg

País

Estados Unidos

Fecha

06/12/2024

Cancelar Guardar

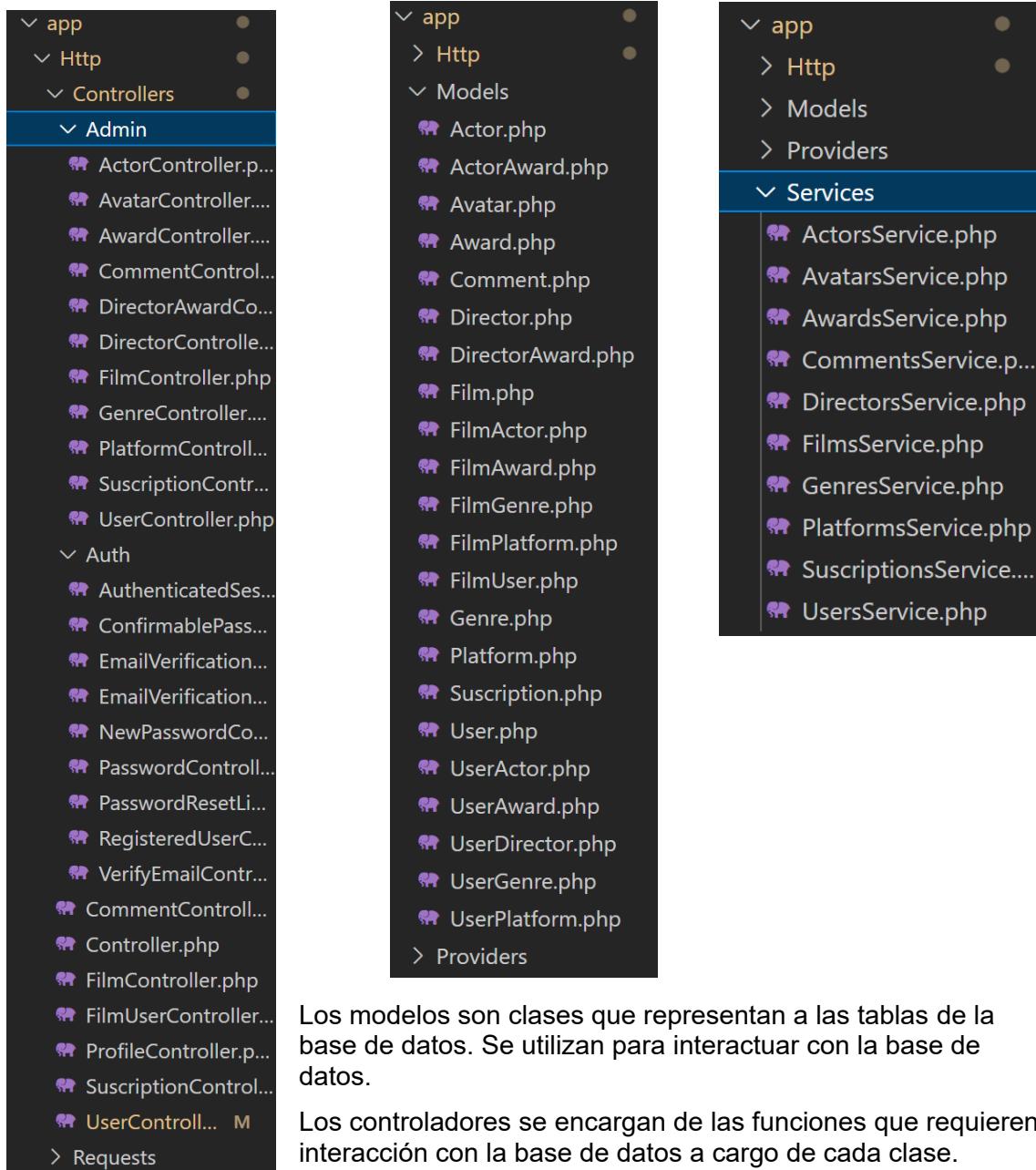
Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Lógica

La parte lógica de la aplicación se estructura de la siguiente manera:



Los modelos son clases que representan a las tablas de la base de datos. Se utilizan para interactuar con la base de datos.

Los controladores se encargan de las funciones que requieren interacción con la base de datos a cargo de cada clase.

Los servicios son la parte lógica que se encarga de realizar funciones más sencillas.

