

Memoria proyecto: PelisApreciadas

29-05-2024

C.I.F.P. "JUAN DE COLONIA"

Departamento de Informática

2º DAW

DIEGO MATÍAS

NABRDALIK FURLANI

**JOSÉ ANTONIO
MARTÍN CENDÓN**

Contenido

Contenido	1
Índice de figuras	3
Índice de tablas	4
1. Introducción.....	5
1.1. Descripción	5
1.2. Justificación	5
2. Planificación	5
2.1. Requisitos.....	5
2.1.1. Requisitos funcionales.....	5
2.1.2. Requisitos no funcionales.....	6
2.2. Recursos.....	8
2.2.1. Recursos hardware.....	8
2.2.2. Recursos software.....	8
2.3. Planificación temporal	9
2.4. Planificación económica	9
2.4.1. Costes en la planificación económica.....	9
2.4.2. Ingresos.....	10
2.4.3. Rentabilidad.....	12
3. Tecnologías.....	12
3.1. HTML	12
3.2. CSS	13
3.3. JavaScript.....	14
3.4. PHP	15
3.5. MySQL	16
3.6. Symfony.....	17
3.7. Twig.....	18
3.8. Bootstrap.....	19
3.9. Doctrine	20
3.10. DBAL.....	22
3.11. API	23
4. Desarrollo y secuenciación temporal.....	24

4.1.	Diseño	24
4.1.1.	Diagrama de aplicación	24
4.1.2.	Diagrama E/R (entidad/relación), modelo relacional y esquema de BBDD 26	
4.1.3.	Diseño de interfaces.....	29
4.2.	Pruebas.....	31
5.	Conclusiones finales	32
5.1.	Grado de cumplimiento de los requisitos fijados.....	32
5.2.	Propuestas de mejora o ampliaciones futuras	32
6.	Guía de usuario	33
7.	Referencias	40

Índice de figuras

FIGURA 3-1-1 LOGO HTML	12
FIGURA 3-2-1 LOGO CSS.....	13
FIGURA 3-3-1 LOGO JS	14
FIGURA 3-4-1 LOGO PHP	15
FIGURA 3-5-1 LOGO MYSQL	16
FIGURA 3-6-1 LOGO SYMFONY.....	17
FIGURA 3-7-1 LOGO TWIG	18
FIGURA 3-8-1 LOGO BOOTSTRAP	19
FIGURA 3-9-1 LOGO DOCTRINE.....	20
FIGURA 3-10-1 LOGO DBAL.....	22
FIGURA 3-11-1 IMAGEN API	23
FIGURA 4-1-1-1 IMAGEN MVC	24
FIGURA 4-1-1-2 IMAGEN ESTRUCTURA DE CARPETAS SYMFONY	25
Figura 4-1-1-3 Diagrama de casos de uso	25
FIGURA 4-1-2-1 IMAGEN DEL ESQUEMA DE ENTIDAD/RELACIÓN	26
FIGURA 4-1-2-2 IMAGEN DE LA BASE DE DATOS.....	27
FIGURA 4-1-3-1 IMAGEN PÁGINA INICIO	29
FIGURA 4-1-3-2 IMAGEN PÁGINA PELÍCULAS.....	30
FIGURA 4-1-3-3 IMAGEN PÁGINA LOG IN	31
Figura 6-1 Imagen página películas.....	33
Figura 6-2 Imagen página actor.....	34
Figura 6-3 Imagen página director	34
Figura 6-4 Leer críticas.....	35
Figura 6-5 Administrar lista	35
Figura 6-6 Puntuar películas.....	36
Figura 6-7 Añadir crítica	36
Figura 6-8 Imagen página películas administrador	37
Figura 6-9 Botón iniciar sesión	38
Figura 6-10 Enlace registrarse.....	38
Figura 6-11 Botón registrarse	38
Figura 6-12 Botón iniciar sesión	39

Figura 6-13 Iniciar sesión	39
----------------------------------	----

Índice de tablas

Tabla 2-3-1 Planificación temporal	9
Tabla 2-4-1- Presupuesto del proyecto	9
Tabla 2-4-3-1 Rentabilidad en 3 años.....	12
Tabla 4-2-1 Pruebas.....	31

1. Introducción

1.1. Descripción

En este proyecto voy a crear una página de reviews de películas en la que se podrá ver la información sobre películas además de eso se podrá valorar y dar un comentario sobre lo que opinas, se podrá ver información sobre los actores, directores. Como usuario podrás crear una lista de películas para ver más tarde.

1.2. Justificación

Durante el curso de desarrollo de aplicaciones web, reflexioné sobre cuál sería mi idea para el proyecto final y decidí crear una página de reviews de películas. Esta elección se fundamenta en mi pasión por el cine y en las ganas de aplicar los conocimientos adquiridos en un proyecto que represente un desafío estimulante. En este proyecto veo la oportunidad de explorar nuevas tecnologías y técnicas, como puede ser el uso de APIs para obtener información detallada sobre las diferentes películas y la implementación de funcionalidades avanzadas con el framework de php Symfony. Estas herramientas mejorarán la experiencia del usuario y contribuirán para el desarrollo de una plataforma interactiva y atractiva para los amantes del cine en línea. En resumen, la creación de esta página de reviews de películas no solo refleja mi pasión por el cine, sino también mi compromiso con continuar aprendiendo.

2. Planificación

2.1. Requisitos

2.1.1. Requisitos funcionales

- ✓ Los usuarios deben poder registrarse e iniciar sesión utilizando el correo electrónico. Se debería de poder hacer de forma segura.
- ✓ Página review de películas se podrá ver información de películas, actores, directores, puntuarlos,
- ✓ Cada película debe tener una página única con los detalles como pueden ser: el título, la sinopsis, los actores y director, la duración, los géneros y la fecha de estreno. Además, se debe de poder ver la información dada por los usuarios como pueden ser las puntuaciones y los comentarios.
- ✓ Los usuarios podrán calificar las películas sin la obligación de añadir antes una crítica o no.

- ✓ La calificación promedio de la película se actualizará en tiempo real y se mostrará en la página de la película.
- ✓ Los usuarios podrán gestionar listas de favoritos.

2.1.2. Requisitos no funcionales

Este informe detalla los requisitos no funcionales de una página web dedicada a proporcionar información sobre películas, actores y directores. La página está desarrollada utilizando Symfony, Bootstrap, Twig, MySQL, Doctrine y DBAL, y utiliza la API de TMDB para obtener información sobre películas y actores. Los requisitos no funcionales son cruciales para garantizar la calidad, el rendimiento y la usabilidad del sistema. A continuación, se describen estos requisitos en detalle.

Usabilidad

La usabilidad se refiere a la facilidad con la que los usuarios pueden aprender a usar el sistema y su eficiencia en la interacción con el mismo. Para nuestra página web, los siguientes puntos son esenciales:

1. **Interfaz Intuitiva:** La interfaz debe ser fácil de entender y navegar, utilizando Bootstrap para garantizar un diseño responsive y atractivo. Las etiquetas y menús deben ser claros y descriptivos.
2. **Navegación Consistente:** La estructura de navegación debe ser coherente en todas las páginas. Se debe utilizar Twig para crear plantillas uniformes.
3. **Tiempo de Aprendizaje Corto:** Los nuevos usuarios deben poder entender cómo usar el sitio web en pocos minutos. Las funciones deben estar donde los usuarios esperan encontrarlas.

Escalabilidad

La escalabilidad se refiere a la capacidad del sistema para manejar un aumento en la carga de trabajo o en la cantidad de datos sin comprometer el rendimiento.

1. **Base de Datos Relacional:** Utilizar MySQL y Doctrine para gestionar las relaciones entre datos de películas, actores y directores.
2. **Carga de Datos Inicial:** Utilizar la API de TMDB para obtener todos los datos necesarios y almacenarlos en la base de datos local.
3. **Optimización del Código:** Escribir código limpio y optimizado para que sea fácil de mantener y escalar.

Rendimiento

El rendimiento se refiere a la velocidad con la que el sistema responde a las acciones del usuario y a la eficiencia general del sistema.

1. **Optimización del Servidor:** Utilizar técnicas de caché para reducir el tiempo de respuesta del servidor.
2. **Carga Asíncrona:** Implementar la carga asíncrona de datos y contenido para mejorar la experiencia del usuario.

Accesibilidad

La accesibilidad garantiza que el sitio web sea usable por la mayor cantidad posible de personas, incluidas aquellas con discapacidades.

1. **Contraste de Colores:** Asegurarse de que el contraste de colores cumpla con las directrices WCAG.
2. **Navegación con Teclado:** Garantizar que todas las funcionalidades sean accesibles mediante el teclado.

Compatibilidad

La compatibilidad asegura que el sitio web funcione correctamente en diferentes navegadores y dispositivos.

1. **Compatibilidad entre Navegadores:** Probar el sitio en múltiples navegadores como Chrome, Opera, Firefox, Safari y Edge.
2. **Compatibilidad Móvil:** Asegurarse de que el diseño responsive funcione en dispositivos móviles y tabletas.

Seguridad

La seguridad protege el sistema y los datos del usuario contra amenazas y ataques.

1. **Protección Contra SQL Injection:** Utilizar consultas preparadas y ORM (Doctrine) para evitar inyecciones SQL.
2. **Autenticación y Autorización:** Implementar un sistema robusto de autenticación y autorización para proteger las áreas restringidas del sitio.
3. **Cifrado de Datos Sensibles:** Cifrar datos sensibles tanto en tránsito como en reposo.

2.2. Recursos

2.2.1. Recursos hardware

1. **Procesador:** Un procesador de al menos 2 GHz o superior (recomendable 3GHz).
Intel Core Series: Los procesadores de la serie Intel Core, como los Core i5 o Core i7 de las generaciones más recientes (por ejemplo, Core i5-11xxx o Core i7-11xxx), ofrecen un buen rendimiento para ejecutar PHP 7.4 en Windows 11.
AMD Ryzen Series: Los procesadores de la serie AMD Ryzen, como Ryzen 5 o Ryzen 7, también son excelentes opciones.
2. **Memoria RAM:** al menos entre 2 y 4 GB de RAM, recomendable 8 GB de RAM para un rendimiento óptimo (para que no entre en paginación en el disco duro, teniendo que utilizar el pagefile.sys y ralentizando el trabajo).
3. **Espacio en Disco:** Los proyectos Symfony y el código fuente en sí mismos no suelen ocupar mucho espacio, especialmente al principio. Sin embargo, a medida que agregue archivos de código, plantillas, activos estáticos y otros recursos, el tamaño del proyecto aumentará. Un proyecto Symfony típico puede ocupar desde unos pocos megabytes hasta varios gigabytes, dependiendo de su complejidad y la cantidad de archivos y dependencias que tenga. Sin embargo, necesitará espacio suficiente para instalar PHP, Composer y cualquier otra herramienta relacionada que utilice en su flujo de trabajo, además de los proyectos Symfony que desarrolle. Un tamaño mínimo de disco duro sería de 512 GB, recomendable entre 1 y 2 TeraBytes

2.2.2. Recursos software

Lista de recursos software

- ✓ **Sistema Operativo:** Windows 10 o Windows 11.
- ✓ **PHP:** Symfony 6.3.12 es compatible con PHP 7.4 o superior. Asegúrese de tener PHP instalado y configurado correctamente en su sistema.
- ✓ **Composer:** Necesitará Composer para gestionar las dependencias del proyecto. Asegúrese de tener Composer instalado en su sistema.
- ✓ **Base de Datos (MySQL):** Symfony es compatible con varias bases de datos como MySQL, PostgreSQL, SQLite, etc. Deberá tener instalada y configurada la base de datos que desee utilizar.
- ✓ **Servidor Web (WAMP):** Puede utilizar el servidor web integrado de PHP o configurar su propio servidor web como Apache o Nginx.
- ✓ **Visual Studio Code** ya que, debido a su amplia gama de extensiones, integración con Git y otras herramientas de colaboración, y su interfaz personalizable que permite un flujo de trabajo eficiente. Lo hace una herramienta ideal para programadores.

