

Diseño e Implementación de una Aplicación Web para la Búsqueda y Visualización de Información sobre Películas

Manuel Antelo Pereira

Grado en Ingeniería Informática
Desarrollo Web

Gregorio Robles Martínez

Santi Caballe Llobet

Fecha de entrega: 09/06/2022



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

AGRADECIMIENTOS

A mi familia, sobre todo a mi mujer por recordarme cada día que podía con ello, y ayudarme con los niños para que yo pudiera sacar tiempo de donde no había.

A mis hijos por hacerme olvidar el agobio que supone trabajar y estudiar a la vez, son un gran aliciente para seguir adelante.

Por último a la UOC por permitirme cumplir mi sueño de poder realizar lo que realmente me gusta.

GRACIAS.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Info Entertainment</i>
Nombre del autor:	<i>Manuel Antelo Pereira</i>
Nombre del consultor/a:	<i>Gregorio Robles Martínez</i>
Nombre del PRA:	<i>Santi Caballe Llobet</i>
Fecha de entrega (mm/aaaa):	09/04/2022
Titulación::	<i>Grado en Ingeniería Informática</i>
Área del Trabajo Final:	<i>Desarrollo Web</i>
Idioma del trabajo:	Castellano
Palabras clave	<i>HTML, CSS, JS6, Bootstrap, API</i>
<p>El presente documento consiste en el plan de Trabajo del TFG de desarrollo web, donde propondremos y veremos los puntos principales para la creación de una aplicación web utilizando lenguajes de programación y herramientas afines a estos, que conforman actualmente la parte visual del desarrollo web, técnicamente llamado FrontEnd.</p> <p>La metodología que se ha utilizado es la obtenida durante todo el proceso de aprendizaje en la carrera de Ingeniería Informática y cursos autodidactas realizados personalmente.</p> <p>Como resultado obtendremos una aplicación web para la gestión de información de títulos de películas que nos permitirá buscar y mostrar títulos con información (Título, Título Original, comentarios, las más populares,...).</p>	
Abstract:	
<p>This document consists of the planning of TFG web development, where we will propose and see the main points for the creation of a web application using programming languages and tools related to these, which currently make up the visual part of web development, technically called FrontEnd.</p> <p>The methodology that has been used is the one obtained during the entire learning process in the Computer Engineering career and self-taught courses</p>	

carried out personally.

As a result, we will obtain a web application for managing title information for movies that will allow us to search for and display titles with information (Title, Original Title, comments, the most popular,...).

Índice

1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	1
1.3 Enfoque y método seguido	2
1.4 Planificación del Trabajo	3
1.5 Breve descripción de las tecnologías a utilizar en el desarrollo	5
1.6 Evaluación de riesgos	7
2. Requisitos del Proyecto	8
2.1 Descripción general	8
2.2 Análisis de requerimientos y funcionalidades de uso	9
2.3 Requisitos específicos y casos de uso	11
3. Diseño del Proyecto	13
3.1 Diseño del logo	13
3.2 Arquitectura del sistema	14
4. Desarrollo e Implementación del proyecto	18
4.1 Lenguajes de programación usados, API y software necesario	20
5. Pruebas del proyecto	27
6. Resultados y Conclusiones	32
7. Bibliografía	36
Anexo I - Entregables del proyecto	37
Anexo II - Acrónimos	37

Lista de figuras

Figura 1- Comparativa distintas webs de búsqueda	3
Figura 2 – Planificación	4
Figura 3 – Planificación detallada	4
Figura 4 – Proceso tecnologías	9
Figura 5 – Diagrama de casos de uso	10
Figura 6 – Logo del proyecto	13
Figura 7 – Ejemplo Cliente/Servidor	14
Figura 8 – Primer boceto UML diseño	14
Figura 9 – Proceso registro de un UNR	15
Figura 10 – Búsqueda de una película	15
Figura 11 – Mapa web	16
Figura 12 – index.html	23
Figura 13 – novedades.html	24
Figura 14 – pelis.html	24
Figura 15 – popular.html	25
Figura 16 – buscador.html	25
Figura 17 – Búsqueda	26
Figura 18 – Resultado prueba css	28
Figura 19 – Búsqueda títulos españoles	29
Figura 20 – Test información película	30
Figura 21 – Registro y login	33
Figura 22 – Formulario de contacto para no registrados	33
Figura 23 – Ejemplo de funcionamiento para no registrados	34

1. Introducción

1.1 Contexto y justificación del Trabajo

Como punto de partida para la realización de este Trabajo Fin de Grado (desde ahora TFG), ha sido la puesta en escena de todo lo aprendido hasta el día de hoy e ir un paso más allá y aplicar todo esto y la búsqueda de una nueva tecnología que nos servirá para la realización de este TFG.

La elección de la temática de este TFG fue una tarea complicada debido a la cantidad de información que podemos encontrarnos y la necesidad de ciertas ideas referentes a las aplicaciones web. En un primer intento se intentó por una aplicación web para la búsqueda de información sobre alimentos y su información nutricional y alérgenos, pero sin mucho auge ni bases de datos de uso libre para la realización de este TFG, por lo que se optó por esta vía.

Y debido a que en la actualidad vivimos en un mundo donde existe un creciente auge en el consumo de contenido audiovisual en todo el mundo. Esto es debido, en parte, a la aparición y popularización de las principales plataformas de contenido multimedia en streaming (Netflix, Prime, Movistar+, HBO, etc.) y a su facilidad de uso. Esto sumado también a la popularización de las redes sociales, hace que muchos de los temas de debate en línea tengan que ver o estén relacionados con películas de actualidad.

Vista la popularidad dentro de este marco, se decidió por este proyecto, el cual pretende crear una aplicación web en la que se pueda consultar información de películas, poder consultar su detalles, puntuación, comentarios, poder comentar nuestro punto de vista y conocer el de los demás.

Existe un amplio abanico de alternativas en la web que permiten la consulta de información y sus detalles, como pueden ser **IMDb**, **Filmaffinity**, **themovieDB**, etc... Pero lo que se busca con este proyecto es crear una aplicación web de películas más centrada en sus usuarios y ofrecer información más detallada, otros puntos de vista, cuales son más recomendables, etc...

1.2 Objetivos del Trabajo

Para que el proyecto cumpla con las expectativas iniciales, es necesario alcance una serie de objetivos. Los definiremos desde dos puntos de vista.

El primero es el más lógico y es desde el punto de vista académico, que tiene como objetivo aplicar los conocimientos adquiridos a lo largo de la formación en el Grado y aumentar dicha formación con nuevos lenguajes y prepararse para el mundo laboral actual donde el desarrollo web está en auge.

El segundo punto, es el software a desarrollar, que debe poder gestionar la información, gestionar los usuarios, la fuente de información, donde poder ver dicha película y tener un mecanismo de evaluación de la misma. Un resumen de las funcionalidades más relevantes de nuestra aplicación será:

- Un módulo para la gestión de los usuarios que agrupará las operaciones relacionadas con el alta en el sistema, edición y eliminación. Pudiendo existir varios tipos de usuario (Administrador y usuario final).
- La gestión de las películas que agrupara las operaciones donde podremos ver la información sobre ellas, acciones de búsqueda, etc.
- La gestión de los comentarios sobre las películas donde los usuarios expondrán su opinión.
- La puntuación que se podrá dar a una determinada película con un rango de 1 a 10, calculando la media de los usuarios obtendremos la puntuación.
- Gestión de recomendaciones y películas más vistas o más populares.
- Gestión de búsquedas que nos permitirá la ejecución de la búsqueda aplicando ciertos filtros.

Con el desarrollo de este TFG busco profundizar en el desarrollo de las aplicaciones web y los lenguajes de programación y herramientas más populares usadas actualmente, así como poner en práctica distintos conceptos aprendidos en la universidad pero que no hemos tenido la ocasión u oportunidad de aplicar en ningún proyecto.

1.3 Enfoque y método seguido

Actualmente existen varias opciones similares en el mercado dada la popularidad y sobre todo a la pandemia del COVID19 la búsqueda sobre información de películas se disparó. Como bien se comentó anteriormente existen páginas como IMDb, Filmaffinity, themovieDB e infinidad de ellas pero no ofrecen todo lo que un usuario final desearía. Algunas características de dichas páginas las podemos ver en la siguiente figura 1 donde se muestra las carencias y virtudes de cada plataforma:

	DCine	IMDb	Filmaffinity
Aplicación móvil	✗	✓	✗
Buscador	✓	✓	✓
Autocompletar búsqueda	✗	✓	✓
Cartelera	✗	✗	✓
Conexión segura	✗	✓	✗
Diseño web adaptativo	✗	✓	✓
Información de los cines	✗	✗	✓
Listas	✓	✓	✓
Próximos estrenos	✓	✓	✓
Redes sociales (Facebook, Twitter...)	✓	✓	✓

Figura 1 – Comparativa distintas webs de búsqueda

El desarrollo de esta aplicación web facilitará mucho la búsqueda y la elección del título, pudiendo llegar a los objetivos marcados por este TFG y pudiendo cubrir algunas carencias que se pueden tener en las webs similares y que hay actualmente.

La metodología que se usará para la parte del diseño usaremos UML y para el desarrollo y debido a su popularidad será tecnología *agile* para poder ir realizando pequeños desarrollos completos como piezas únicas para una implantación más ágil.

1.4 Planificación del Trabajo

Partiendo de la base que este TFG va ser realizado por una sola persona y no por un equipo multidisciplinar.

Para la realización de la planificación del trabajo utilizaremos el software de Microsoft perteneciente al paquete Office denominado Microsoft Project, en el cual realizaremos un diagrama de Gantt como se muestra a continuación:

			Modo de	Nombre de tarea	Duración	Comienzo	Fin
1				PEC 1 - Plan de Trabajo	10 días	mié 16/02/22	mar 01/03/22
2				Definición de Objetivos	2 días		
3				Tecnologías que se utilizarán	5 días		
4				Planificación temporal	2 días		
5				Evaluación de riesgos	1 día		
6				PEC 2 - Análisis y Diseño	23 días	mié 02/03/22	vie 01/04/22
7				Preparación del entorno de trabajo y Requisitos	3 días	mié 02/03/22	vie 04/03/22
8				Análisis de requisitos	2 días	dom 06/03/22	lun 07/03/22
9				Análisis de funcionalidades	2 días	mar 08/03/22	mié 09/03/22
10				Diseño del proyecto			
11				Diagramas	6 días	jue 10/03/22	jue 17/03/22
12				Maquetación	3 días	jue 17/03/22	sáb 19/03/22
13				PEC 3 - Implementación y Desarrollo	37 días	sáb 02/04/22	lun 23/05/22
14				Instalación del entorno	2 días	sáb 02/04/22	lun 04/04/22
15				Arquitectura	4 días	mar 05/04/22	vie 08/04/22
16				Aprendizaje de las tecnologías a usar	22 días	sáb 02/04/22	lun 02/05/22
17				Desarrollo del proyecto			
18				Uso lenguajes de programación	8 días	jue 12/05/22	lun 23/05/22
19				Frameworks	4 días	lun 16/05/22	jue 19/05/22
20				Diseño responsive	4 días	mié 18/05/22	lun 23/05/22
21				Análisis de riesgos	2 días	vie 20/05/22	lun 23/05/22
22				Documentación	2 días	jue 19/05/22	vie 20/05/22
23				Pruebas del proyecto	3 días	dom 20/03/22	mar 22/03/22
24				Cambios respecto a la planificación inicial	2 días	dom 22/05/22	lun 23/05/22
25				Implementación y pruebas	12 días	mar 03/05/22	mié 18/05/22
26				Documentación	8 días	jue 12/05/22	lun 23/05/22
27				PEC 4 - Memoria y Presentación	13 días	mar 24/05/22	jue 09/06/22
28				Realización de la Memoria	6 días	mar 24/05/22	mar 31/05/22
29				Presentación	4 días	mié 01/06/22	lun 06/06/22