Jesús Sorribas Álvarez
 CURSO: 2024-2025
 PROYECTO
Implementación

Para la parte visual, todas las vistas utilizan la plantilla app.blade.php:

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()>getLocale()) }}>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="csrf-token" content="{{ csrf_token() }}>
    <link rel="icon" type="image/png" href="/pictures/MeEstánFilmando3.png"/>

    <title>Me Están Filmando</title>

    <!-- Fonts -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link href="https://fonts.googleapis.com/css?family=figtree:400,500,600&display=swap" rel="stylesheet" />

    <!-- Styles -->
    <link rel="stylesheet" href="/css/bootstrap.min.css" />
    <link href="{{ URL::asset('/assets/css/icons.min.css') }}" rel="stylesheet" type="text/css" />
    <link href="{{ URL::asset('/assets/css/app.min.css') }}" id="app-style" rel="stylesheet" type="text/css" />
    <style>
      .body {
        height: 100%;
        display: flex;
        flex-direction: column;
      }

      .body > * {
        flex-shrink: 0;
      }

      .footer {
        flex-grow: 1;
      }
    </style>

    <!-- Scripts -->
    @vite(['resources/css/app.css', 'resources/js/app.js'])
  </head>
  <body class="font-sans antialiased">
    <div class="min-h-screen bg-gray-800 body">
      <!-- Page Navigation bar -->
      @include('layouts.navigation')

      <!-- Admin Navigation bar -->
      @auth
        @if(Auth::user()>admin === 1)
          @include('layouts.admin-nav')
        @endif
      @endauth

      <!-- Page Heading -->
      @isset($header)
        <header class="bg-gray-900 shadow">
          <div class="max-w-7xl mx-auto py-6 px-4 sm:px-6 lg:px-8">
            {{ $header }}
          </div>
        </header>
      @endisset

      <!-- Page Content -->
      <main>
        {{ $slot }}
      </main>

      <!-- Page Footer -->
      @include('layouts.footer')
    </div>
  </body>
</html>
```

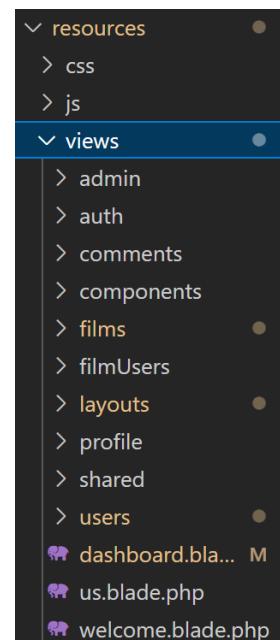
Utilizando más vistas a modo de componentes, modularizando y optimizando el código. A continuación, un ejemplo:

```
resources > views > filmUsers > 🐾 show.blade.php
1  <x-app-layout>
2    <x-slot name="header">
3      <div class="flex">
4        <h2 class="font-semibold text-xl text-warning leading-tight flex">
5          | {{ __('Aquí tienes tu recomendación. Esperamos que te guste!') }}
6        </h2>
7      </div>
8    </x-slot>
9
10   <?php
11     $user = Auth::user();
12   ?>
13
14   <div class="py-12">
15     <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
16       <div class="bg-gray-900 overflow-hidden shadow-sm sm:rounded-lg">
17         <div class="p-6 text-white">
18           <div class="row">
19             @include('layouts.show-film')
20             @include('filmUsers.partials.show-buttons')
21           </div>
22         </div>
23       </div>
24     </div>
25   </div>
26 </x-app-layout>
```

Todas las vistas implementan el motor Blade.

Las vistas se organizan de la siguiente manera:

Dentro de los recursos, organizando cada vista en función de la clase o tabla a la que hacen referencia.



Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Para la parte lógica, tenemos los modelos:

```
app > Models > User.php > ...
1  <?php
2
3  namespace App\Models;
4
5  // use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Database\Eloquent\Relations\BelongsTo;
9  use Illuminate\Database\Eloquent\Relations\BelongsToMany;
10 use Illuminate\Notifications\Notifiable;
11
12 class User extends Authenticatable
13 {
14     use HasFactory, Notifiable;
15
16     /**
17      * The attributes that are mass assignable.
18      *
19      * @var array<int, string>
20      */
21     protected $fillable = [
22         'name',
23         'email',
24         'password',
25         'suscription',
26         'recomendation',
27         'admin',
28         'avatar_id',
29         'suscription_id'
30     ];
31
32     /**
33      * The attributes that should be hidden for serialization.
34      *
35      * @var array<int, string>
36      */
37     protected $hidden = [
38         'password',
39         'remember_token',
40     ];
41
42     /**
43      * Casting attributes
44      */
45     protected $casts = [
46         'created_at' => 'date'
47     ];

```

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Que representan a las tablas y sus relaciones:

```
app > Models > User.php > ...
62     /**
63      * Get the avatar that owns the user.
64      */
65     public function avatar(): BelongsTo
66     {
67         return $this->belongsTo(Avatar::class);
68     }
69
70     /**
71      * Get the suscription that owns the user.
72      */
73     public function suscription(): BelongsTo
74     {
75         return $this->belongsTo(Suscription::class);
76     }
77
78     /**
79      * The cards that belong to the user.
80      */
81     public function cards(): BelongsToMany
82     {
83         return $this->belongsToMany(Card::class);
84     }
85
86     /**
87      * The films that belong to the user.
88      */
89     public function films(): BelongsToMany
90     {
91         return $this->belongsToMany(Film::class, 'film_users');
92     }
93
94     /**
95      * The platforms that belong to the user.
96      */
97     public function platforms(): BelongsToMany
98     {
99         return $this->belongsToMany(Platform::class, 'user_platforms');
100    }
101
102    /**
103     * The genres that belong to the user.
104     */
105    public function genres(): BelongsToMany
106    {
107        return $this->belongsToMany(Genre::class, 'user_genres');
108    }
```

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Y los controladores con las funciones que interactúan con la base de datos:

```
app > Http > Controllers > Admin >  ActorController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Models\Actor;
6  use App\Http\Controllers\Controller;
7  use App\Services\ActorsService;
8  use Illuminate\Http\Request;
9
10 class ActorController extends Controller
11 {
12     private $actorsService;
13
14     public function __construct(ActorsService $actorsService)
15     {
16         $this->actorsService = $actorsService;
17     }
18
19     /**
20      * Display a listing of the resource.
21      */
22     public function index(Request $request)
23     {
24         $actors = $this->actorsService->search($request)->paginate($request->get('per_page', 10));
25         return view('admin.actors.index', compact('actors', 'request'));
26     }
27
28     /**
29      * Show the form for creating a new resource.
30      */
31     public function create()
32     {
33         $actor = new Actor();
34         return view('admin.actors.create', compact('actor'));
35     }
36
37     /**
38      * Store a newly created resource in storage.
39      */
40     public function store(Request $request)
41     {
42         $actor = Actor::create($request->all());
43         $actor->save();
44
45         return to_route('admin.actors.index');
46     }

```