2.3. Planificación temporal

Tabla 2-3-1 Planificación temporal

Descripción de la tarea	Fecha inicio	Fecha fin	Nº de horas
Análisis de la aplicación	25-03-2024	27-03-2024	8
Diseño inicial	27-03-2024	29-03-2024	4
Diagramación base de datos	28-03-2024	02-04-2024	6
Búsqueda de API y pruebas (análisis de requisitos)	28-03-2024	15-04-2024	6
Creación y manejo de base de datos (Desarrollo de la aplicación web)	08-04-2024	24-04-2024	10
Investigación Symfony, Twig, Doctrine	12-04-2024	24-05-2024	6
Codificación de backend (Desarrollo de la aplicación web)	16-04-2024	15-05-2024	25
Maquetación de la página (diseño final)	18-04-2024	30-04-2024	7
Implementación	24-05-2024	29-05-2024	2
Pruebas	24-05-2024	28-05-2024	4
TOTAL HORAS			78

2.4. Planificación económica

2.4.1. Costes en la planificación económica

Tabla 2-4-1- Presupuesto del proyecto

DESCRIPCIÓN DE LA TAREA	PRECIO (€)
Horas de codificación (30€/hora)	2000€
Precio del equipo necesario	1708€
Precio del host virtual	250€/mes
Publicidad	700€/mes
Mantenimiento 6 horas semanales	600€/mes
TOTAL INICIAL (€)	3708€
TOTAL MANTENIMIENTO MENSUAL	1550€
TOTAL 3 AÑOS (TOTAL INICIAL + TOTAL MANTENIMIENTO MENSUAL)	59.508€

2.4.2. Ingresos

Supuestos Iniciales

1. **Tráfico Web:** 15,000 visitas mensuales.
2. **Tasa de Conversión:**
 - **Suscripciones:** 1% de los visitantes se convierten en suscriptores.
 - **Publicidad:** Tasa de clics (CTR) del 0.5% y costo por clic (CPC) promedio de 0.10€.
 - **Marketing de Afiliados:** Tasa de conversión del 1% y comisión promedio de 5€ por venta.
 - **Donaciones:** 0.1% de los visitantes realizan donaciones de 5€ cada una.
 - **Venta de Productos (merchandising):** Tasa de conversión del 0.5% y valor promedio de compra de 20€.

Ingresos por Mes

1. **Suscripciones:**
 - Visitas mensuales: 15,000
 - Tasa de conversión: 1% (150 suscriptores)
 - Precio de suscripción mensual: 5€
 - Ingresos mensuales: $150 * 5€ = 750€$
2. **Publicidad:**
 - Visitas mensuales: 15,000
 - CTR: 0.5% (75 clics)
 - CPC: 0.10€
 - Ingresos mensuales: $75 * 0.10€ = 7.50€$
3. **Marketing de Afiliados:**
 - Visitas mensuales: 15,000
 - Tasa de conversión: 1% (150 ventas)
 - Comisión promedio: 5€

- Ingresos mensuales: $150 * 5€ = 750€$

4. Donaciones:

- Visitas mensuales: 15,000
- Tasa de conversión: 0.1% (15 donaciones)
- Monto de donación: 5€
- Ingresos mensuales: $15 * 5€ = 75€$

5. Venta de Productos:

- Visitas mensuales: 15,000
- Tasa de conversión: 0.5% (75 compras)
- Valor promedio de compra: 20€
- Ingresos mensuales: $75 * 20€ = 1.500€$

Ingresos Totales por Mes

- **Ingresos por suscripciones:** 750€
- **Ingresos por publicidad:** 7.50€
- **Ingresos por marketing de afiliados:** 750€
- **Ingresos por donaciones:** 75€
- **Ingresos por venta de productos:** 1.500€

Ingresos Totales por Año

- **Ingresos anuales:** $3.082,50€ * 12 = 36.990€$

Ingresos Totales por 3 Años

- **Ingresos en 3 años:** $36.990€ * 3 = 110.970€$

Consideraciones Adicionales

- **Costos Operativos:** Hay que considerar los costos de servidores, marketing, personal, desarrollo y mantenimiento del sitio.
- **Variabilidad de Ingresos:** Los ingresos pueden variar según las campañas de marketing, estacionalidad, y cambios en las tasas de conversión.

Basado en estos supuestos, la rentabilidad estimada para los tres primeros años de una página web de cine con los ingresos y gastos calculados es de 51.462€. Esto indica que, con los datos actuales, el proyecto sería rentable en el periodo de tres años. Es importante ajustar las estrategias de ingresos y/o reducir los costos para mejorar aún más la rentabilidad del proyecto.

2.4.3. Rentabilidad

Tabla 2-4-3-1 Rentabilidad en 3 años

RENTABILIDAD EN 3 AÑOS	
Ingresos	110.970€
Gastos	59.508€
TOTAL RENTABILIDAD	51.462€

Basado en estos supuestos, la rentabilidad estimada para los tres primeros años de una página web de cine con los ingresos y gastos calculados es de -40.518€. Esto indica que, con los datos actuales, el proyecto no sería rentable en el periodo de tres años.

3. Tecnologías

3.1.HTML



FIGURA 3-1-1 LOGO HTML

HTML (HyperText Markup Language) es el lenguaje estándar para crear y estructurar contenido en la web. Desarrollado por Tim Berners-Lee en 1991, HTML ha evolucionado significativamente, convirtiéndose en la base fundamental del desarrollo web. Utiliza una serie de etiquetas para organizar texto, imágenes, enlaces y otros elementos, permitiendo a los navegadores web interpretar y mostrar el contenido de manera coherente.

HTML es esencialmente el esqueleto de una página web. A través de etiquetas, el HTML estructura el contenido de forma lógica y jerárquica. Es compatible con todos los navegadores modernos, lo que garantiza que las páginas web se vean y funcionen de manera consistente en diferentes dispositivos. Además, HTML es fundamental para la accesibilidad web, permitiendo que las personas con discapacidades naveguen y comprendan el contenido gracias a su estructura clara y semántica.

El uso de HTML no se limita a la mera presentación del contenido; también permite la integración de multimedia, como imágenes, videos y audio, enriqueciendo y mejorando la experiencia del usuario. Con el soporte continuo y la evolución del HTML5, se han añadido nuevas capacidades, como gráficos 2D/3D, almacenamiento local y una mejor integración de formularios interactivos.

3.2.CSS



FIGURA 3-2-1 LOGO CSS

CSS (Cascading Style Sheets) es un lenguaje utilizado para describir la presentación de documentos escritos en HTML o XML. Introducido por primera vez en 1996 por el W3C, CSS permite a los desarrolladores web separar el contenido de la estructura visual de una página web, facilitando el diseño y el mantenimiento.

CSS es fundamental para definir cómo se ven y se sienten las páginas web. Mediante el uso de selectores y reglas, los diseñadores pueden especificar estilos como colores, fuentes, márgenes, espaciados y la disposición general de los elementos. Por ejemplo, con CSS se puede cambiar el color del texto de un encabezado, ajustar el espaciado de un párrafo, o crear diseños complejos utilizando flexbox o grid.

La capacidad de CSS para controlar la presentación de múltiples páginas web a la vez es una de sus mayores fortalezas. Con una sola hoja de estilos, es posible aplicar un diseño uniforme a varias páginas de un sitio, simplificando considerablemente el proceso de actualización y mantenimiento. Además, CSS permite la creación de sitios web responsivos, que se adaptan a diferentes tamaños de pantalla y dispositivos, mejorando la accesibilidad y la experiencia del usuario.

3.3.JavaScript



FIGURA 3-3-1 LOGO JS

JavaScript es un lenguaje de programación dinámico y orientado a objetos que se utiliza principalmente para desarrollar contenido interactivo en las páginas web. Fue creado por Brendan Eich en 1995 mientras trabajaba en Netscape Communications Corporation y rápidamente se convirtió en una de las tecnologías fundamentales de la web, junto con HTML y CSS.

JavaScript es un lenguaje interpretado, lo que significa que se ejecuta directamente en el navegador web del usuario sin necesidad de compilación previa. Su mayor fortaleza radica en la capacidad de manipular el Document Object Model (DOM), lo que permite actualizar y modificar el contenido de una página web de manera dinámica y en tiempo real. Esto es esencial para crear experiencias de usuario interactivas, como la validación de formularios en tiempo real, la actualización de contenido sin recargar la página (mediante técnicas como AJAX) y la creación de aplicaciones de una sola página (SPA) utilizando frameworks como React, Angular y Vue.js.

Además de su uso en el frontend, JavaScript ha extendido su alcance al backend con la introducción de Node.js. Esta plataforma permite ejecutar JavaScript en el servidor, facilitando el desarrollo de aplicaciones completas utilizando un solo lenguaje de programación tanto en el cliente como en el servidor. Esto ha simplificado enormemente el flujo de trabajo de los desarrolladores y ha fomentado el uso de JavaScript en una amplia variedad de aplicaciones, desde servidores web hasta aplicaciones de red en tiempo real.

JavaScript sigue evolucionando gracias a la especificación ECMAScript (ES), que introduce nuevas características y mejoras al lenguaje de forma regular. La versión más reciente, ES2021, incluye mejoras significativas como nuevas funciones y optimizaciones que hacen el desarrollo más eficiente y poderoso.

3.4.PHP



FIGURA 3-4-1 LOGO PHP

PHP (Hypertext Preprocessor) es un lenguaje de scripting de código abierto que se utiliza principalmente para el desarrollo web del lado del servidor. Fue creado por Rasmus Lerdorf en 1994, convirtiéndose en uno de los lenguajes más populares para la creación de sitios web dinámicos y aplicaciones web.

PHP está diseñado para ser sencillo de aprender y usar. Su sintaxis se asemeja a la de C y Perl, lo que facilita su adopción por parte de desarrolladores familiarizados con estos lenguajes. PHP se ejecuta en el servidor, generando HTML que luego se envía al navegador del cliente. Esto permite la creación de contenido web dinámico que puede interactuar con bases de datos, gestionar sesiones de usuario y realizar diversas tareas del lado del servidor.