Figura 2 – Planificación

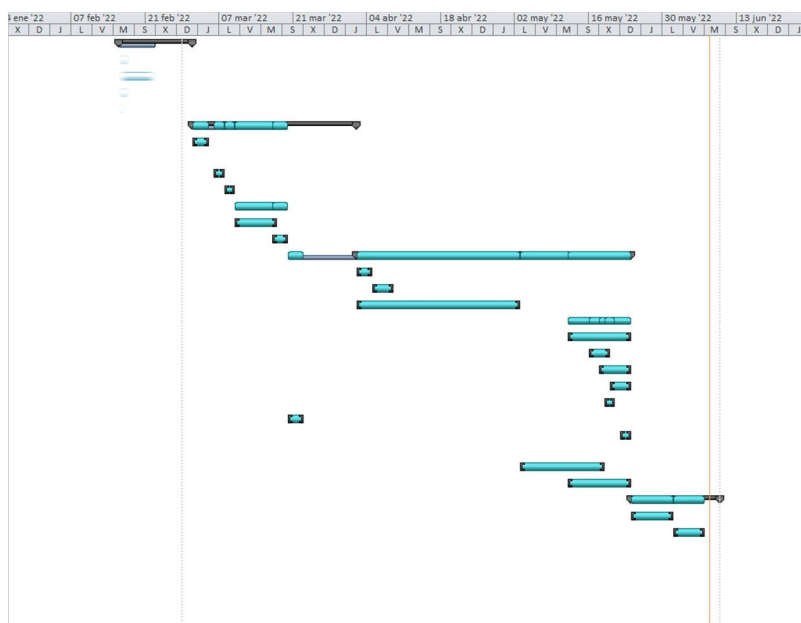


Figura 3 – Planificación detallada

1.5 Breve descripción de las tecnologías a utilizar en el desarrollo

En este apartado del plan de trabajo se explicarán, de forma resumida, las tecnologías y herramientas que se van utilizar para la realización de este proyecto.

- **JavaScript** es un lenguaje de programación diseñado en un principio para añadir interactividad a las páginas webs y crear aplicaciones web. A pesar de la similitud en el nombre, no está relacionado con Java. Se emplea en el desarrollo de páginas web para tareas como cambiar automáticamente la fecha de una página, hacer que una página aparezca en una ventana emergente al hacer clic en un enlace o que un texto o imagen cambien al pasar el ratón por encima. También suele emplearse para hacer encuestas y formularios. Se ejecuta en el ordenador del visitante a la web, por lo que no requiere descargas constantes desde el sitio web.
- **CSS** (hojas de estilo en cascada) es un lenguaje que define la apariencia de un documento escrito en un lenguaje de marcado (por ejemplo, HTML).

Así, a los elementos de la página web creados con HTML se les dará la apariencia que se desee utilizando CSS: colores, espacios entre elementos, tipos de letra,... separando de esta forma la estructura de la presentación.

- **Bootstrap** es un framework CSS de código abierto que favorece el desarrollo web de un modo más sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con la que es posible modificar tipografías, formularios, botones, tablas, navegaciones, menús desplegados, etc. También existe la posibilidad de utilizar extensiones de Javascript adicionales.
- **HTML** (Lenguaje de Marcas de Hipertexto, del inglés *HyperText Markup Language*) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).
- **API Rest:** La forma completa de REST API es la interfaz de programación de aplicaciones de transferencia de estado representacional, más comúnmente conocida como servicio web REST API. Significa que cuando se llama a una API RESTful, el servidor

transferir a representación de los recursos solicitados estado al sistema del cliente.

- **Git** es un sistema de control de versiones nos va a servir para trabajar en equipo de una manera mucho más simple y optima cuando estamos desarrollando software.

Con Git vamos a poder controlar todos los cambios que se hacen en nuestra aplicación y en nuestro código y vamos a tener control absoluto de todo lo que pasa en el código, pudiendo volver atrás en el tiempo, pudiendo abrir diferentes ramas de desarrollo, etc.

- **Github** es una plataforma de alojamiento, propiedad de Microsoft, que ofrece a los desarrolladores la posibilidad de crear repositorios de código y guardarlos en la nube de forma segura, usando un sistema de control de versiones, llamado [Git](#).

Facilita la organización de proyectos y permite la colaboración de varios desarrolladores en tiempo real. Es decir, nos permite **centralizar el contenido del repositorio** para poder colaborar con los otros miembros de nuestra organización.

- **Visual Studio Code** es un editor de código fuente desarrollado por Microsoft para Windows, Linux, y Web. Incluye soporte para la depuración, control integrado de [Git](#), resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Visual Studio Code se basa en [Electron](#), un *framework* que se utiliza para implementar [Chromium](#) y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el *framework* Electron, el software no usa [Atom](#) y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online).

- **Postman** es un programa que cuenta con una interfaz gráfica sencilla que permite el diseño,"mock" o simulacro, debug o depurado, la realización de pruebas, documentación, monitorización y publicación de APIs, todo desde un único sitio. Para este proyecto se ha utilizado Postman únicamente para la realización de pruebas sobre el API del servicio.

- **MySQL** es un software que presenta un servidor robusto, rápido, multithread y multiusuario de bases de datos SQL (Structured Query Language). MySQL está destinado a sistemas de producción de carga pesada de misión crítica, así como a la incorporación en software de implementación masiva. Es posible utilizar MySQL bajo dos licencias distintas, puede ser utilizado como un producto Open Source bajo las condiciones de la licencia GNU (General Public License), o se puede adquirir la versión comercial estándar de Oracle.
- **APIs:** Para la realización del TFG se tendrán en cuenta las siguientes APIs:

<http://www.omdbapi.com/>

<https://developers.themoviedb.org/3/getting-started/introduction>

<https://developer.imdb.com/>

<https://github.com/xsga/filmaffinity-api>

1.6 Evaluación de Riesgos

En este proyecto el riesgo principal es el relacionado con el personal, ya que es un proyecto realizado individualmente y con un tiempo marcado y justo para dicha realización.

Al estar dividido el TFG en varias PECs, sobre todo el primer entregable, el peor riesgo es intentar definir al 100% todo el proceso en tan poco tiempo disponible y con el aliciente de una persona trabajadora, siendo un riesgo a evaluar la introducción a nuevos lenguajes de programación y su implementación.

Otro riesgo a evaluar, son los imprevistos que puedan surgir en el día a día en la realización de este TFG, al ser una única persona la encargada de la realización, no podría ser sustituida por otra, siendo una posibilidad no una probabilidad.

Fuera de los riesgos personales en este TFG, nos centraremos en los técnicos que son los que nos vamos a encontrar en dicho proyecto. Uno de ellos puede ser el desconocimiento de las tecnologías a usar, con su consiguiente retraso a la hora de avanzar en el desarrollo e implementación. Como se comentó en este plan de trabajo usaremos tecnologías ágiles que nos va permitir realizar desarrollos pequeños pero completos.

Por último, un riesgo a tener en cuenta es la no adaptabilidad de los lenguajes seleccionados y las herramientas, lo que supondría un retraso y pérdida de tiempo buscando una mejor tecnología que se adapte a nuestro TFG.

2. Requisitos del Proyecto

En este apartado se identificará y clasificará los requisitos del futuro sistema para poder analizar las necesidades funcionales del mismo. Esta aplicación web va dirigida a cualquier usuario que quiera información sobre películas desde el título original al reparto y equipo de producción.

El nombre de la aplicación es: “Info Entertainment”. El objetivo principal de la aplicación web es mostrar información de cualquier película estrenada o por estrenar junto a la participación del usuario para llevar un seguimiento de sus películas, ya sea película vista, favorita, pendiente de ver o directamente descartada. Otros objetivos a desarrollar en un futuro, para la evolución de la aplicación, serían el poder obtener información de los cines, obtener ofertas de su sala de cine cercana y tener un registro de usuarios con su parte BackEnd. El objetivo de este sistema es llegar a ser un modelo de negocio rentable, óptimo y convertirse en una de las aplicaciones de referencia en la búsqueda de títulos.

2.1 Descripción general

El sistema es una aplicación web independiente, es decir, requiere la utilización de una API externa para obtener la información de las películas. Desarrollada de manera individual y con una descripción detallada en este mismo documento.

En el sistema existen tres tipos de funciones básicas en las que el usuario podrá acceder a información sobre películas, ver las novedades, las más populares y acceder al buscador. Para la realización del proyecto usaremos tecnologías ágiles siguiendo el proceso como se muestra en la figura 4 en donde el método de desarrollo de software ágil intenta reducir la complejidad de los proyectos de software actuales e incrementar su flexibilidad. El desarrollo de software ágil tiene muy en cuenta los cambios solicitados y responde de manera flexible a los nuevos requisitos surgidos durante el transcurso del proyecto.

Los pasos del proceso Agile son los siguientes:

- **Diagnóstico** (*Product Owner, Stakeholders, Agile Manager*): Incluye el análisis de los requerimientos y la tormenta de ideas.
- **Diseño** (*Product Owner, Stakeholders, Scrum Master*): Incluye la documentación del diseño y el prototipado.

- **Desarrollo** (*Scrum Master, Team Members*): Incluye la construcción, las pruebas, la revisión y la clarificación. Es un proceso iterativo.
- **Aseguramiento de Calidad** (*Scrum Master, Team Members*): Incluye la identificación y la resolución de defectos.
- **Despliegue** (*Team Members, Product Owner*): Incluye la puesta en producción y el soporte post-venta.

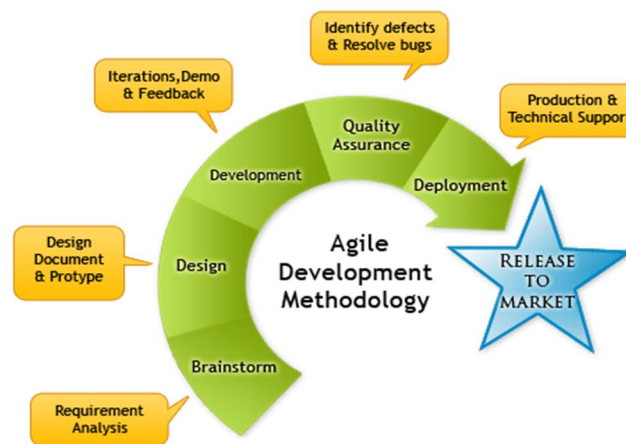


Figura 4 – Proceso tecnologías ágiles

La primera parte es el **Diagnóstico** (*Product Owner, Stakeholders, Agile Manager*): Incluye el análisis de los requerimientos y la tormenta de ideas (ya vistas anteriormente).

2.2 Análisis de requerimientos y funcionalidades de uso

Los requerimientos desde el punto de vista funcional los podemos dividir en los siguientes puntos:

- El acceso a la aplicación será a través de una web abierta a todo el público.
- Desde el punto de vista de la tecnología estará basada y desarrollada en FrontEnd usando las tecnologías expuestas anteriormente.

Con respecto a las funcionalidades que debe seguir la aplicación y para facilitar el uso de las mismas, cada una estará acompañada de una etiqueta que la identificar como caso de uso.