```
48     /**
49      * Display the specified resource.
50      */
51     public function show(Actor $actor)
52     {
53         //
54     }
55
56     /**
57      * Show the form for editing the specified resource.
58      */
59     public function edit(Actor $actor)
60     {
61         return view('admin.actors.edit', compact('actor'));
62     }
63
64     /**
65      * Update the specified resource in storage.
66      */
67     public function update(Request $request, Actor $actor)
68     {
69         $actor->update($request->all());
70
71         return to_route('admin.actors.index');
72     }
73
74     /**
75      * Remove the specified resource from storage.
76      */
77     public function destroy(Actor $actor)
78     {
79         $actor->delete();
80         return to_route('admin.actors.index');
81     }
82 }
83
```

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Despliegue

Para el despliegue de la aplicación en nuestro ordenador vamos a necesitar seguir una serie de pasos.

Lo primero será descargarxampp si no lo tenemos descargado ya:

Home / Browse Open Source / Database / Database Engines/Servers / XAMPP / Files

XAMPP Files

An easy to install Apache distribution containing

Brought to you by: [beltranrueda](#), [bitnami](#), [koswall](#)

Summary **Files** Reviews

Download Latest Version
[xampp-windows-x64-8.2.12-0-VS16-installer.exe \(157.6 MB\)](#)

Descargamos la última versión.

A continuación, descargamos Composer:

getcomposer.org/download/

Home | Getting Started | Download | Documentation | Browse Packages

Download Composer Latest: v2.8.3

Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run `Composer-Setup.exe` - it will install the latest composer version whenever it is executed.

Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

Y al ejecutar el asistente de instalación, lo hacemos de este modo:

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

No activamos el modo desarrollador:

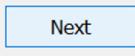
 Composer Setup**Installation Options**

Choose your installation type.

Setup will install Composer to a fixed location for all users. This includes a Control Panel uninstaller and is the recommended option. Click Next to use it.

 Developer mode

Take control and just install Composer. An uninstaller will not be included.

 Next

Cancel

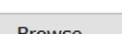
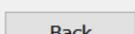
A la siguiente ruta, que será por defecto:

 Composer Setup**Settings Check**

We need to check your PHP and other settings.

Choose the command-line PHP you want to use:

C:\xampp\php\php.exe

 Browse... Add this PHP to your path? Back

Next

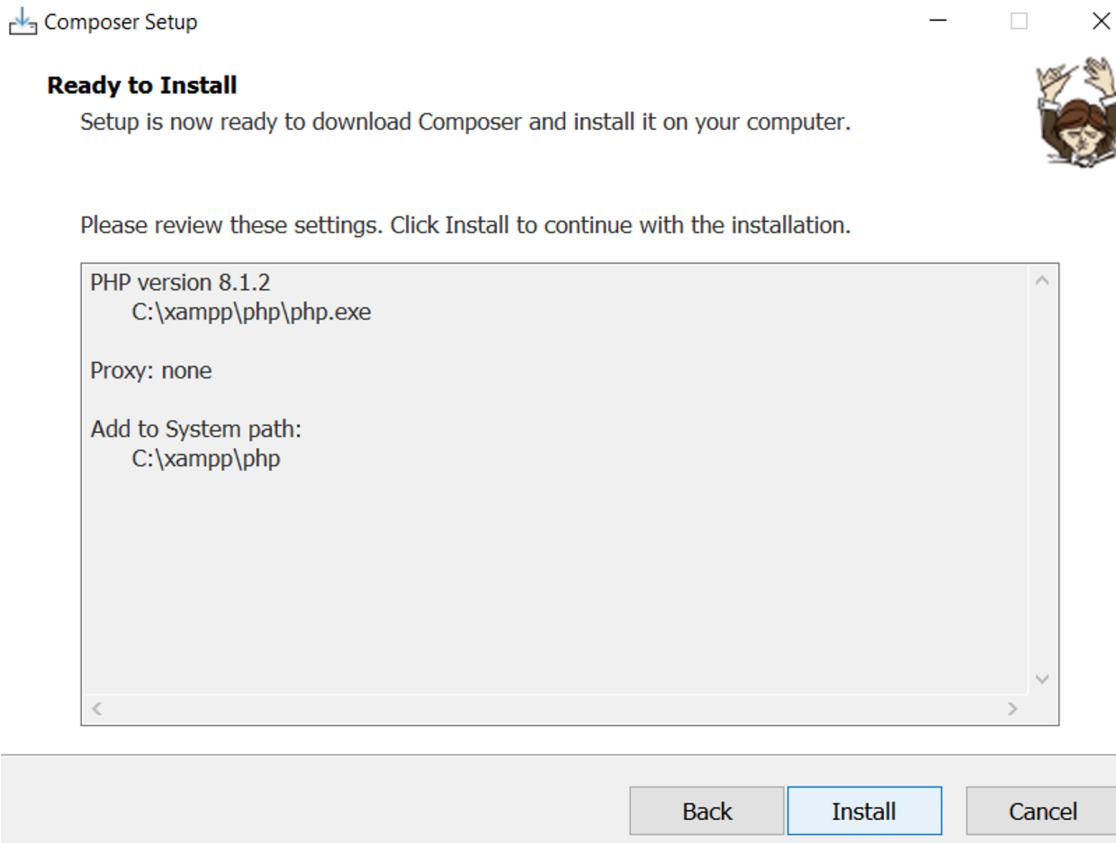
Cancel

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Y sin proxy. Entonces le damos a descarga:



Para comprobar que se instaló Composer correctamente, vamos a la consola y ejecutamos lo siguiente:

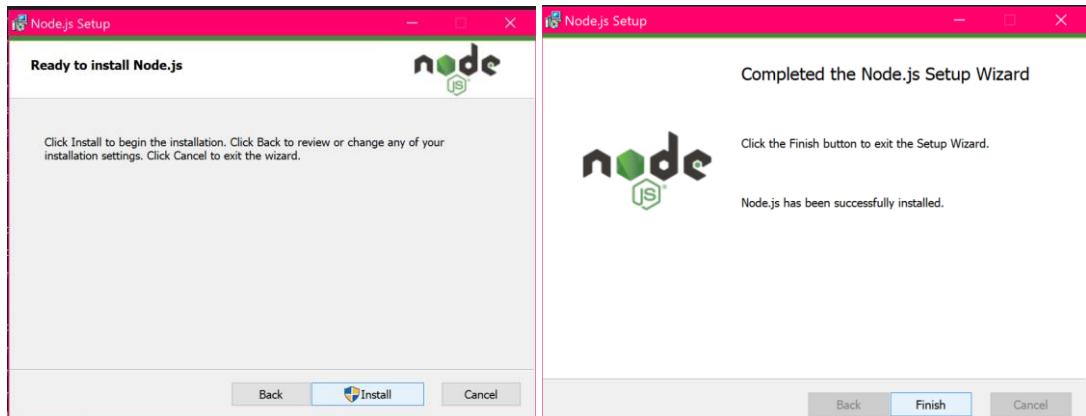
Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

A continuación, descargamos node.js:

Le damos a next en todas las pestañas del asistente de instalación y lo instalamos:



Puede que se abra powershell para seguir la instalación. Esperamos a que acabe y luego lo cerramos.

Ahora, por consola, nos posicionamos en C:\xampp\htdocs y hacemos git init:

```
PS C:\xampp\htdocs> git init
Initialized empty Git repository in C:/xampp/htdocs/.git/
```

Vamos al enlace del repositorio en git y lo copiamos:

Jesús Sorribas Álvarez

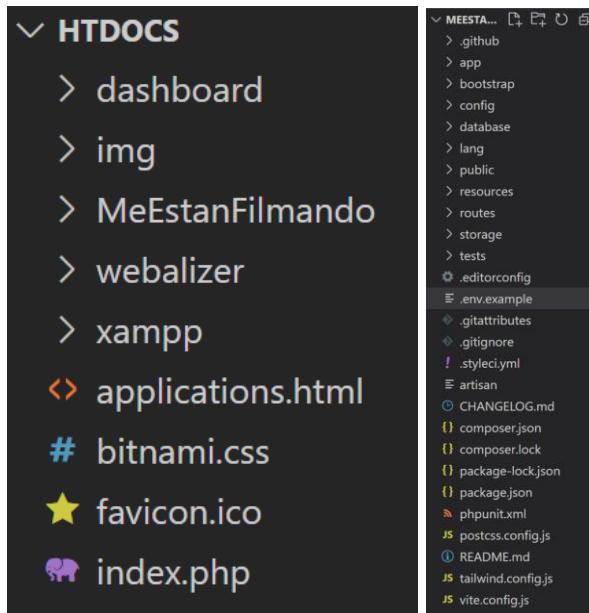
CURSO: 2024-2025

PROYECTO

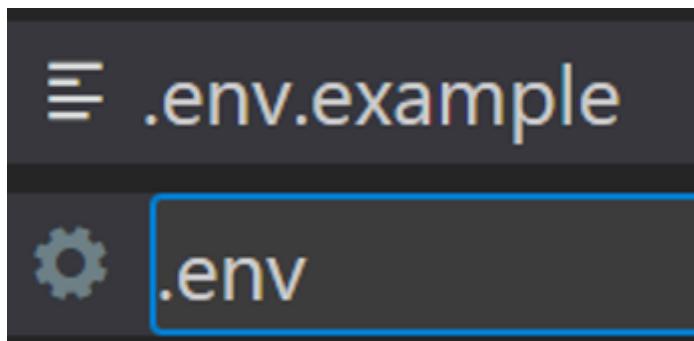
Y lo clonamos dentro de la carpeta:

```
PS C:\xampp\htdocs> git clone https://github.com/JeSoAl/MeEstanFilmando.git
Cloning into 'MeEstanFilmando'...
remote: Enumerating objects: 901, done.
remote: Total 901 (delta 0), reused 0 (delta 0), pack-reused 901 (from 1)
Receiving objects: 100% (901/901), 192.72 MiB | 5.65 MiB/s, done.
Resolving deltas: 100% (445/445), done.
Updating files: 100% (446/446), done.
```

Entonces ya tenemos el sistema de carpetas del proyecto dentro de htdocs:



Vamos al archivo .env.example y le quitamos el '.example' del nombre:



Dejándolo nombrado como .env

Nos aseguramos que el puerto referido en este archivo sea el mismo que el de nuestro servidor:

```
22 DB_CONNECTION=mysql
23 DB_HOST=localhost
24 DB_PORT=3306
25 DB_DATABASE=MeEstanFilmando
26 DB_USERNAME=root
27 DB_PASSWORD=
```

Se puede comprobar en los archivos php.ini y my.ini:

Service	Module	PID(s)	Port(s)	Actions
	Apache	11388 35588	80, 443	Stop Admin Config Logs Shell
	MySQL	24724	3307	Stop Admin Config Apache (httpd.conf) Apache (httpd-ssl.conf) Apache (httpd-xampp.conf)
	FileZilla			Start Admin Config
	Mercury			Start Admin Config PHP (php.ini)

Service	Module	PID(s)	Port(s)	Actions
	Apache	11388 35588	80, 443	Stop Admin Config Logs Shell
	MySQL	24724	3307	Stop Admin Config Logs Explorer my.ini
	FileZilla			Start Admin Config

Entonces, posicionándonos en la carpeta descargada, ejecutamos el siguiente comando:

```
PS C:\xampp\htdocs\MeEstanFilmando> composer update
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 2 updates, 0 removals
- Upgrading monolog/monolog (3.8.0 => 3.8.1)
- Upgrading sebastian/exporter (6.1.3 => 6.3.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 107 installs, 0 updates, 0 removals
Failed to download doctrine/inflector from dist: The zip extension and unzip/7z commands are both missing, skipping.
The php.ini used by your command-line PHP is: C:\xampp\php\php.ini
```

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Puede dar algún error como el resaltado en amarillo. No importa, esperamos a que acabe y continuamos.

Luego hacemos:

```
PS C:\xampp\htdocs\MeEstanFilmando> php artisan key:generate
```

Y:

```
PS C:\xampp\htdocs\MeEstanFilmando> php artisan config:cache
```

```
INFO Configuration cached successfully.
```

Lo siguiente será ejecutar PowerShell como administrador y ponemos lo siguiente:

```
PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted
Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkId=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] No [T] No a todo [U] Suspender [?] Ayuda (el valor predeterminado es "N"): s
PS C:\Windows\system32>
```

Le damos a Sí (s).

Ejecutamos ‘npm install’ y a continuación ‘npm run build’:

```
PS C:\xampp\htdocs\MeEstanFilmando> npm install
```

```
PS C:\xampp\htdocs\MeEstanFilmando> npm run build
```

```
> build
> vite build
```

```
vite v5.2.13 building for production...
```

Y ya tendríamos el proyecto listo para ser iniciado.

Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Ahora, iniciamos Apache y MySQL desde xampp:

The screenshot shows the XAMPP Control Panel interface. On the left, there's a sidebar titled "Modules" with icons for Apache, MySQL, FileZilla, Mercury, and Tomcat. The main table lists services by Service (Apache, MySQL), Module, PID(s), Port(s), and Actions (Stop, Admin, Config, Logs). Apache is listed with PID 11388 and port 80, 443, with "Config" highlighted. MySQL is listed with PID 24724 and port 3307, also with "Config" highlighted. To the right of the table are several buttons: Config, Netstat, Shell, Explorer, Services, Help, and Quit.

Vamos a la siguiente ruta y hacemos click en phpMyAdmin:

The screenshot shows a browser window with the URL "localhost/dashboard/" in the address bar. The page has a dark header with the Apache Friends logo and navigation links for FAQs, HOW-TO Guides, PHPInfo, and phpMyAdmin. The "phpMyAdmin" link is highlighted in blue.

Creamos una base de datos con el nombre del proyecto (“MeEstanFilmando”):

The screenshot shows the phpMyAdmin interface for a MySQL server at 127.0.0.1. The left sidebar shows databases like information_schema, mysql, performance_schema, phpmyadmin, and test. The main area is titled "Bases de datos" and shows a "Crear base de datos" (Create database) form with the name "MeEstanFilmando" and character set "utf8mb4_general_ci". Below the form is a table listing existing databases: information_schema, mysql, performance_schema, phpmyadmin, and test. Each database row includes a checkbox, a "Cotejamiento" (Collation) column, and an "Acción" (Action) column with a "Seleccionar privilegios" (Select privileges) button.

E importamos la base de datos situada en la carpeta “database” del proyecto:

W10 SSD M2 (C:) > xampp > htdocs > MeEstanFilmando > database

Nueva carpeta

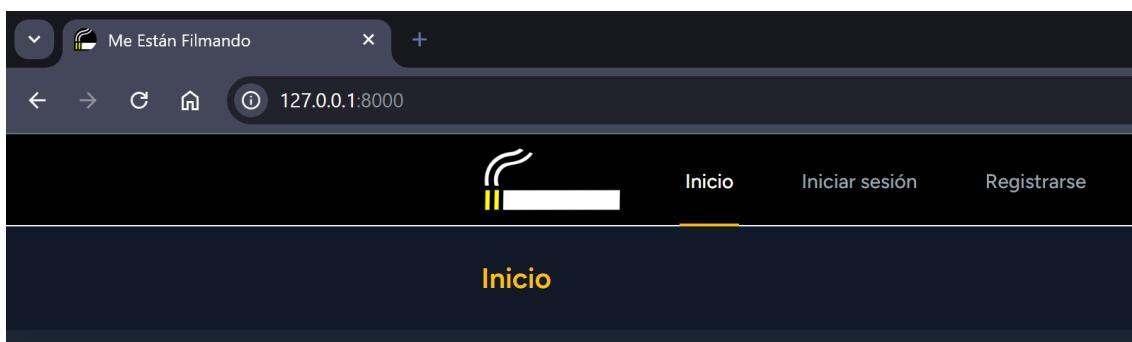
Nombre	Fecha de modificación
factories	05/12/2024 23:54
migrations	05/12/2024 23:54
seeders	05/12/2024 23:54
.gitignore	05/12/2024 23:54
meestanfilmando.sql	05/12/2024 23:54

The screenshot shows the phpMyAdmin interface for importing a SQL file into the 'meestanfilmando' database. The left sidebar lists databases: Nueva, information_schema, meestanfilmando, mysql, performance_schema, phpmyadmin, and test. The main area has a title 'Importando en la base de datos "meestanfilmando"' and a sub-section 'Archivo a importar:' with a file input field containing 'meestanfilmando.sql'. Below it are fields for 'Buscar en su ordenador' (Maximum: 40MB) and 'Seleccionar archivo' (set to 'meestanfilmando.sql'). There are also sections for 'También puede arrastrar un archivo en cualquier página.' and 'Conjunto de caracteres del archivo:' (set to 'utf-8').

Ahora solo nos queda ejecutar ‘php artisan serve’:

```
PS C:\xampp\htdocs\MeEstanFilmando> php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

Y copiamos el enlace indicado en un buscador para visualizar la aplicación:



Jesús Sorribas Álvarez

CURSO: 2024-2025

PROYECTO

Y listo, nuestro proyecto ya funciona.

Cabe la posibilidad de que nos de algún error (yo he probado en dos ordenadores distintos y funciona). En caso de error, copiamos ese error y buscamos soluciones por internet.

Para iniciar sesión como administrador, los datos son:

- Correo: admin@admin.com
- Contraseña: admin1234

Para iniciar sesión como cualquiera de los otros dos usuarios:

- Correo : eduardo@gmail.com ó gerarda@gmail.com
- Contraseñas respectivas: 87654321 y 12345678

También se puede registrar con un correo nuevo libremente.

Conclusión

La aplicación tiene mucho a mejorar, sobre todo en cuanto a estilos y apariencia; feedback al cliente (mensajes de error); y seguridad, ya que no se implementa más que lo que por defecto hace Laravel. Pero es un proyecto con mucho trabajo en backend, con mucha lógica bien optimizada y muchas tablas relacionales en la base de datos. Por diversas causas, se han descartado funcionalidades, que hacen que la aplicación tenga aún mucho margen de mejora y sea una buena propuesta a futuro.

Personalmente, creo que volcarme en el proyecto ha hecho que programe mejor y gane experiencia, formándome más en un lenguaje y un framework en específico.

Fuentes bibliográficas

Laravel: <https://laravel.com/>

Xampp: <https://sourceforge.net/projects/xampp/files/>

Composer: <https://getcomposer.org/download/>

Node.js: <https://nodejs.org/en/download/package-manager/current>

HTML: <https://www.w3schools.com/>

Bootstrap: <https://getbootstrap.com/>

Tailwind: <https://tailwindcss.com/>