Uno de los mayores puntos fuertes de PHP es su integración con una amplia variedad de sistemas de gestión de bases de datos, especialmente MySQL, lo que lo convierte en una opción ideal para desarrollar aplicaciones web que requieren almacenamiento y recuperación de datos. Además, PHP es altamente extensible mediante el uso de numerosas bibliotecas y frameworks como Laravel, Symfony y CodeIgniter, que facilitan el desarrollo de aplicaciones web robustas y escalables.

PHP también es conocido por su comunidad activa y una gran cantidad de recursos disponibles, incluyendo documentación detallada, tutoriales y foros de soporte. Esta comunidad contribuye continuamente al desarrollo del lenguaje, asegurando que PHP se mantenga actualizado con las últimas tendencias y necesidades del desarrollo web.

A pesar de la competencia de otros lenguajes de scripting del lado del servidor como Python y Node.js, PHP sigue siendo una opción preferida por muchos desarrolladores y empresas debido a su flexibilidad, facilidad de uso y la madurez de su ecosistema.

3.5.MySQL

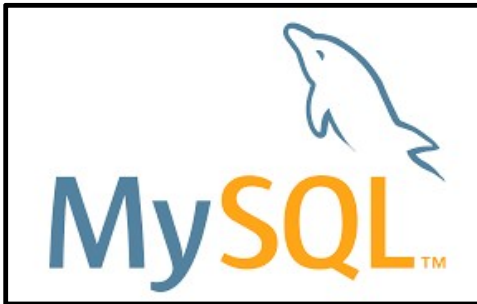


FIGURA 3-5-1 LOGO MYSQL

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, desarrollado inicialmente por MySQL AB en 1995. Actualmente es propiedad de Oracle Corporation. Es uno de los sistemas de bases de datos más populares en el mundo, utilizado para aplicaciones web y servicios en línea.

MySQL utiliza el lenguaje SQL (Structured Query Language) para gestionar y manipular bases de datos. Es conocido por su alta performance, fiabilidad y facilidad de uso. MySQL es la base de datos elegida para muchas aplicaciones web, gracias a su capacidad para manejar grandes cantidades de datos de manera eficiente y su compatibilidad con una amplia gama de sistemas operativos y plataformas, como Linux, Windows y macOS.

Uno de los aspectos más destacados de MySQL es su integración con diversas tecnologías web, especialmente PHP, formando la base de la pila LAMP (Linux, Apache, MySQL, PHP/Perl/Python). Esta combinación es fundamental para el desarrollo de sitios web dinámicos y aplicaciones de comercio electrónico. Además, MySQL ofrece características avanzadas como replicación, clustering y soporte para transacciones, lo que lo hace adecuado tanto para pequeñas aplicaciones como para grandes sistemas empresariales.

MySQL se mantiene en constante evolución, con actualizaciones regulares que mejoran su funcionalidad, seguridad y rendimiento. La comunidad de MySQL es extensa, proporcionando abundantes recursos, documentación y soporte para desarrolladores de todos los niveles.

3.6.Symfony



FIGURA 3-6-1 LOGO SYMFONY

Symfony es un framework de desarrollo web de código abierto escrito en PHP, diseñado para simplificar y acelerar el proceso de creación de aplicaciones web complejas y escalables. Fue creado por Fabien Potencier en 2005 y ha ganado una gran popularidad en la comunidad de desarrollo web.

Una de las características distintivas de Symfony es su adopción del patrón de arquitectura Modelo-Vista-Controlador (MVC). Este enfoque proporciona una clara separación de la lógica de negocio, la presentación y la gestión de datos, lo que resulta en una estructura organizada y modular que facilita el desarrollo de aplicaciones. Al seguir este patrón, Symfony permite a los desarrolladores trabajar de manera eficiente y colaborativa en proyectos de cualquier tamaño.

Además, Symfony se destaca por su enfoque en la reutilización de componentes. El framework ofrece una amplia variedad de componentes independientes y reutilizables que abarcan desde el manejo de formularios hasta la seguridad y la caché. Esta modularidad hace que Symfony sea altamente flexible y adaptable a una variedad de casos de uso y requisitos específicos, permitiendo a los desarrolladores construir aplicaciones personalizadas de manera eficiente y escalable.

Symfony también es conocido por su compromiso con la calidad del código y las mejores prácticas de desarrollo. El framework sigue estrictos estándares de codificación y proporciona herramientas integradas para realizar pruebas unitarias y funcionales, lo que garantiza la estabilidad y confiabilidad de las aplicaciones desarrolladas con Symfony.

En cuanto a la integración con otras tecnologías, Symfony se integra sin problemas con una variedad de herramientas y tecnologías populares en el ecosistema de PHP. Por ejemplo, es compatible con el ORM (Mapeo Objeto-Relacional) Doctrine, que simplifica la gestión de bases de datos relacionales en aplicaciones Symfony. También se puede combinar con Twig, un motor de plantillas que facilita la creación de interfaces de usuario dinámicas y atractivas.

3.7. Twig

Twig es un motor de plantillas flexible y potente diseñado específicamente para la creación de interfaces de usuario dinámicas y atractivas en aplicaciones web desarrolladas



en PHP. Desarrollado por Fabien Potencier, el mismo creador de Symfony, Twig es una opción popular entre los desarrolladores PHP debido a su sintaxis clara y legible, su rendimiento eficiente y su amplia gama de características.

FIGURA 3-7-1 LOGO TWIG

La sintaxis de Twig está diseñada para ser intuitiva y fácil de entender, lo que facilita a los desarrolladores escribir y mantener plantillas HTML de manera eficiente. Utilizando una combinación de etiquetas y variables, los desarrolladores pueden crear fácilmente plantillas que se integren sin problemas con el código PHP subyacente. Además, Twig ofrece una serie de características avanzadas, como la herencia de plantillas, los bloques de contenido y los filtros de salida, que permiten a los desarrolladores crear diseños flexibles y modulares para sus aplicaciones web.

Una de las características más destacadas de Twig es su capacidad para prevenir vulnerabilidades de seguridad comunes, como los ataques de inyección de código y XSS (Cross-Site Scripting). Twig automáticamente escapa las variables de salida por defecto, lo que ayuda a prevenir la ejecución de código malicioso en el navegador del usuario final. Esto hace que Twig sea una opción segura y confiable para la creación de plantillas en aplicaciones web PHP.

Además de su seguridad y facilidad de uso, Twig también ofrece un rendimiento excepcional. El motor de plantillas está diseñado para ser rápido y eficiente, lo que garantiza tiempos de renderizados rápidos incluso en aplicaciones web complejas con grandes cantidades de contenido dinámico. Twig utiliza un sistema de almacenamiento en caché inteligente para minimizar la carga en el servidor y optimizar el rendimiento de las aplicaciones web.

3.8.Bootstrap

Bootstrap es un popular framework de código abierto desarrollado por Twitter que facilita la creación de sitios web y aplicaciones web responsivos y visualmente atractivos.



Desde su lanzamiento en 2011, Bootstrap ha ganado una gran popularidad entre los desarrolladores web debido a su facilidad de uso, su amplia gama de componentes predefinidos y su capacidad para agilizar el proceso de desarrollo.

FIGURA 3-8-1 LOGO BOOTSTRAP

Bootstrap está basado en HTML, CSS y JavaScript, lo que lo hace compatible con una variedad de navegadores y dispositivos. Una de las características más destacadas de Bootstrap es su sistema de rejilla flexible, que permite a los desarrolladores crear diseños responsivos que se ajustan automáticamente al tamaño de la pantalla del dispositivo. Esto garantiza una experiencia de usuario consistente en dispositivos de escritorio, tabletas y teléfonos móviles.

Además de su sistema de rejilla, Bootstrap ofrece una amplia gama de componentes predefinidos, como botones, formularios, menús desplegados, carruseles y más. Estos componentes facilitan la creación de interfaces de usuario interactivas y funcionales sin necesidad de escribir mucho código personalizado. Además, Bootstrap incluye una variedad de estilos y temas prediseñados que los desarrolladores pueden utilizar para personalizar fácilmente el aspecto y la sensación de sus sitios web y aplicaciones.

Otra ventaja de Bootstrap es su comunidad activa y su amplia base de usuarios. La comunidad de Bootstrap está compuesta por miles de desarrolladores de todo el mundo que contribuyen con código, ejemplos y recursos para ayudar a otros a aprender y utilizar el framework de manera efectiva. Esto significa que los desarrolladores que utilizan Bootstrap tienen acceso a una gran cantidad de documentación, tutoriales y recursos en línea que pueden ayudarles a resolver problemas y mejorar sus habilidades.

3.9.Doctrine

Doctrine es una biblioteca de mapeo objeto-relacional (ORM) de código abierto para PHP, diseñada para facilitar y optimizar la interacción entre las aplicaciones PHP y las bases de datos relacionales, permitiendo a los desarrolladores trabajar con datos como si fueran objetos en lugar de utilizar consultas SQL directamente. Doctrine fue desarrollado inicialmente por Konsta Vesterinen y Roman Shtylman. Desde su creación, Doctrine ha sido adoptado ampliamente por la comunidad de desarrolladores PHP debido a su flexibilidad y capacidad para simplificar el manejo de datos en aplicaciones complejas.



FIGURA 3-9-1 LOGO DOCTRINE

Doctrine sigue el patrón de diseño Data Mapper, lo que permite a los desarrolladores mapear las tablas de la base de datos a objetos PHP de manera automática y viceversa. Esto desacopla la lógica de negocio de la lógica de acceso a datos, proporcionando una estructura más organizada y modular para el desarrollo de aplicaciones.

Una de las principales características de Doctrine es su enfoque en la abstracción de bases de datos. Utiliza el Doctrine Query Language (DQL), un lenguaje de consulta similar a SQL pero orientado a objetos, lo que facilita la escritura de consultas complejas de una manera más intuitiva y natural para los desarrolladores acostumbrados a la programación orientada a objetos. Además, Doctrine soporta múltiples tipos de bases de datos, permitiendo a los desarrolladores cambiar de motor de base de datos sin necesidad de reescribir el código de acceso a datos.

Doctrine se compone de varios componentes independientes y reutilizables que pueden integrarse según las necesidades del proyecto. Entre estos componentes destacan:

1. **Entity Manager:** El Entity Manager es el núcleo de Doctrine, encargado de gestionar el ciclo de vida de las entidades. Facilita operaciones como insertar, actualizar, eliminar y consultar registros en la base de datos, simplificando así el manejo de datos.
2. **DQL (Doctrine Query Language):** DQL es un lenguaje de consulta similar a SQL, pero diseñado para trabajar con objetos en lugar de tablas. Esto permite a los desarrolladores escribir consultas de una manera que se siente más natural dentro del contexto de la programación orientada a objetos.

3. **Migraciones:** El sistema de migraciones de Doctrine permite a los desarrolladores versionar el esquema de la base de datos y aplicar cambios de manera incremental. Esto es crucial para mantener la coherencia y la integridad del esquema a lo largo del ciclo de vida de una aplicación.
4. **Caching:** Doctrine soporta varios tipos de caché, incluyendo caché de consultas, resultados y metadatos. Esto mejora significativamente el rendimiento de la aplicación al reducir la carga sobre la base de datos.
5. **Extensions:** Doctrine es altamente extensible. Existen numerosas extensiones y paquetes adicionales que amplían su funcionalidad, ofreciendo comportamientos adicionales como versionado, localización y auditoría.