Dispondremos de un usuario no registrado (UNR), un usuario registrado (UR) y un administrador (A), de los cuales obtendremos los siguientes casos de uso:

- CU_1: Registrarse en la aplicación (A, UNR, UR)
- CU_2: Identificarse en la aplicación (A, UR)
- CU_3: Buscar información sobre cualquier película/serie (A, UR)
- CU_4: Ver información de los cines (A, UR)
- CU_5: Ver la cartelera de su país (A, UR)
- CU_6: Votar película (A, UR)
- CU_7: Comentar película (A, UR)
- CU_8: Editar comentarios (A, UR)
- CU_9: Editar mis datos (A, UR)

Como podremos observar en el siguiente diagrama de casos de uso en la figura 5 veremos lo expuesto anteriormente con los actores de nuestra aplicación:

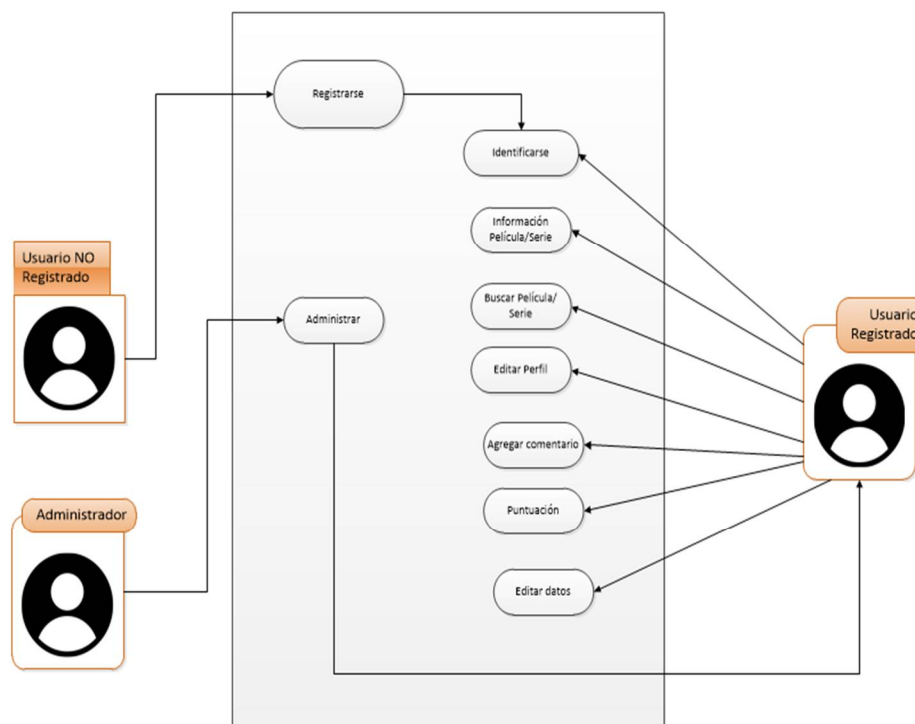


Figura 5 – Diagrama de casos de uso

Como parte de **restricciones** deberemos contar con un ordenador con acceso a internet, con el que poder desarrollar el sistema y mantener el servidor activo, teniendo en cuenta la seguridad con un cifrado de datos de autenticación de los usuarios.

La utilización de una API para obtener información de las películas es una dependencia directa, pudiéndose dar el caso que alguna modificación en las subrutinas genere fichas de que no se muestren correctamente en la aplicación, teniendo en cuenta la continua revisión de dicho posible fallo.

Otra parte a tener en cuenta son los requisitos futuros de nuestra aplicación, pensando en futuras mejoras que podrían aplicarse entre las que podríamos destacar las siguientes:

- Mantenimiento y actualización
- Implementación de un modelo Premium de pago con login de usuarios.
- Una base de datos propia
- Integración en plataformas populares como Prime o Netflix.
- Mejora de ciertos puntos como visualización de trailers en la propia aplicación sin enlace externo.

2.3 Requisitos específicos y casos de uso

Nombre	Diseño web adaptativo o responsive
Referencia	RF01
Caso de Uso	-
Prioridad	Alta
Descripción	Debe ser correctamente visible desde cualquier dispositivo con un navegador, ya sea móvil u ordenador.

Nombre	Alta en la web como usuario
Referencia	RF02
Caso de Uso	CU_1
Prioridad	Alta
Descripción	Poder acceder a la información completa de la web y todas sus funcionalidades en la opción Premium

Nombre	Identificarse en la web
Referencia	RF03
Caso de Uso	CU_2
Prioridad	Alta
Descripción	Estar logeado como usuario para poder acceder a las funcionalidades Premium

Nombre	Buscar información sobre cualquier película o serie
Referencia	RF04
Caso de Uso	CU_3
Prioridad	Alta
Descripción	Obtener información de cualquier película o serie dese la API

Nombre	Ver información e los cines
Referencia	RF05
Caso de Uso	CU_4
Prioridad	Media
Descripción	Obtener información de la cartelera en el momento actual

Nombre	Ver cartelera de su país
Referencia	RF06
Caso de Uso	CU_5
Prioridad	Media
Descripción	Poder obtener información valida de la cartelera de su país

Nombre	Votar película
Referencia	RF07
Caso de Uso	CU_6
Prioridad	Media
Descripción	Poder votar según criterios de la película vista

Nombre	Comentar película
Referencia	RF08
Caso de Uso	CU_7
Prioridad	Media
Descripción	Opinar sobre cualquier película o serie y que sea público al resto de UR

Nombre	Editar comentarios
Referencia	RF09
Caso de Uso	CU_8
Prioridad	Baja
Descripción	Poder cambiar la opinión anteriormente expuesta sobre una serie o película.

Nombre	Editar mi perfil
Referencia	RF10
Caso de Uso	CU_9
Prioridad	Alta
Descripción	Poder editar los datos de mi perfil, como puede ser el nombre, email o cambiar la contraseña.

Por último y no menos importante en este punto sobre los requisitos del sistema, debemos tener en cuenta sus atributos, que en nuestro caso podemos distinguir los siguientes:

- Disponibilidad
- Interacción
- Fiabilidad
- Portabilidad
- Seguridad

El primero es obvio y nuestra aplicación debe estar disponible siempre que el usuario lo necesite y ofrecer una interacción con otros usuarios, esto implica que sea fiable pasando unos test unitarios para garantizar el correcto comportamiento de las funcionalidades. Otro atributo a tener en cuenta es la portabilidad, pudiendo ser accesible desde cualquier navegador de cualquier dispositivo. Y por último y no menos importante es la seguridad en donde los datos de los usuarios se encuentren almacenados, cifrados y protegidos.

3.- Diseño del Proyecto

Para planificar nuestro proyecto web, nos es imprescindible definir distintas tareas que deben realizarse en cada fase del proyecto. Respecto al diseño del proyecto Info Entertainment se ha dividido en los siguientes puntos:

3.1 Diseño del logo

Algo a tener en cuenta es la imagen corporativa y el símbolo que lo representa, tanto a la marca como a la personalidad y para ello está el logo, que es la imagen de tu proyecto.



Figura 6. Logo del proyecto

El diseño se ha elegido de manera simple aplicando un color diferente en cada una de las partes para resaltar más el nombre y destacarlo, haciendo visible la imagen de un reproductor de video y una televisión como referencia a la funcionalidad de la web.

3.2 Arquitectura del sistema

Como podemos observar en la figura 7 que se muestra a continuación, el sistema está basado en el modelo Cliente/Servidor en donde los clientes realizan peticiones de información y el servidor nos responde. Esto se hace para que un usuario no pueda acceder a datos que no está autorizado y mantener recursos e integridad de los datos.

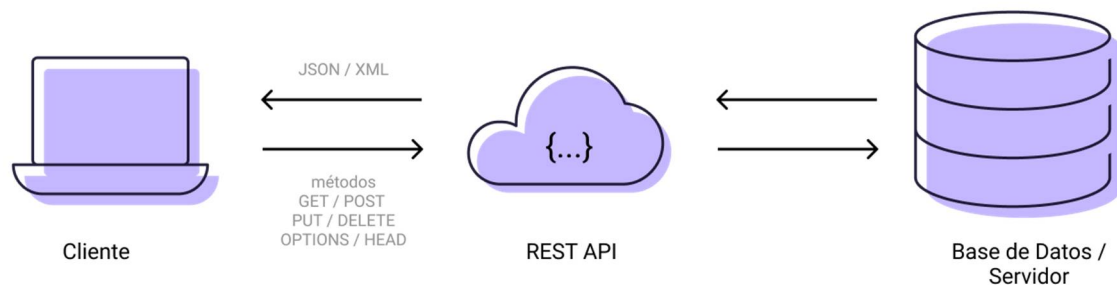


Figura 7 – Ejemplo Cliente/Servidor

A continuación podemos ver un diagrama UML con la configuración básica el cual fue el primer boceto para el diseño de la web:

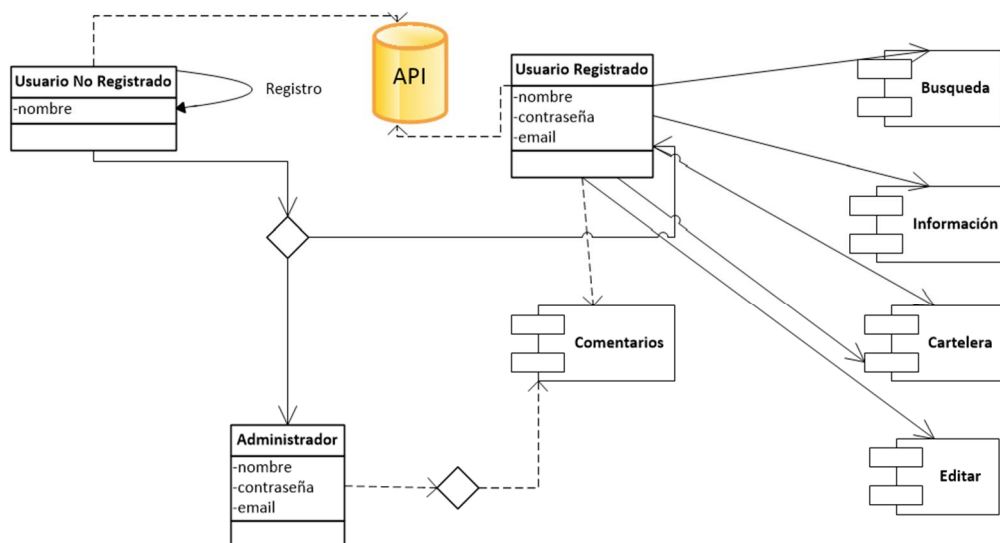


Figura 8 – Primer boceto UML diseño

El proceso de registro de un usuario y a continuación la búsqueda de información de una película o serie puede plasmarse gráficamente (figura 9) de la siguiente manera en donde se explica que se hace en cada paso:

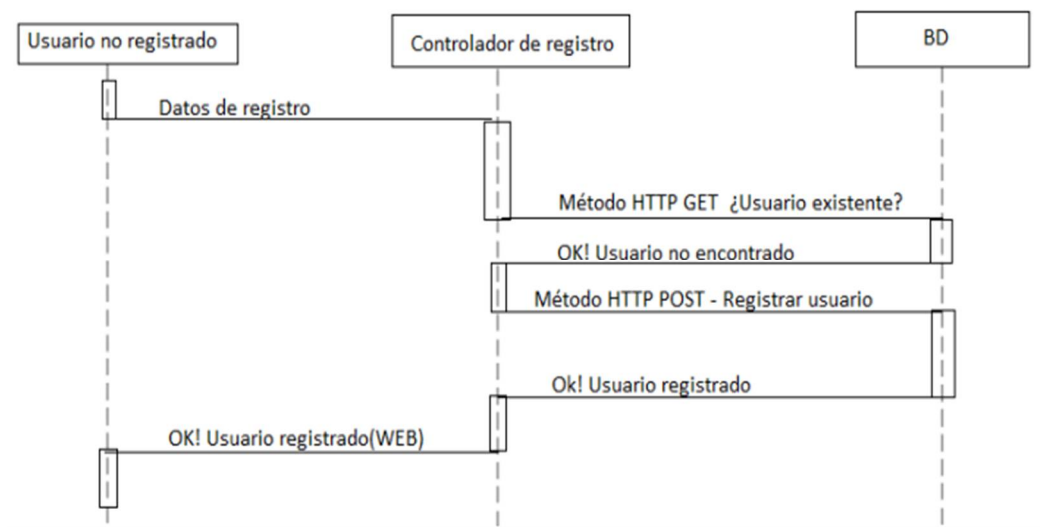


Figura 9 – Proceso de registro de un UNR

Como ejemplo de búsqueda por parte de un UR y el resultado de consultar nuestra API podemos plasmarlo (figura 10) de la siguiente manera gráfica:

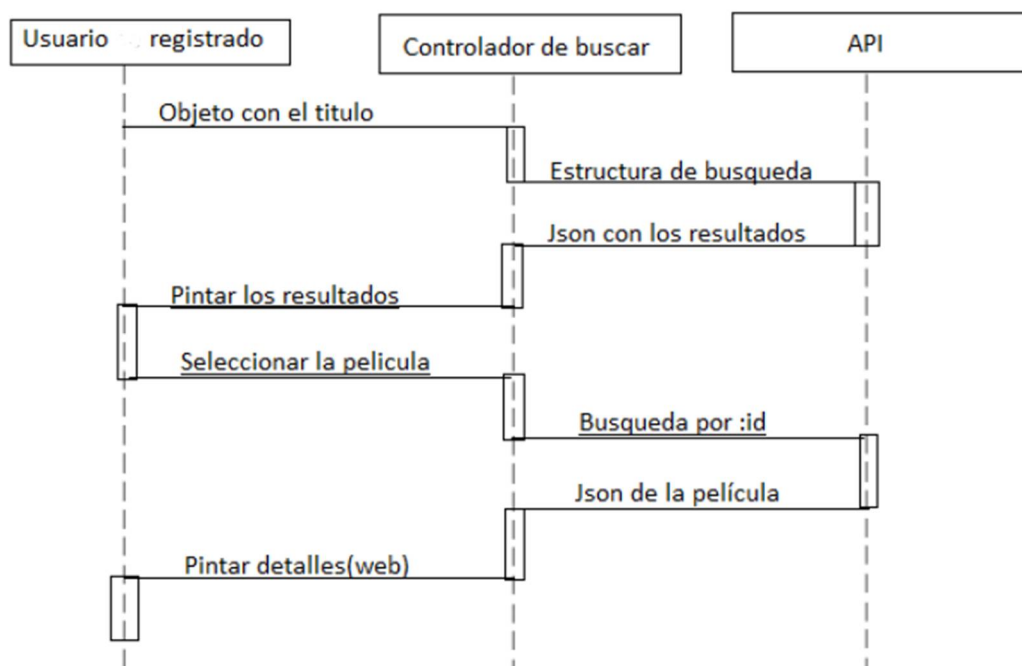


Figura 10– Búsqueda de una película

Una vez definidos los requisitos de la aplicación, pudimos comenzar a pensar en cómo podía diseñarse la aplicación. Para ello se decidió realizar unos bocetos para ver cómo sería gráficamente y poder pasar al desarrollo final e implementación. Hay que tener en cuenta que estos diseños no están hechos

con la intención de ser los diseños finales, sino que están pensados para tener una idea general de cómo se vería la aplicación final, pero a lo largo del desarrollo de cada una de las pantallas se irán añadiendo novedades y cambiando otras. A continuación se muestra el mapa web básico de nuestra aplicación pudiendo sufrir cambios a lo largo del desarrollo:

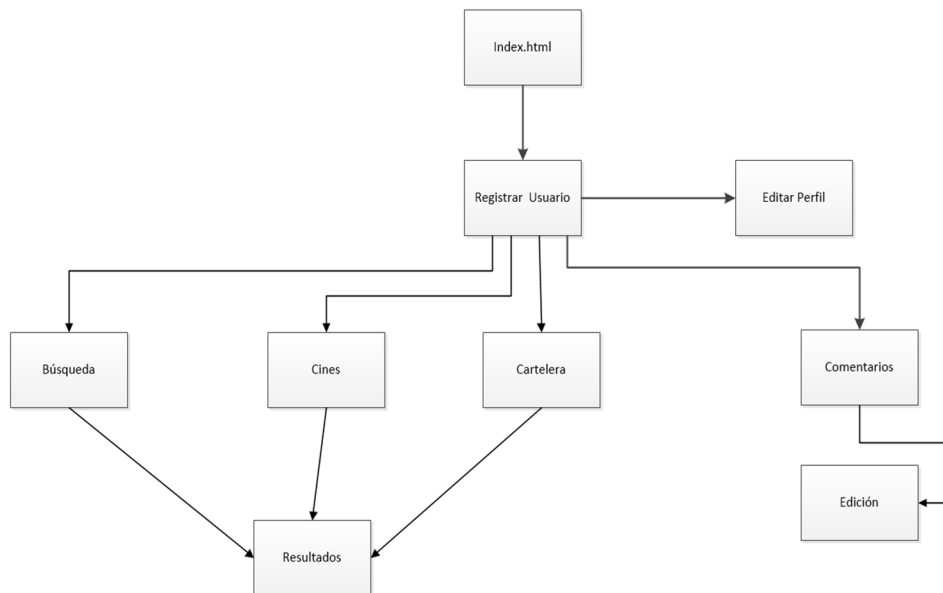


Figura 11 – Mapa Web

A continuación, se puede ver el diseño inicial de la pantalla de bienvenida que se muestra al lanzar la aplicación, para ello usaremos el Low-Fidelity prototyping (Prototipado de baja fidelidad):



A continuación se muestra un ejemplo de cómo podría ser el registro de usuario y el login, en donde se observa los datos necesarios a introducir para poder completar correctamente el registro en la aplicación, aceptando los términos de uso:



The screenshot shows a registration form on a teal background. At the top left is the 'Info Entertainment' logo. The navigation bar includes 'Inicio', 'Registro', 'Información', and a highlighted 'Contacto' button. The main heading is 'BIENVENID@'. The form fields are: 'Nombre Usuario', 'email', 'Contraseña', and 'Repita contraseña'. Below these is a red 'Registro' button. To the right of the button is a checked checkbox for 'Acepto los terminos de uso' with a link to 'Ver Terminos'.

El proceso de login puede ser el siguiente, en donde el usuario introduce su nombre y su contraseña para poder acceder a las funcionalidades:

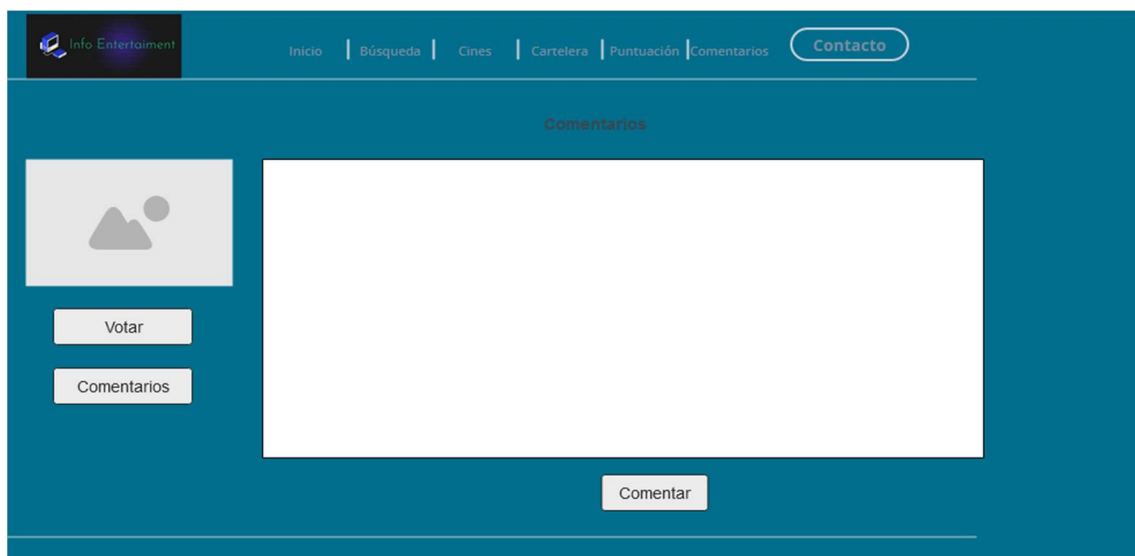


The screenshot shows a login form on a teal background. The navigation bar includes 'Inicio', 'Búsqueda', 'Cines', 'Cartelera', 'Puntuación', 'Comentarios', and a highlighted 'Contacto' button. The main heading is 'BIENVENID@'. The form fields are: 'Nombre Usuario' (containing 'usuario') and 'Contraseña' (containing 'contraseña'). Below these is a red 'Entrar' button. At the bottom is a checked checkbox for 'Recordarme'.

Una vez logeado y dentro de la aplicación podremos ver las distintas funcionalidades, a continuación se muestra un ejemplo genérico de cómo podrían ser el resultado final, en donde se mostrarían las películas:



en esta última captura se puede observar cómo sería lo deseado conforme a la pantalla de información de nuestra película seleccionada.



como última captura podemos observar cómo sería nuestra página de comentarios y votaciones, en donde se observa los botones de lanzar la votación y redactar el comentario deseado sobre nuestra película seleccionada.

Los bocetos serán una guía a la hora de hacer el desarrollo de la aplicación y a la vez un ejemplo de cómo estructurar la página principal, y el resto de páginas desde el login hasta los comentarios.

4. Desarrollo e implementación del proyecto

Como se ha explicado en este documento, el desarrollo de este proyecto tiene como objetivo la creación de una plataforma web de búsqueda que permita a un usuario final poder visualizar y buscar diferentes títulos tanto actuales como no actuales. A continuación se explicará cómo se ha llevado a cabo el proceso de desarrollo de dicha web.

Desde un principio se ha desarrollado el buscador pensando en la usabilidad y en el funcionamiento de una plataforma con un registro de usuario pero según avanzaba el proyecto y según el estudio realizado a un número determinados de usuarios, se ha planteado eliminar la opción de registro dado el rechazo de la mayoría de los usuarios consultados sobre tener que registrarse para una aplicación en la que su función principal es la búsqueda ágil de películas y su información, por lo que se detallará en los cambios que se propondrán respecto a la planificación inicial.

Con tal de facilitar la búsqueda a los usuarios, la aplicación web cuenta con una serie de menús sobre, estrenos, lo más visto y lo más popular para poder orientarlo a la hora de decidirse por seleccionar una película determinada.

Por otra parte, para que los usuarios puedan moverse por la interfaz sin tener problemas ni dificultades, la aplicación cuenta con un menú en la parte superior de la pantalla que contiene los siguientes enlaces:

- Inicio con enlace directo desde el logo en donde se muestra un resumen de lo más popular y lo mejor valorado en votos.
- Enlace a las última novedades
- Enlace a lo más popular
- Realizar búsqueda personalizada
- Zona no funcional de registro
- Contacto
- Ejemplo de funcionamiento

Cabe destacar que los últimos enlaces no son funcionales, sería una muestra de cómo funcionaría la web con registro y el límite de acceso que se tendría sin dicho registro, descartado por la incomodidad que presento dicha funcionalidad a los usuarios.