Doctrine se integra perfectamente con el framework Symfony, facilitando tanto el desarrollo como el mantenimiento del código.

Ventajas de Usar Doctrine:

1. **Abstracción de Base de Datos:** Facilita la portabilidad y el mantenimiento del código.
2. **Reducción de Código Repetitivo:** Reduce la cantidad de código SQL repetitivo.
3. **Integración con Symfony:** Facilita su adopción y uso en proyectos Symfony.
4. **Comunidad y Documentación:** Facilita encontrar soluciones a problemas comunes y aprender a utilizar sus características avanzadas.

Desventajas y Consideraciones:

1. **Curva de Aprendizaje:** Para los desarrolladores nuevos en ORM o en el concepto de mapeo objeto-relacional, Doctrine puede tener una curva de aprendizaje pronunciada.
2. **Sobrecarga de Rendimiento:** El uso de un ORM puede introducir una sobrecarga adicional en comparación con las consultas SQL escritas a mano, especialmente en aplicaciones de alto rendimiento.

3.10. DBAL

DBAL. Doctrine DBAL (Database Abstraction Layer) es una parte integral de la suite de herramientas de Doctrine y proporciona una capa de abstracción de bases de datos para



FIGURA 3-10-1 LOGO DBAL

aplicaciones PHP. Su principal objetivo es simplificar el acceso y la manipulación de datos en diferentes sistemas de gestión de bases de datos (DBMS), independientemente del proveedor específico de la base de datos.

Doctrine DBAL facilita la escritura de consultas SQL compatibles con varios DBMS, lo que permite a los desarrolladores crear aplicaciones PHP que funcionen sin problemas en diferentes entornos de bases de datos. Además de abstraer la sintaxis SQL, DBAL maneja de manera transparente las diferencias de comportamiento entre los diferentes DBMS, como la sintaxis de funciones y los tipos de datos específicos.

Una de las características más útiles de Doctrine DBAL es su manejo de la seguridad y la prevención de ataques de inyección SQL. DBAL utiliza consultas preparadas y vinculación de parámetros para proteger contra ataques de inyección SQL, lo que garantiza que las consultas SQL sean seguras y no vulnerables a manipulaciones maliciosas.

Además de proporcionar una capa de abstracción de bases de datos, Doctrine DBAL también ofrece otras funcionalidades útiles, como la gestión de conexiones de base de datos, la creación y ejecución de consultas SQL personalizadas, y la gestión de transacciones.

3.11. API

API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés) es un conjunto de definiciones y protocolos que permiten que dos aplicaciones se comuniquen entre sí. Actúa como un intermediario que permite que los desarrolladores utilicen funcionalidades de un software sin tener que entender o modificar su código fuente. Las API son esenciales en el desarrollo de software moderno, ya que facilitan la integración de diversas aplicaciones y servicios, permitiendo una mayor eficiencia y colaboración.

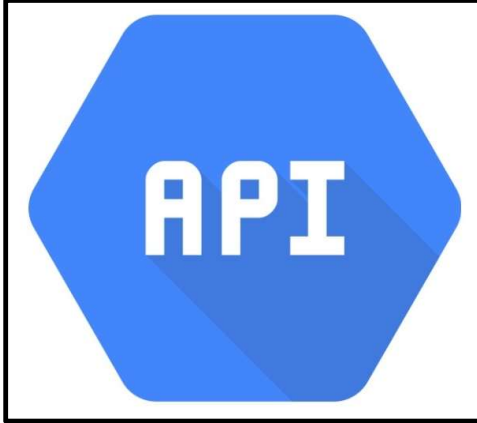


FIGURA 3-11-1 IMAGEN API

Las API han transformado la manera en que se desarrollan y distribuyen las aplicaciones de software. En lugar de construir todas las funcionalidades desde cero, los desarrolladores pueden aprovechar las capacidades ofrecidas por otras aplicaciones a través de sus API. Esto no solo acelera el proceso de desarrollo, sino que también asegura la consistencia y fiabilidad de las funcionalidades integradas.

Existen diferentes tipos de API según su uso y su acceso. Las API públicas están disponibles para cualquier desarrollador y son generalmente utilizadas para integrar servicios populares como redes sociales, pagos en línea o mapas. Las API privadas, por otro lado, son utilizadas internamente por una organización para conectar sus propios sistemas y servicios. Además, las API pueden ser locales (para uso en una única máquina) o basadas en la web (para uso a través de la red).

Una de las API más comunes y utilizadas hoy en día es la API REST (Transferencia de Estado Representacional, por sus siglas en inglés). REST es un estilo de arquitectura que utiliza HTTP y es fácil de usar, escalar y mantener. Las API RESTful son muy populares debido a su simplicidad y flexibilidad. Otro tipo común es la API SOAP (Protocolo Simple de Acceso a Objetos, por sus siglas en inglés), que es más complejo y utiliza XML para la mensajería.

Las API son cruciales para la economía digital, permitiendo la interoperabilidad entre diferentes sistemas y plataformas. Por ejemplo, las API de pago, como las de PayPal o Stripe, permiten que los sitios web de comercio electrónico procesen pagos de manera segura y eficiente. Las API de redes sociales permiten que las aplicaciones compartan contenido y obtengan información de plataformas como Facebook, Twitter o Instagram.

Además de facilitar la integración, las API permiten a los desarrolladores crear aplicaciones más robustas y con menos errores. Al utilizar una API probada y bien documentada, los desarrolladores pueden confiar en que las funcionalidades que integran funcionarán como se espera. Esto reduce significativamente el tiempo y el esfuerzo necesarios para desarrollar nuevas aplicaciones.

4. Desarrollo y secuenciación temporal

4.1. Diseño

4.1.1. Diagrama de aplicación

El Modelo-Vista-Controlador es un patrón que divide a la aplicación en tres niveles distintos, uno que representa a la interfaz gráfica o al cliente (Vista), otro que representa el tratamiento de datos (Modelo) y otro que se encarga de toda la lógica que se tiene que llevar a cabo por la aplicación (Controlador). Esto se hace para permitir una mayor portabilidad de una aplicación, e incluso facilitar su mantenimiento.

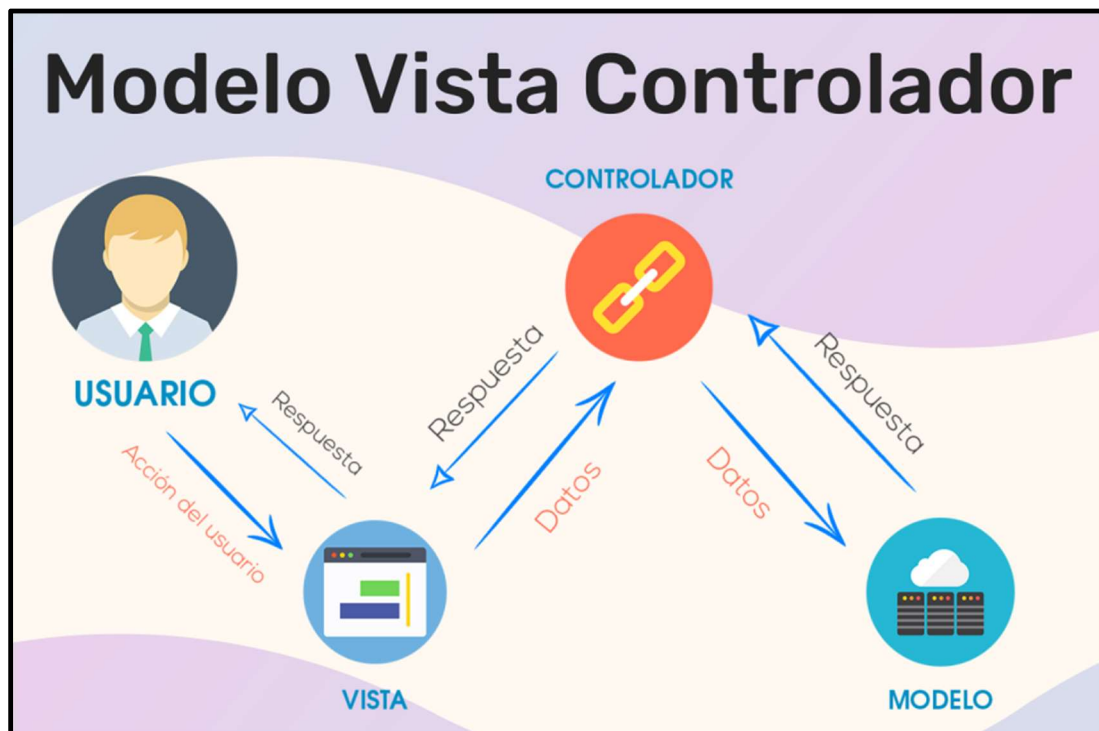
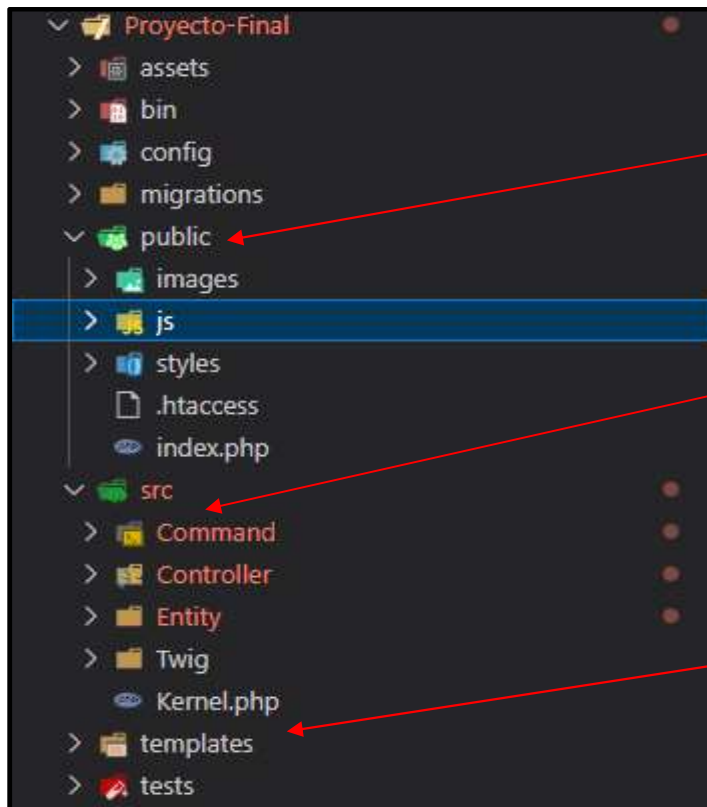


FIGURA 4-1-1-1 IMAGEN MVC



Contiene recursos de la página como imágenes, javascript, estilos.

Contiene la carpeta de los controladores (controller) y modelos (entity)

Es la carpeta donde se guardan las vistas

FIGURA 4-1-1-2 IMAGEN ESTRUCTURA DE CARPETAS SYMFONY

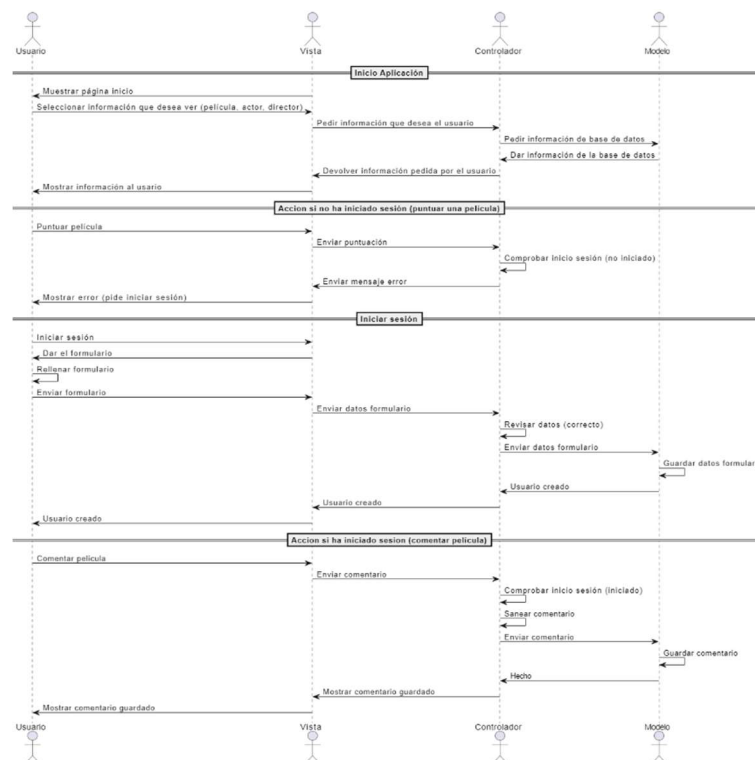


Figura 4-1-1-3 Diagrama de casos de uso

4.1.2. Diagrama E/R (entidad/relación), modelo relacional y esquema de BBDD

El esquema de la BBDD deber contar como mínimo con cuatro tablas.

Tener como referencia para este apartado al módulo de Bases de Datos

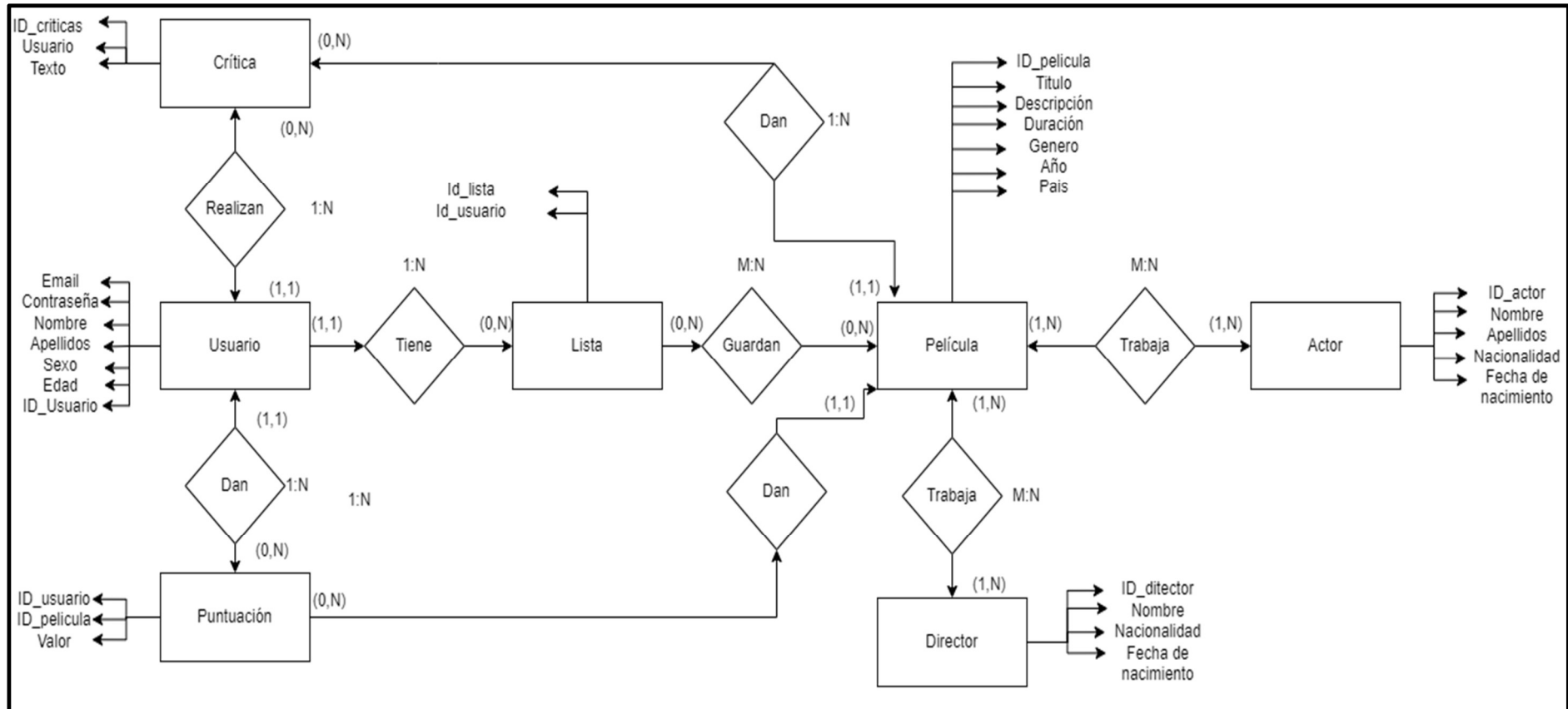


FIGURA 4-1-2-1 IMAGEN DEL ESQUEMA DE ENTIDAD/RELACIÓN

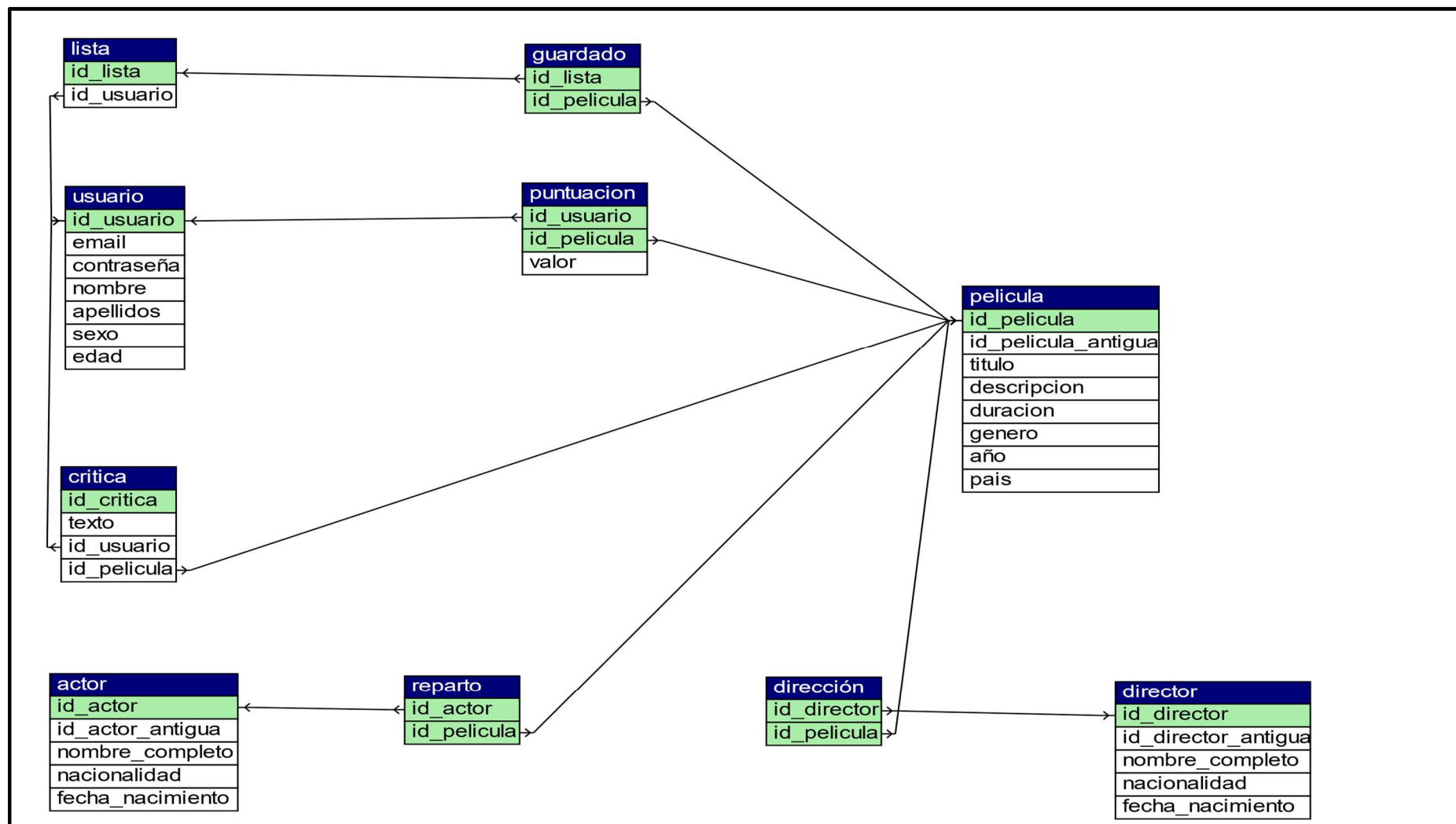


FIGURA 4-1-2-2 IMAGEN DE LA BASE DE DATOS

Modelo relacional

Tablas

Crítica (ID_criticas, Texto, ID_usuario, ID_pelicula)

Usuario (ID_usuario, Email, Contraseña, Nombre, Apellidos, Sexo, Edad)

Puntuación (ID_usuario, ID_pelicula, Valor)

Película (ID_pelicula, Titulo, Descripción, Duración, Genero, Año, País)

Director (ID_ditector, Nombre, Nacionalidad, Fecha de nacimiento)

Actor (ID_actor, Nombre, Apellidos, Nacionalidad, Fecha_Nacimiento)

Lista (ID_lista, ID_usuario)

Guardado (ID_lista, ID_pelicula)

Dirección (ID_pelicula, ID_ditector)

Reparto (ID_pelicula, ID_actor)

Dependencias

Crítica (ID_usuario → Usuario (ID_usuario))

Crítica (ID_pelicula → Película (ID_pelicula))

Puntuación (ID_usuario → Usuario (ID_usuario))

Puntuación (ID_pelicula → Película (ID_pelicula))

Dirección (ID_pelicula → Película (ID_pelicula))

Dirección (ID_ditector → Director (ID_ditector))

Reparto (ID_pelicula → Película (ID_pelicula))

Reparto (ID_actor → Actor (ID_actor))

Lista (ID_usuario → Usuario (ID_usuario))

Guardado (ID_lista → Lista (ID_lista))

Guardado (ID_pelicula → Película (ID_pelicula))

4.1.3. Diseño de interfaces

Tener como referencia para este apartado al módulo de Diseño de Interfaces

He elegido tonos de amarillo y naranja para mi página de reviews de películas porque crean una armonía visual que transmite energía y calidez. El amarillo atrae la atención y hace la interfaz acogedora, mientras que el naranja genera entusiasmo y participación.