Otro punto a destacar, es la mejora de la aplicación web de punto que se tenían previsto realizar, pero que por falta de tiempo, no se han podido

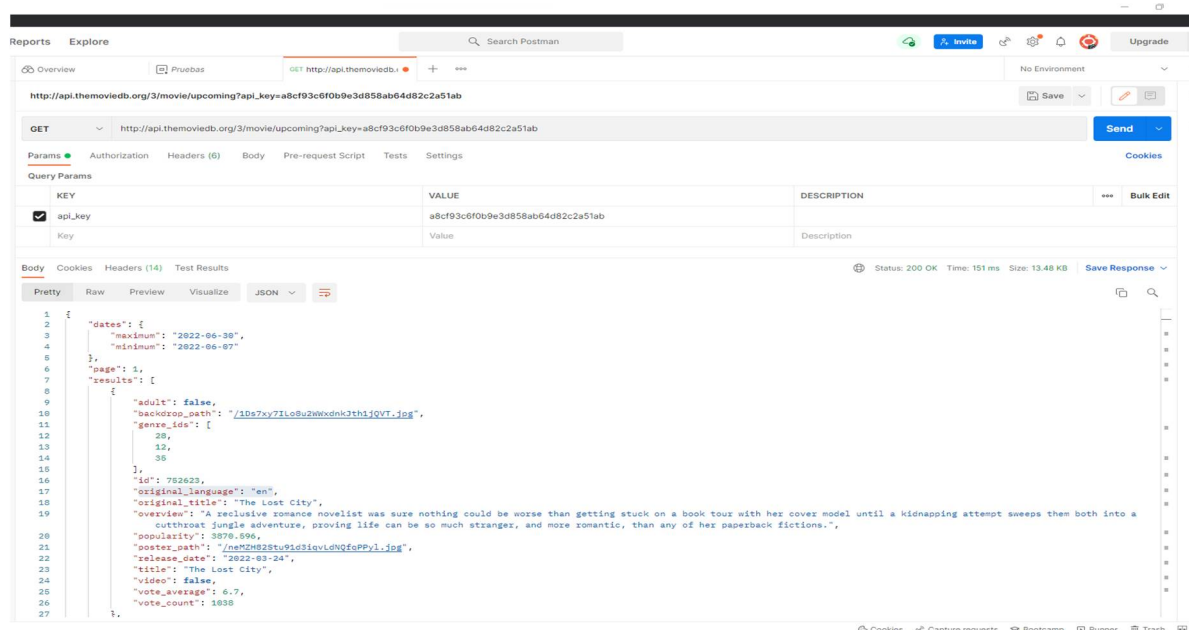
completar. Las características se definen a continuación y los problemas encontrados:

- Registro de usuarios y administrador, descartado por pruebas con usuarios finales, que rechazaban el registro y dar información de contacto, aparte del uso de tecnologías no previstas en dicha planificación como es la parte BackEnd y el lenguaje php.
- Mejora de la interfaz y hacerla 100 % responsive.
- Añadir la función de comentarios y votos, necesitaría lo descrito anteriormente y se necesitaría más tiempo para formarse y aplicar las tecnologías necesarias para llevarlo a cabo.
- Añadir filtros para la búsqueda
- Buscar una API con mucha más información y con series en su base de datos, que las previstas solo trabajan con películas.
- Añadir idiomas.

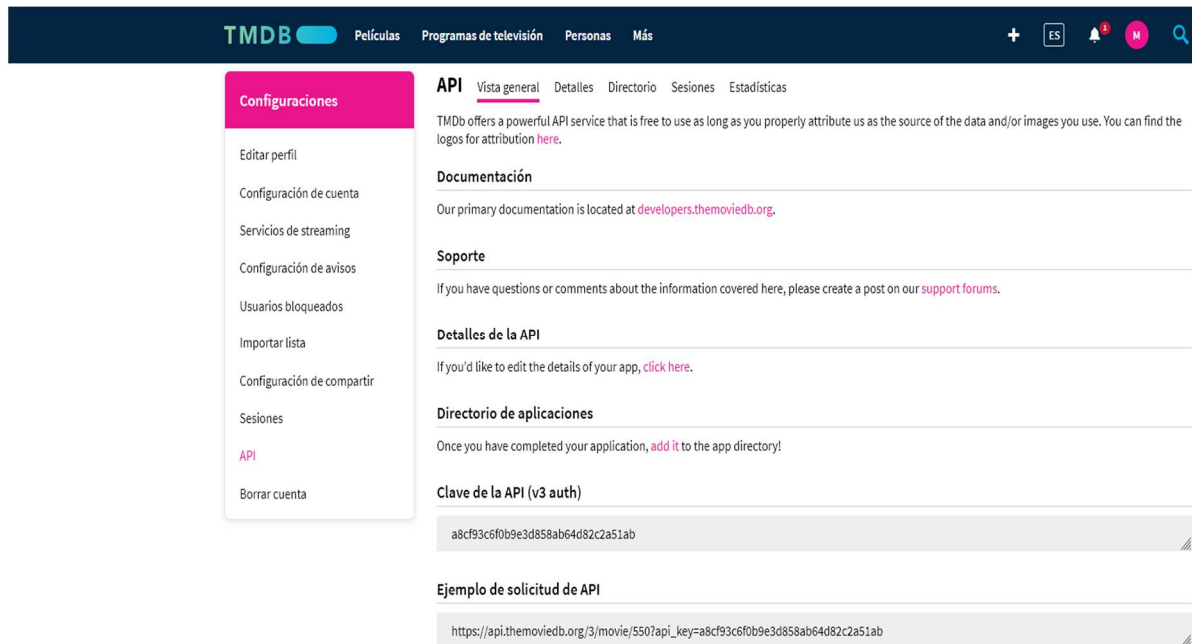
4.1 Lenguajes de programación usados, API y software necesario.

Lo primero a tener en cuenta es un servidor web, en nuestro caso Apache [1]. Los servidores web se encuentran alojados en un ordenador que posea conexión a internet a la espera de que algún navegador le realice una petición para así, atender la solicitud. Para nuestro proyecto instalamos Xampp [2].

Postman es un programa que cuenta con una interfaz gráfica sencilla que permite el diseño, "mock" o simulacro, debug o depurado, la realización de pruebas, documentación, monitorización y publicación de APIs, todo desde un único sitio. Para este proyecto se ha utilizado Postman únicamente para la realización de pruebas sobre el API del servicio. A continuación vemos una captura de una de las pruebas realizadas con el programa en la que realizamos un GET a la API seleccionada para nuestra aplicación y vemos como resultado nos devuelve la información solicitada, en este caso:



Con respecto a la API se ha elegido la siguiente: <https://www.themoviedb.org/documentation/api>, teniendo que registrarnos para poder tener acceso a ella, obteniendo una key como se muestra a continuación:



Esta API nos permite a los desarrolladores obtener información sobre películas solamente dejando de lado la funcionalidad de series, pero obteniendo muchas más funcionalidades que otras APIs expuestas en la planificación y usando una interfaz muy amigable para el usuario final, que lo que realmente quiere es buena información pero concisa.

Para esto usaremos un método GET sobre el endpoint “/configuration”, el cual obtiene los principales datos de configuración de la base de datos de TMDb. Esta respuesta se devuelve en un objeto JSON con los siguientes datos:

- La URL base para la obtención de imágenes.
- Los tamaños de los distintos tipos de imágenes, tanto como del contenido de películas y series (pósters, backdrops y fotos de perfil), como el tamaño de los logos de las productoras.

Esta información servirá en las siguientes llamadas para poder formar las URLs que se usarán para obtener las imágenes de los distintos títulos.

Dentro de los datos obtenidos de la API destacamos los siguientes:

- Información de la película (Método GET sobre el endpoint “/movie/{movie_id}”).

```
const getMovieDetails = (movieId) => {  
  const url = `${URL_PATH}/3/movie/${movieId}?api_key=${API_KEY}&language=es-ES`;  
  
  return fetch(url)  
    .then(response => response.json())  
    .then(result => result)  
    .catch(error => console.log(error))  
}
```

- Las más populares (Método GET sobre el endpoint “/movie/popular”).

```
const getPopularMovies = (page) => {  
  const url = `${URL_PATH}/3/movie/popular?api_key=${API_KEY}&language=es-ES&page=${page}`;  
  
  return fetch(url)  
    .then(response => response.json())  
    .then(result => result.results)  
    .catch(error => console.log(error))  
}
```

- Las más votadas (Método GET sobre el endpoint “/movie/top_rated”).
- Actualmente en cartelera (Método GET sobre el endpoint “/movie/now_playing”).
-

```
const getNewsMovies = (page) => {  
  const url = `${URL_PATH}/3/movie/now_playing?api_key=${API_KEY}&language=es-ES&page=${page}`;  
  
  return fetch(url)  
    .then(response => response.json())  
    .then(result => result.results)  
    .catch(error => console.log(error))  
}
```

- Obtención de las imágenes de las películas (Método GET sobre el endpoint “/movie/images”)

- Por último usaremos la funcionalidad de búsqueda que nos ofrece la API.

```

1  const URL_PATH = "https://api.themoviedb.org";
2  const API_KEY = "a8cf93c6f0b9e3d858ab64d82c2a51ab";
3
4  const searchMovies = async () => {
5    const textSerach = document.getElementById("search-movie").value;
6    if (textSerach.length < 3) {
7      return;
8    }
9
10   const movies = await getMovies(textSerach);
11
12   let html = "";
13   movies.forEach(movie => {
14     const { id, title, overview, poster_path } = movie;
15     const urlMoreInfo = `../pelis.html?id=${id}`;
16     const urlImage = `https://image.tmdb.org/t/p/w500${poster_path}`;
17
18     html += `
19       <div class="col-4 custom-card">
20         <div class="card">
21           <div class="row no-gutters">
22             <div class="col-md-4">
23               
24             </div>
25             <div class="col-md-8">
26               <div class="card-body">
27                 <h5 class="card-title">${title}</h5>
28                 <p class="card-text">${overview.substr(0, 40)}...</p>
29                 <a href="${urlMoreInfo}" class="btn btn-primary">Ver más</a>
30               </div>
31             </div>
32           </div>
33         </div>
34       </div>
35     `;
36   });
37   document.getElementsByClassName("list-cards")[0].innerHTML = html;
38 }
39
40 const getMovies = textSerach => {
41   const url = `${URL_PATH}/3/search/movie?api_key=${API_KEY}&language=es-ES&query=${textSerach}&page=1&include_adult=true`;
42
43   return fetch(url)
44     .then(response => response.json())
45     .then(result => result.results)
46     .catch(error => console.log(error));
47 };

```

Continuamos con la página principal o index.html en la que se muestra el inicio de nuestra aplicación web ofreciéndonos de primeras información sobre lo más popular y lo más votado como se puede ver a continuación en la siguiente captura:

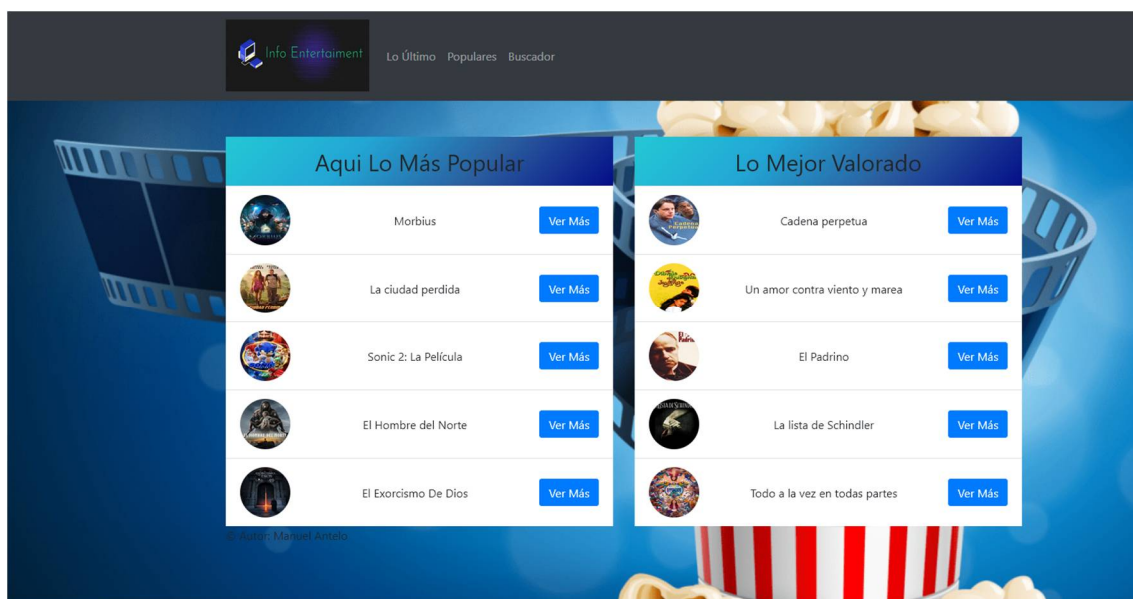


Figura 12– index.html

Como se puede observar tenemos un menú superior con los enlaces: Lo último, Populares y Buscador, haciendo la imagen como enlace directo siempre a la página principal, en la que podemos ver lo explicado anteriormente y acceder a la información de cada película. El fondo de nuestra aplicación web ha sido sacado del siguiente enlace [3] en donde se puede observar la multitud de imágenes disponibles.

Si hacemos clic en lo último veremos que nuestra aplicación nos muestra una serie de películas como se puede observar en la figura 13:

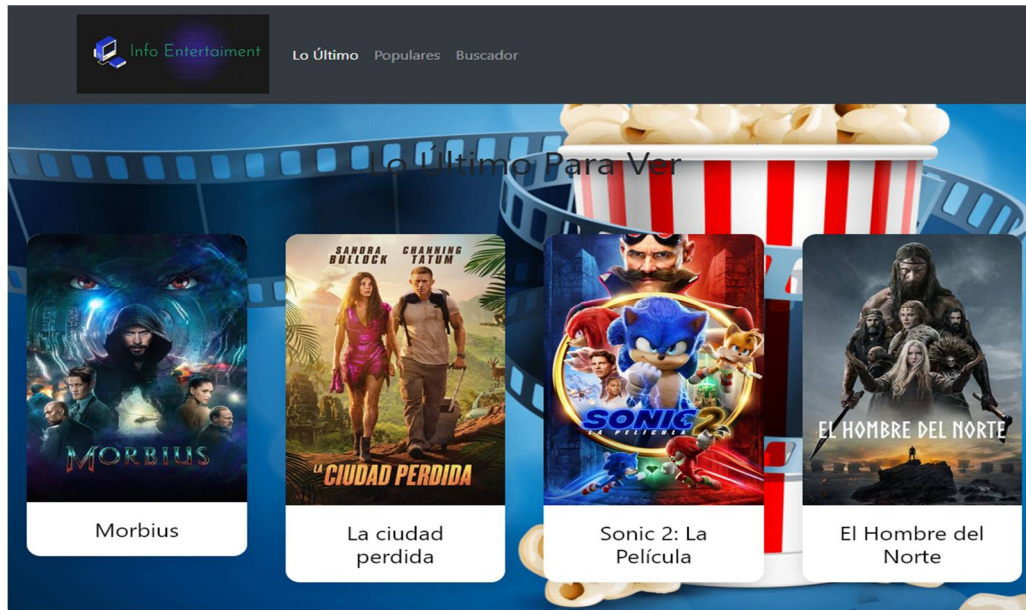


Figura 13 – novedades.html

y si hacemos clic en la imagen o el título nos llevará a otra página en donde será detallada la información de la película seleccionada (figura 14) y pudiéndose observar como el fondo es el de la citada película.

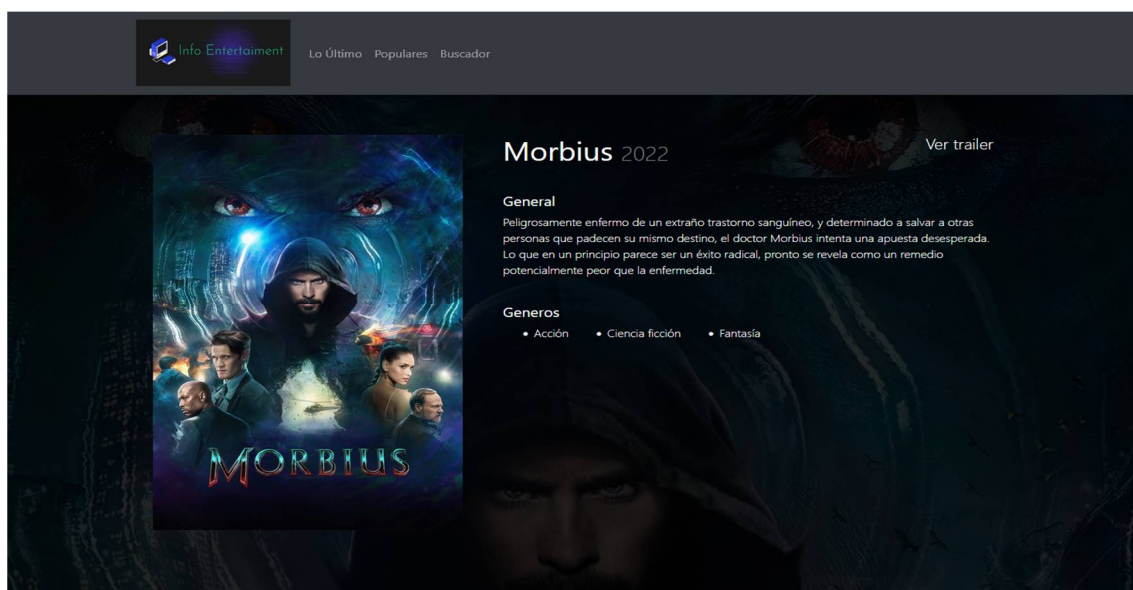


Figura 14 – pelis.html

Lo mismo nos va suceder con el menú populares que se puede observar en la figura 15 que mostrara las más populares hasta el momento, en nuestro caso, las primeras coinciden con las novedades como era de esperar, pero se observa bajando que la búsqueda muestra otras opciones.

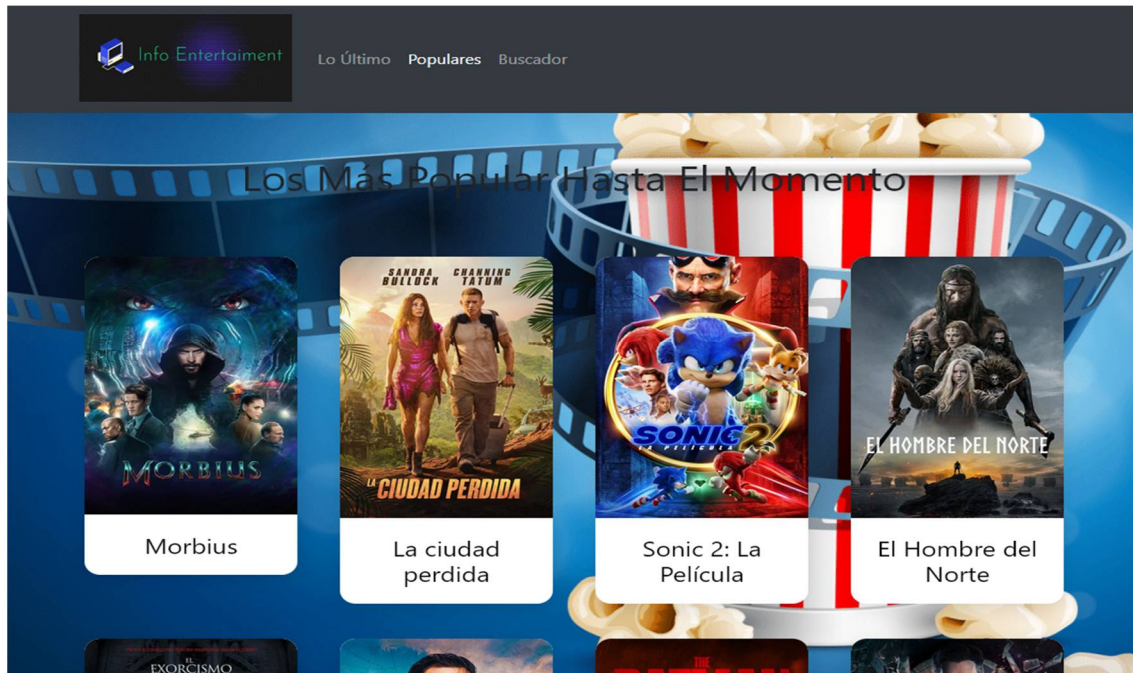


Figura 15 – popular.html

Por último y lo más importante del desarrollo de esta aplicación web es el buscador en el cual podemos hacer una búsqueda del título (ver figura 16) que queramos como se puede observar a continuación:

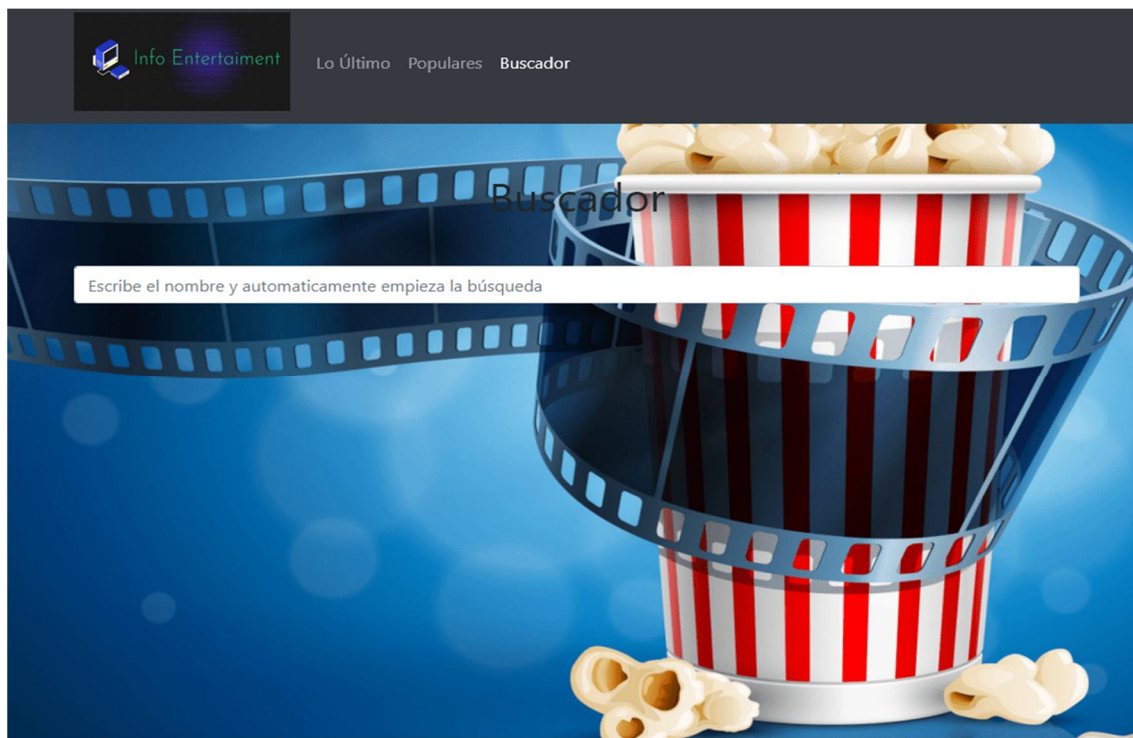


Figura 16 – buscador.html

Al introducir el nombre nos va ir mostrando los posibles resultados (ver figura 17) que coincidan con el nombre escrito hasta encontrar el título deseado