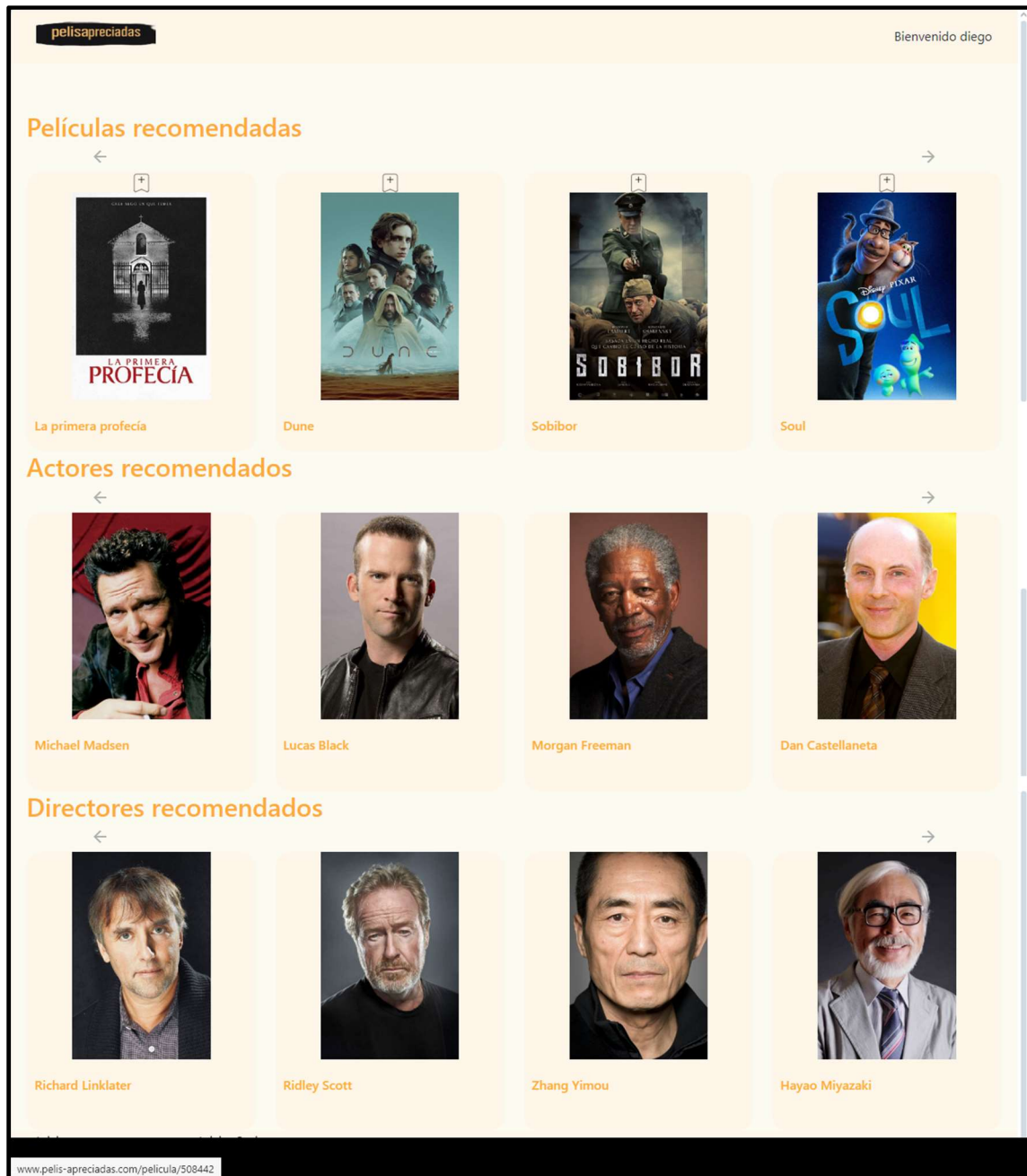
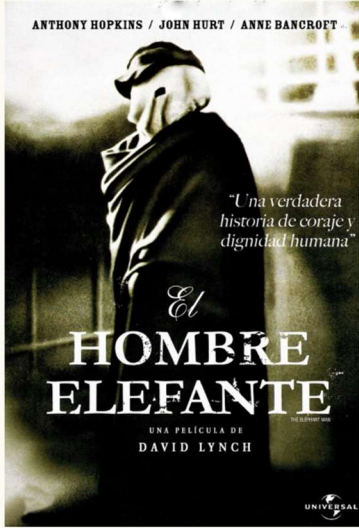


FIGURA 4-1-3-1 IMAGEN PÁGINA INICIO

pelisapreciadas

Iniciar Sesión



ANTHONY HOPKINS / JOHN HURT / ANNE BANCROFT

"Una verdadera historia de coraje y dignidad humana"

El
HOMBRE ELEFANTE

UNA PELÍCULA DE
DAVID LYNCH

UNIVERSAL

El hombre elefante

A finales del siglo XIX, el doctor Frederick Treves descubre en un circo a un hombre llamado John Merrick. Se trata de un ciudadano británico con la cabeza monstruosamente deformada, que vive en una situación de constante humillación y sufrimiento al ser exhibido diariamente como una atracción de feria.


Duración: 124
Año de estreno: 1980-10-09
País de origen: United States of America

Generos: Drama, Historia

Puntuacion de la película

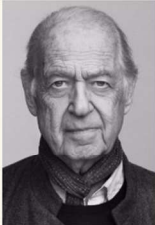
☆☆☆☆☆☆☆☆

Director de la película




[David Lynch](#)


Actores de la película



John Standing
Dr. Fox



John Gielgud
Carr Gomm



Freddie Jones
Bytes



Wendy Hiller
Mothershead

Comentarios sobre la película

Añadir comentario:

Añadir comentario

FIGURA 4-1-3-2 IMAGEN PÁGINA PELÍCULAS

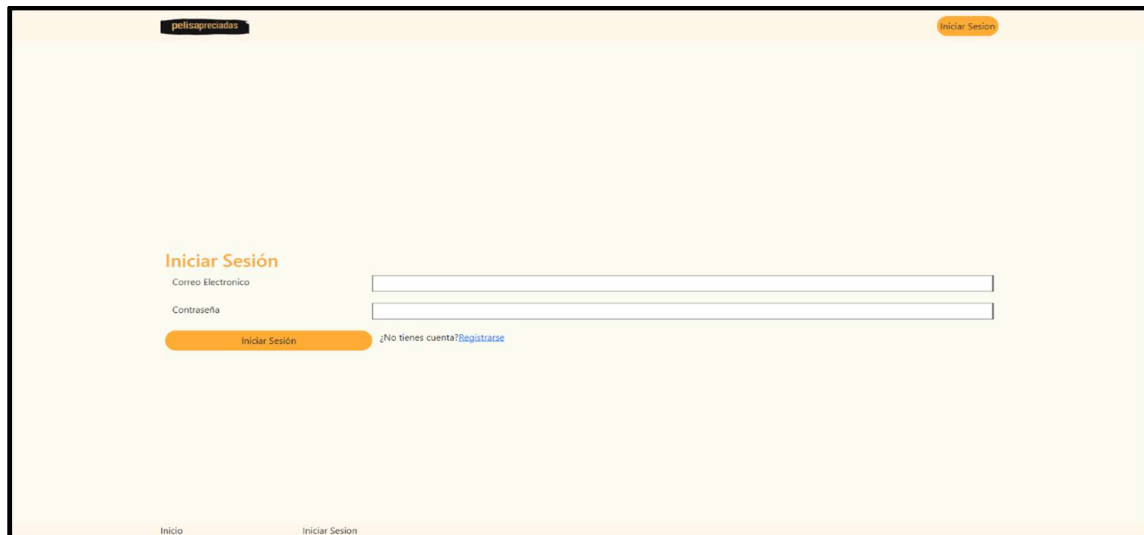


FIGURA 4-1-3-3 IMAGEN PÁGINA LOG IN

4.2. Pruebas

Tabla 4-2-1 Pruebas

TIPO DE PRUEBA	DESCRIPCIÓN PRUEBA	RESULTADO ESPERADO	RESULTADO OBTENIDO
Funcional	Prueba lista sin iniciar sesión.	En la página de inicio si no estas logueado no te debería de dejar añadir a lista de favoritos la película que seleccionas.	El esperado
Funcional.	Prueba añadir puntuación en películas sin iniciar sesión.	En la página de películas si no estas logueado no te debería de dejar dar tu puntuación de las películas.	El esperado
Funcional	Prueba ver criticas usuarios sin iniciar sesión.	En la página de películas si no estas logueado te debería de dejar ver las críticas de otros usuarios.	El esperado
Carga	Prueba cargar datos de la API	Se debería de cargar la información necesitada de la API según la url dada en la base de datos.	El esperado

5. Conclusiones finales

5.1. Grado de cumplimiento de los requisitos fijados

Todos los requisitos funcionales especificados en el apartado 2 de este mismo archivo han sido cumplidos con éxito en mi página de reviews de películas. Los usuarios pueden registrarse e iniciar sesión de manera segura utilizando su correo electrónico, garantizando así una experiencia protegida y personalizada. Además, la página de reviews de películas permite a los usuarios acceder a información detallada sobre cada película, incluyendo sinopsis, elenco, director, duración, géneros y fecha de estreno, en páginas únicas dedicadas a cada película. Asimismo, los usuarios tienen la capacidad de calificar las películas sin necesidad de añadir una crítica previa, y la calificación promedio de cada película se actualiza en tiempo real y se muestra en su respectiva página. También se ha implementado la funcionalidad para que los usuarios gestionen listas de películas favoritas, proporcionando así una experiencia completa y satisfactoria para nuestra comunidad de usuarios cinéfilos.

5.2. Propuestas de mejora o ampliaciones futuras

Tras realizar un análisis de las desviaciones de la aplicación, he detectado algunas áreas donde el proyecto necesita mejoras para optimizar y mejorar la experiencia del usuario.

Esas posibles mejoras son:

- ✓ Permitir la creación de múltiples listas para guardar películas, ofreciendo así una organización más flexible y cómoda para los usuarios.
- ✓ Habilitar la opción de comentar en los comentarios de otros usuarios, aumentando la interacción entre usuarios y generando debates dentro de la comunidad.
- ✓ Implementar una función de búsqueda que permita buscar películas, directores y actores por nombre.
- ✓ Incorporación de filtros de búsqueda por género, facilitando a los usuarios encontrar películas que se ajusten a las preferencias especificadas.
- ✓ Desarrollar un sistema de recomendaciones basado en las películas seleccionadas por el usuario, ofreciendo sugerencias personalizadas y relevantes.

6. Guía de usuario

Introducción

Bienvenido a la aplicación de reviews de películas, donde podrás ver una amplia variedad de títulos, leer críticas y compartir tus opiniones. Esta guía te ayudará a entender cómo utilizar las funciones disponibles tanto si estás logueado como si no lo estás.

Navegación Sin Estar Logueado

Al no estar logueado en nuestra aplicación, tendrás acceso a las siguientes funcionalidades:

1. Explorar películas:

- Podrás ver toda la información de las películas disponibles en la plataforma, incluyendo sinopsis, géneros, duración, país de origen, elenco.

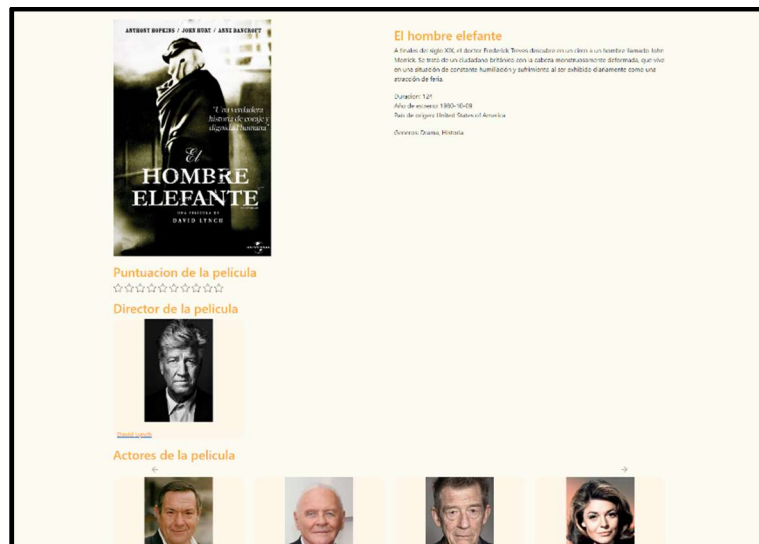


Figura 6-1 Imagen página películas

2. Explorar actores:

- Podrás ver toda la información de los actores disponibles en la plataforma, incluyendo nombre completo, fecha de nacimiento, país de nacimiento y películas relacionadas.

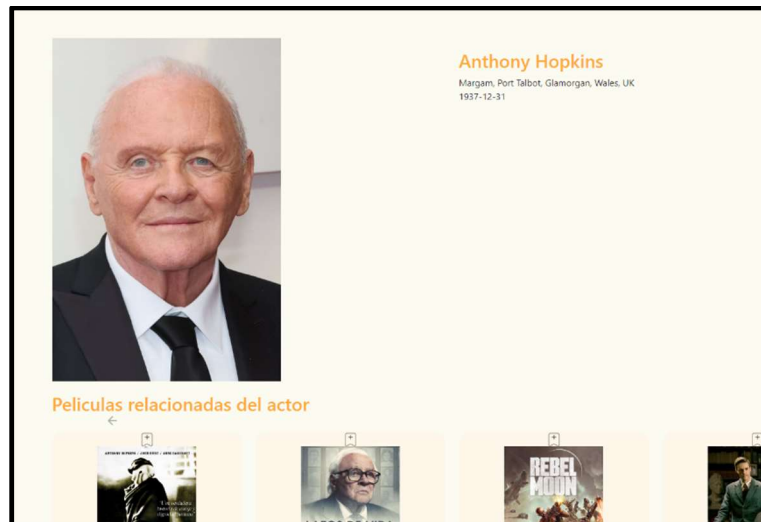


Figura 6-2 Imagen página actor

3. Explorar directores:

- Podrás ver toda la información de los actores disponibles en la plataforma, incluyendo nombre completo, fecha de nacimiento, país de nacimiento y películas relacionadas.

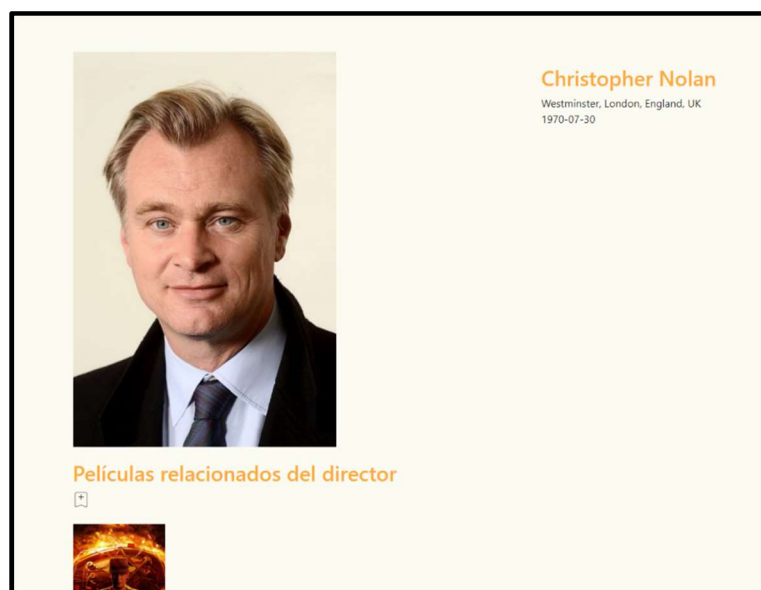


Figura 6-3 Imagen página director

4. Leer Críticas:

- Puedes leer las críticas y ver las puntuaciones que otros usuarios han dejado sobre las películas en la página de cada película.

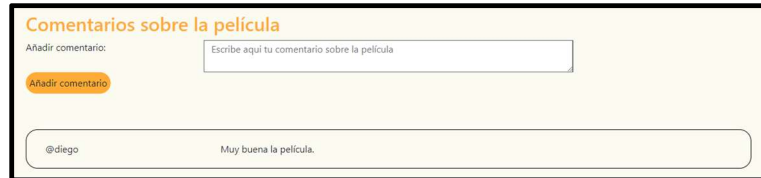


Figura 6-4 Leer críticas

Limitaciones:

- No podrás guardar películas en listas personalizadas.
- No podrás puntuar las películas.
- No podrás escribir críticas sobre las películas.

Navegación Estando Logueado

Al crear una cuenta e iniciar sesión, se desbloquearán funciones adicionales que te permitirán interactuar de manera más completa con nuestra aplicación.

1. Administrar Listas:

- Podrás insertar películas a una lista. Y ver todas las películas que están en esa lista. Para ello deberás darle al botón con el icono del más.

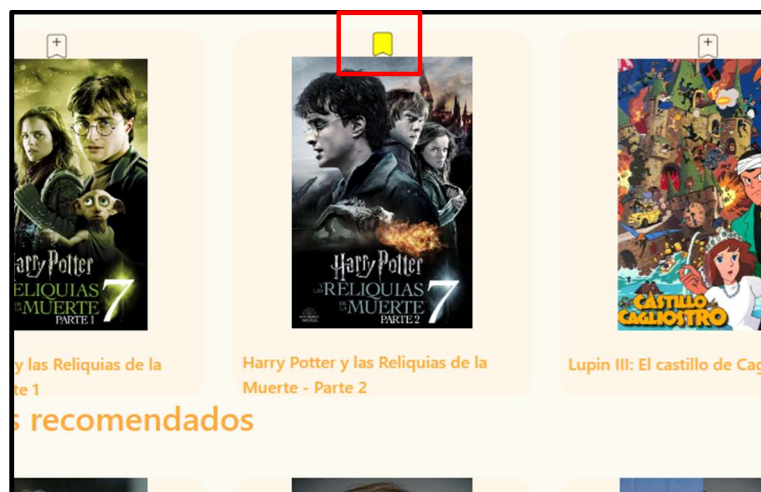


Figura 6-5 Administrar lista

2. Puntuar Películas:

- Tendrás la opción de puntuar las películas. Las puntuaciones ayudan a otros usuarios a descubrir nuevas películas y a evaluar la calidad de las mismas.
- Para puntuar una película, navega a la página y en el apartado de puntuaciones haz click en.



Figura 6-6 Puntuar películas

3. Escribir Críticas:

- Puedes escribir tus propias críticas sobre las películas. Compartir tus opiniones y experiencias con la comunidad.
- Para escribir una crítica, ve a la página de la película y en el apartado de escribir crítica rellena el formulario con tu crítica.

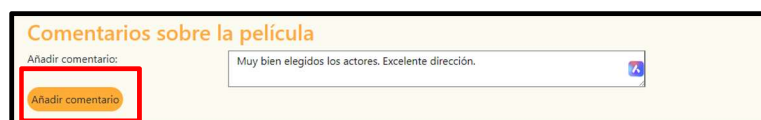


Figura 6-7 Añadir crítica

4. Ver Información Completa:

- Al estar logueado, seguirás teniendo acceso a toda la información de las películas, igual que cuando no estás logueado.

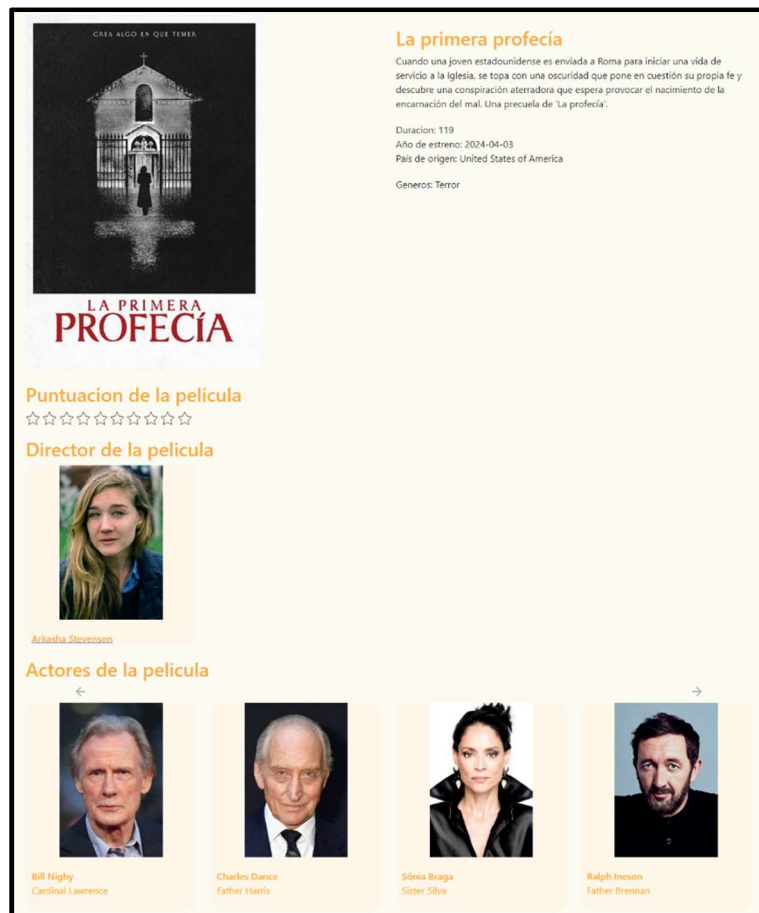


Figura 6-8 Imagen página películas administrador

Cómo Crear una Cuenta

Para disfrutar de todas las funcionalidades mencionadas, sigue estos pasos para crear una cuenta:

1. Registro:

- En la página de inicio, haz clic en "Iniciar sesión y en el formulario pulsa en la opción de registrarse."
- Completa el formulario de registro con tus datos personales (nombre, correo electrónico, contraseña...).