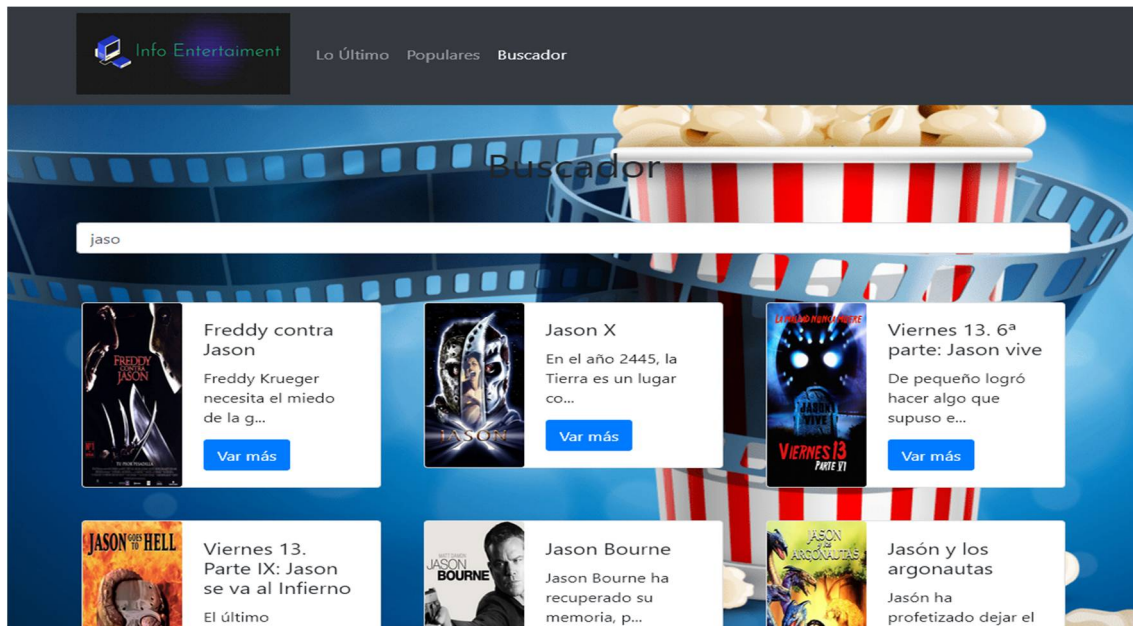
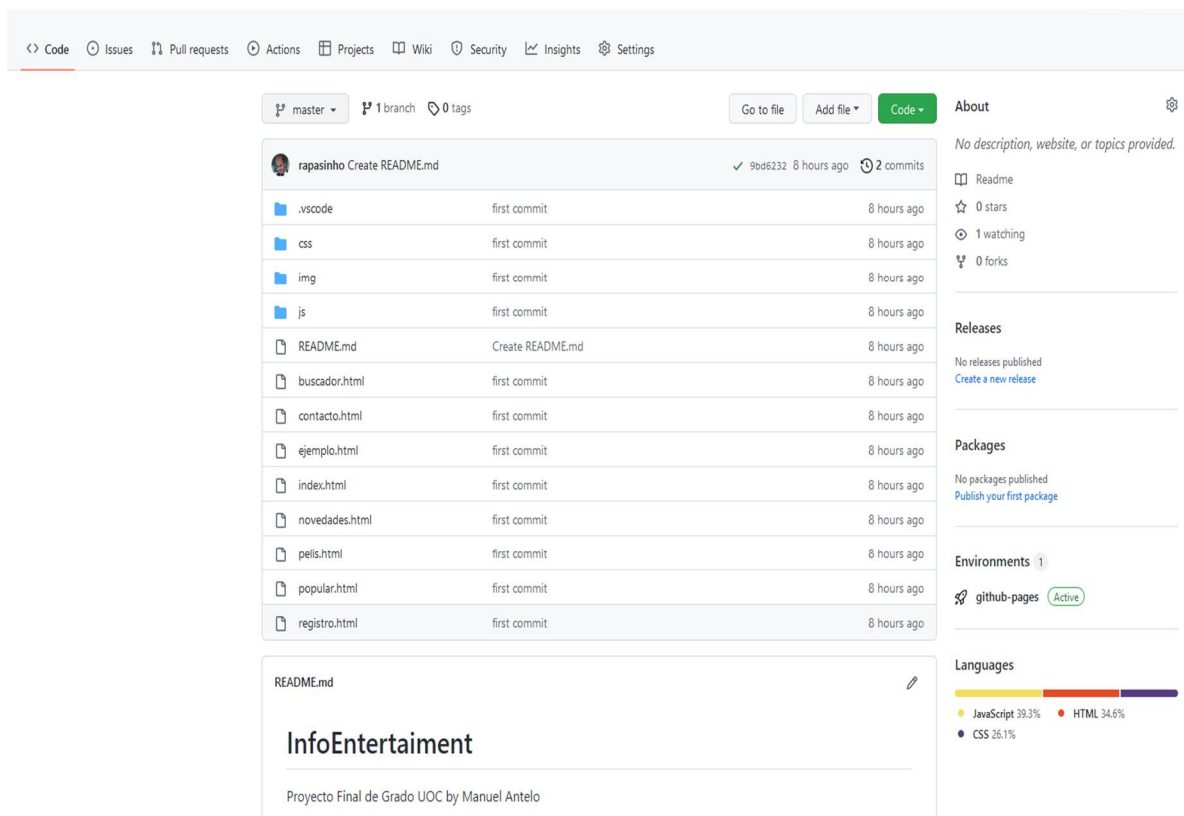


Figura 17 – Búsqueda

Toda esta aplicación, tanto su desarrollo como su implementación se ha realizado utilizando los lenguajes HTML, CSS, Javascript, Bootstrap y las herramientas Git y GitHub como se puede observar a continuación que es donde está el código de nuestro proyecto:



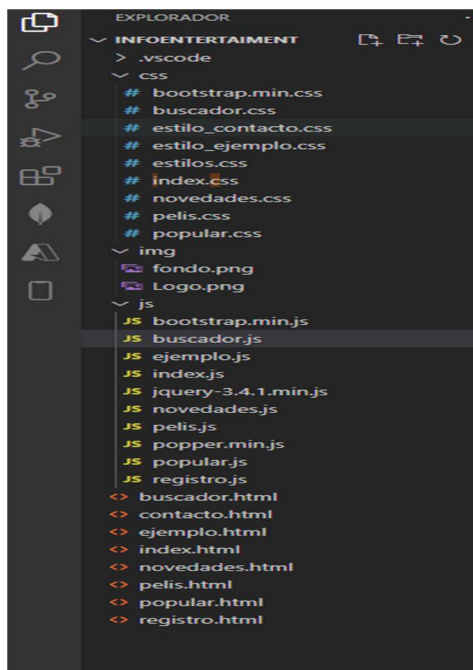
El código del proyecto lo podremos encontrar en el siguiente enlace:

<https://github.com/rapasinho/InfoEntertainment>

y la funcionalidad que nos permite Github, desde el siguiente enlace:

<https://rapasinho.github.io/InfoEntertainment/>

Toda la implementación ha sido realizada en visual studio code, separando cada página con sus respectivos archivos de javascript y css como se puede observar en la siguiente figura:



5. Pruebas del proyecto

La fase de evaluación y pruebas concluye el ciclo de vida de nuestro proyecto y lo prepara para subirlo al servidor y que se ejecute. Esta fase mide el nivel de calidad que ofrece al usuario la aplicación creada. Las herramientas que vamos a usar para realizar estas pruebas son gratuitas y las podemos encontrar online en las direcciones:

<http://jigsaw.w3.org/css-validator/>

<http://validator.w3.org/>

Dado que se han utilizado hojas de estilo para la interfaz de la aplicación, se ha realizado una prueba de validación de CSS. Para ello, se ha utilizado la página web <http://jigsaw.w3.org/css-validator/>. En cuanto al resultado lo podemos

observar en la figura 18, se podría decir que la hoja de estilo ha pasado la validación:



El Servicio de Validación de CSS del W3C

Resultados del Validador CSS del W3C para <https://rapasinho.github.io/InfoEntertainment/> (CSS versión 3 + SVG)

Ir a:
[Las Advertencias \(656\)](#)
[Su Hoja de Estilo validada](#)

Resultados del Validador CSS del W3C para <https://rapasinho.github.io/InfoEntertainment/> (CSS versión 3 + SVG)

¡Enhorabuena! No error encontrado.

(Este documento es [CSS versión 3 + SVG](#) válido)

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preocupado por crear una página Web interoperable. A continuación se encuentra el XHTML que puede usar para añadir el icono a su página Web:

```
<p>
<a href="https://jigsaw.w3.org/css-validator/check/referen">

</a>
</p>
```

```
<p>
<a href="https://jigsaw.w3.org/css-validator/check/referen">

</a>
</p>
```

(cierra la etiqueta img con > en lugar de >> si utiliza HTML <= 4.01)



Interested in "developing" your developer skills? In W3C's hands-on Professional Certificate Program, learn how to code the right way by creating Web sites and apps that use the latest Web standards.

[Find out more!](#)

[Donate](#) and help us build better tools for a better web.

Si lo desea, puede descargar una copia de la imagen para guardarla en su directorio web local y cambiar el fragmento anterior de XHTML para referenciar a la imagen en local en lugar de a la de éste servidor.

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change.

Showing results for https://rapasinho.github.io/InfoEntertainment/

Checker Input

Show

☒ source

☒ outline

☒ image report

Options...

Check by

address

https://rapasinho.github.io/InfoEntertainment/

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

Document checking completed. No errors or warnings to show.


Image report

The 1ng elements of the page are shown below categorized by their type of textual alternative. Please review that the images in each group match that group's definition.

Images with textual alternative

The following images have textual alternatives. Please review that the textual alternatives make sense considering the purpose of the image in the context of the page and that phrases like "image of ..." are avoided.

Note that iconic images that are redundant with text next to them or purely decorative should have alt="" instead.