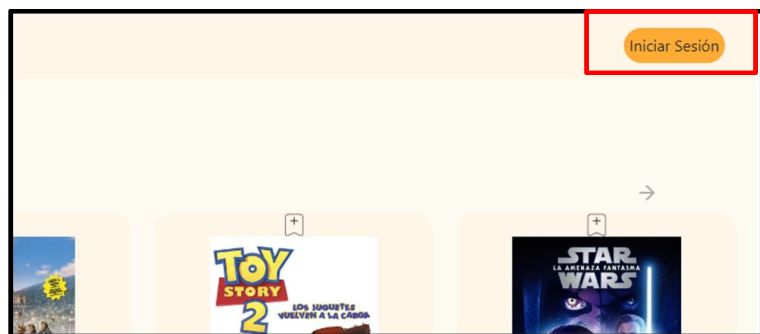


Figura 6-9 Botón iniciar sesión



Figura 6-10 Enlace registrarse



Figura 6-11 Botón registrarse

Si ya tienes una cuenta creada:

1. Iniciar Sesión:

- Haz clic en "Iniciar Sesión".
- Introduce tu correo electrónico y contraseña.
- Haz clic en "Entrar" para acceder a tu cuenta. Y listo

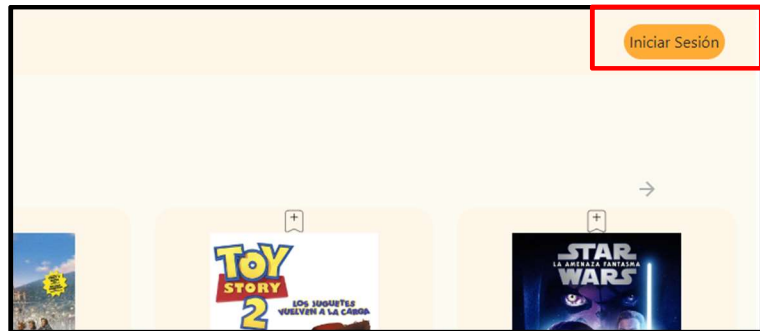


Figura 6-12 Botón iniciar sesión

A screenshot of the login form. The title 'Iniciar Sesión' is in orange. Below it, there are two input fields: 'Correo Electronico' with the value 'diego@gmail.com' and 'Contraseña' with masked characters '*****'. At the bottom left is a yellow 'Iniciar Sesión' button, and at the bottom right is a link that says '¿No tienes cuenta? [Registrarse](#)'.

Figura 6-13 Iniciar sesión

Esperamos que esta guía te ayude a sacar el máximo provecho de nuestra aplicación de películas. Ya sea que estés explorando nuestras colecciones sin estar logueado o que estés disfrutando de las funciones avanzadas como usuario registrado, estamos seguros de que encontrarás algo que te encantará. ¡Disfruta de la experiencia y felices visualizaciones!

7. Referencias

<https://flotadigital.com/club/tutoriales/windows/descargar-e-instalar-composer-en-windows-10/>

<https://flotadigital.com/club/tutoriales/windows/instalar-symfony-en-windows-10/>

<https://symfony.com/doc/current/setup.html>

<https://symfony.com/doc/current/scheduler.html>

<https://insidert.medium.com/fix-curl-error-60-in-wamp-server-a0ffff8dbb29>

<https://curl.se/docs/caextract.html>

<https://symfony.com/doc/current/templates.html>

Referencia HTML

1. **W3C HTML Specification:** Documentación oficial del World Wide Web Consortium (W3C) sobre la especificación HTML5.
2. **MDN Web Docs:** Amplia referencia y guía sobre HTML, ofrecida por Mozilla Developer Network.
3. **HTML Living Standard:** Versión en desarrollo continuo del HTML, mantenida por WHATWG (Web Hypertext Application Technology Working Group).

Referencia CSS

1. **W3C CSS Specifications:** Documentación oficial del World Wide Web Consortium (W3C) sobre las especificaciones de CSS.
2. **MDN Web Docs:** Completa referencia y guía sobre CSS, proporcionada por Mozilla Developer Network.
3. **CSS-Tricks:** Blog y recurso educativo con tutoriales, ejemplos y artículos sobre el uso de CSS en el desarrollo web.

Referencia JS

1. **MDN Web Docs:** Una guía completa y detallada sobre JavaScript, ofrecida por Mozilla Developer Network.
2. **ECMAScript Specification:** Documentación oficial de la especificación ECMAScript, que define las nuevas características y estándares del lenguaje.

3. **JavaScript.info:** Un recurso educativo con tutoriales y artículos profundos sobre JavaScript y su aplicación en el desarrollo web.
4. **Node.js Official Website:** Sitio oficial de Node.js, con documentación y recursos para usar JavaScript en el servidor.

Referencia PHP

1. **PHP Official Website:** Sitio oficial de PHP, con documentación completa, noticias y recursos sobre el lenguaje.
2. **W3Schools PHP Tutorial:** Tutorial interactivo sobre PHP, ideal para principiantes y desarrolladores intermedios.
3. **Laravel Documentation:** Documentación oficial del framework Laravel, uno de los más populares para PHP.
4. **Symfony Documentation:** Documentación oficial del framework Symfony, ampliamente utilizado para aplicaciones web en PHP.

Referencia MySQL

1. **MySQL Official Website:** Sitio oficial de MySQL, con documentación, descargas y noticias.
2. **MySQL Documentation:** Documentación completa y detallada sobre el uso y administración de MySQL.
3. **W3Schools MySQL Tutorial:** Tutorial interactivo y recursos para aprender MySQL desde lo básico hasta avanzado.

Referencia Symfony

1. **Symfony Official Website:** El sitio oficial de Symfony proporciona documentación completa, tutoriales y recursos para aprender y comenzar a trabajar con el framework.
2. **Symfony Documentation:** Documentación detallada sobre todas las características y componentes de Symfony, incluyendo guías de inicio rápido, tutoriales y referencias.
3. **SymfonyCasts:** Una plataforma de aprendizaje que ofrece una amplia variedad de cursos y tutoriales en video sobre Symfony y otras tecnologías relacionadas.
4. **Symfony GitHub Repository:** Repositorio oficial de Symfony en GitHub, donde se puede acceder al código fuente, informar problemas y contribuir al desarrollo del framework.
5. **Symfony Blog:** El blog oficial de Symfony proporciona noticias, actualizaciones y anuncios sobre el desarrollo del framework y la comunidad que lo rodea.

Referencia Twig

1. **Sitio Web Oficial de Twig:** El sitio web oficial de Twig proporciona una documentación completa y detallada sobre todas las características y funcionalidades del motor de plantillas, incluyendo ejemplos de código y guías de referencia.
2. **SymfonyCasts:** SymfonyCasts ofrece numerosos tutoriales en línea, libros y cursos disponibles para aprender y dominar Twig, así como otras tecnologías relacionadas con PHP.
3. **Udemy y YouTube:** Plataformas como Udemy y YouTube también ofrecen una variedad de recursos para aprender Twig, incluyendo cursos en video y tutoriales gratuitos impartidos por expertos en el tema.

Referencia Bootstrap

1. **Sitio Web Oficial de Bootstrap:** El sitio web oficial de Bootstrap proporciona documentación detallada, ejemplos de código y guías de referencia para aprender y utilizar el framework de manera efectiva.
2. **Bootstrap Themes:** Hay numerosos sitios web que ofrecen temas y plantillas Bootstrap premium y gratuitos que los desarrolladores pueden utilizar para personalizar el aspecto y la sensación de sus sitios web y aplicaciones.
3. **Stack Overflow:** Stack Overflow es una comunidad en línea de desarrolladores que ofrece una amplia gama de preguntas y respuestas sobre Bootstrap, lo que la convierte en un recurso valioso para resolver problemas y aprender nuevas técnicas.
4. **GitHub:** El repositorio oficial de Bootstrap en GitHub es otro recurso útil para los desarrolladores, ya que pueden acceder al código fuente, informar problemas y contribuir al desarrollo continuo del framework.

Referencia Doctrine

1. **Documentación Oficial de Doctrine:** El sitio oficial de Doctrine proporciona documentación completa sobre sus características y funcionalidades.
2. **Tutorial de Doctrine en SymfonyCasts:** Serie de tutoriales en video sobre cómo usar Doctrine dentro del framework Symfony.
3. **Guía de Migraciones de Doctrine:** Documentación detallada sobre el sistema de migraciones de Doctrine.
4. **Doctrine GitHub Repository:** Repositorio oficial de Doctrine en GitHub, donde se puede acceder al código fuente, informar problemas y contribuir al desarrollo del ORM.

5. **PHP The Right Way: Doctrine:** Un recurso adicional que proporciona contexto y ejemplos sobre cómo usar Doctrine en proyectos PHP.

Referencia DBAL

1. **Documentación Oficial de Doctrine:** El sitio oficial de Doctrine proporciona documentación completa sobre sus características y funcionalidades.
2. **Tutorial de Doctrine en SymfonyCasts:** Serie de tutoriales en video sobre cómo usar Doctrine dentro del framework Symfony.
3. **Guía de Migraciones de Doctrine:** Documentación detallada sobre el sistema de migraciones de Doctrine.
4. **Doctrine GitHub Repository:** Repositorio oficial de Doctrine en GitHub, donde se puede acceder al código fuente, informar problemas y contribuir al desarrollo del ORM.
5. **PHP The Right Way: Doctrine:** Un recurso adicional que proporciona contexto y ejemplos sobre cómo usar Doctrine en proyectos PHP.

Referencia API

1. **API Academy:** Un recurso en línea que ofrece cursos y artículos sobre diseño y gestión de API, ayudando a los desarrolladores a entender mejor cómo construir y utilizar API efectivas.
2. **Documentación de API de Google:** Google ofrece documentación exhaustiva sobre sus diversas API, desde Google Maps hasta Google Cloud, proporcionando guías y ejemplos para desarrolladores de todos los niveles.
3. **API Gateway de AWS:** Amazon Web Services proporciona una detallada documentación sobre su API Gateway, una herramienta para crear, publicar, mantener, monitorear y asegurar APIs a cualquier escala.
4. **Documentación de RESTful API:** Sitios como "RESTful API Tutorial" proporcionan guías detalladas y ejemplos sobre cómo construir y consumir API RESTful, facilitando la comprensión de este tipo de arquitectura.
5. **ProgrammableWeb:** Un directorio de API que contiene información sobre miles de API públicas, incluyendo detalles sobre cómo usarlas y ejemplos de aplicaciones que las integran.