Image	Textual alternative	Location
	Início	From line 25, column 17, to line 25, column 82

Source

```

1. <!DOCTYPE html>
2. <html lang="es">
3.
4. <head>
5.   <meta charset="UTF-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7.   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8.   <title>Info Entertainment by Manuel Antelo</title>

```

Figura 18 – Resultado prueba css

Las pruebas en navegadores distintos han resultado óptimos y por último las pruebas de búsqueda de películas de otros países ha resultado satisfactoria

como se muestra en la figura 19 en una búsqueda con un título español y como resultado el siguiente:

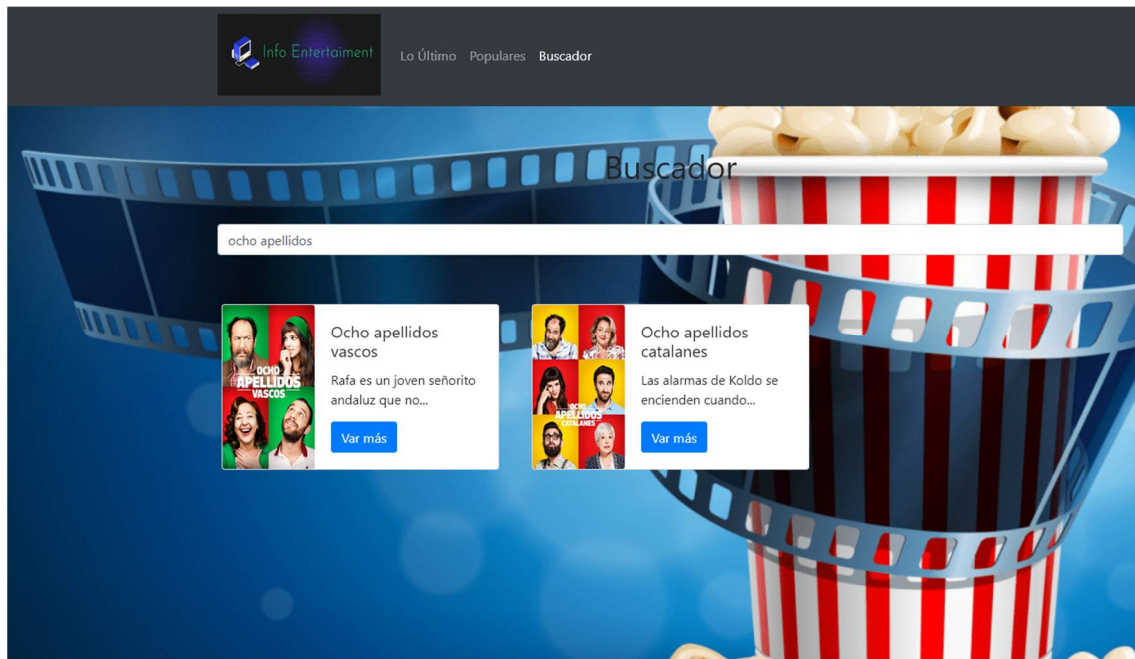


Figura 19 – Búsqueda títulos españoles

Las pruebas de software son una parte integral del ciclo de vida del desarrollo de software. Las pruebas son la forma en que puede estar seguro acerca de la funcionalidad, el rendimiento y la experiencia del usuario. Ya sea que realice sus pruebas manualmente o a través de la automatización, cuanto antes y más a menudo pueda llevar a cabo pruebas, más probable es que identifique errores y errores, no sólo ahorrándole a usted y a su equipo de posibles simulacros de incendio más adelante, sino también asegurándose de que su aplicación de software haya sido revisada y auditada a fondo antes de que esté frente a sus usuarios. Si los problemas se arrastran al entorno de producción, los más caros y lentos que van a solucionar.

Las pruebas de software se pueden dividir en dos tipos diferentes: pruebas funcionales y no funcionales.

Pruebas funcionales: Las pruebas funcionales se llevan a cabo para comprobar las características críticas para el negocio, la funcionalidad y la usabilidad. Las pruebas funcionales garantizan que las características y funcionalidades del software se comportan según lo esperado sin ningún problema. Valida principalmente toda la aplicación con respecto a las especificaciones. Los tipos de pruebas funcionales incluyen pruebas unitarias, pruebas de interfaz, pruebas de regresión, además de muchas. En nuestro caso podemos comprobar como nuestra web se comporta a la hora de acceder a la información de una película determinada y ver el resultado que obtenemos como se muestra en la siguiente figura:

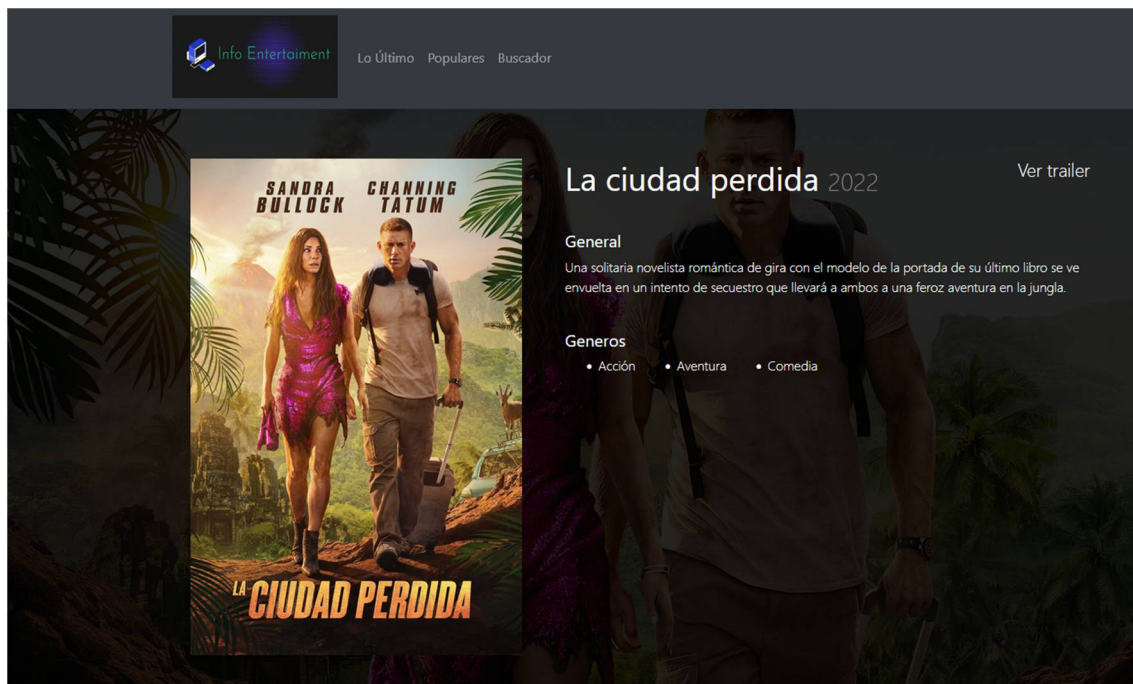
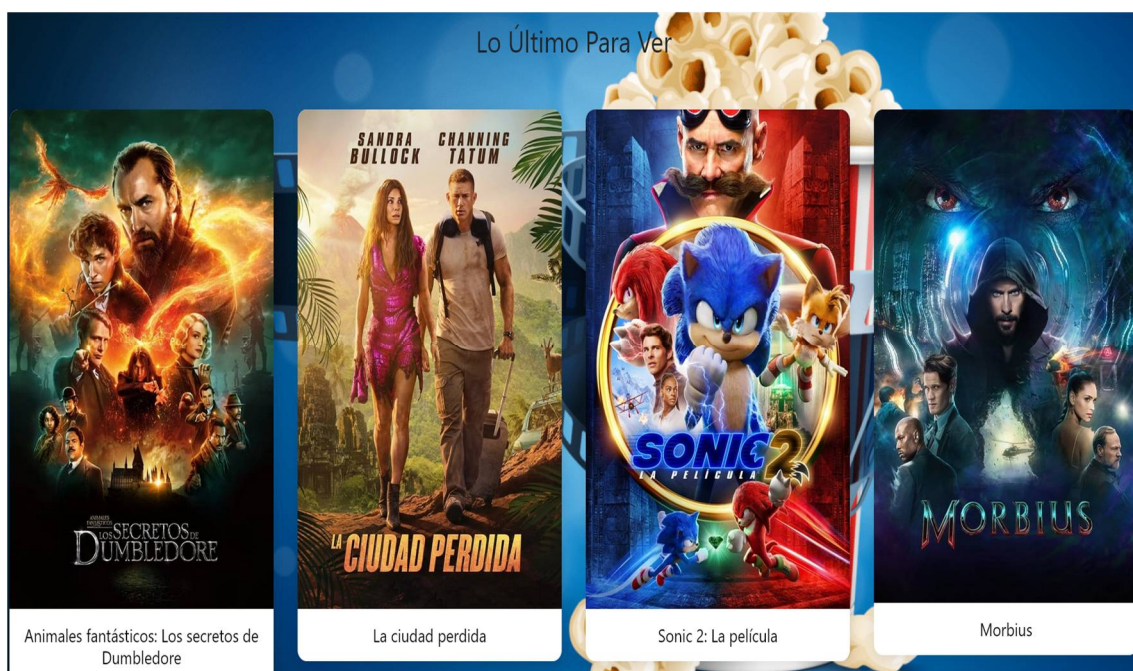
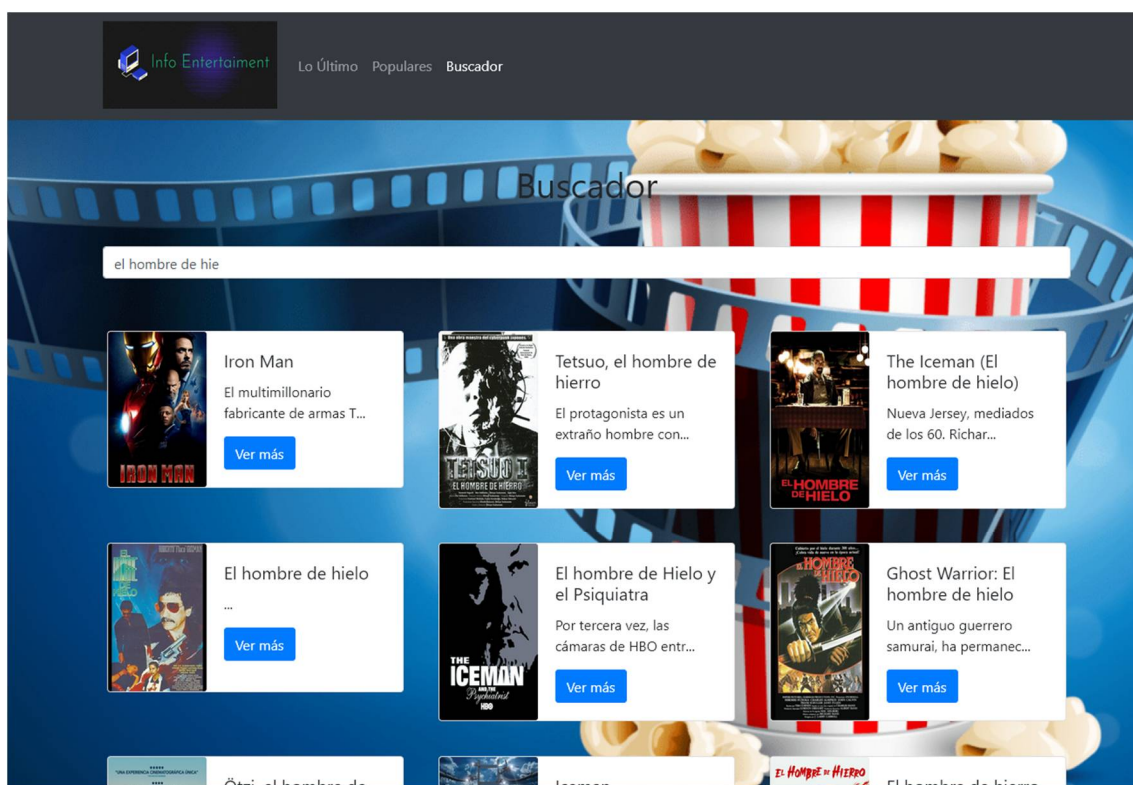
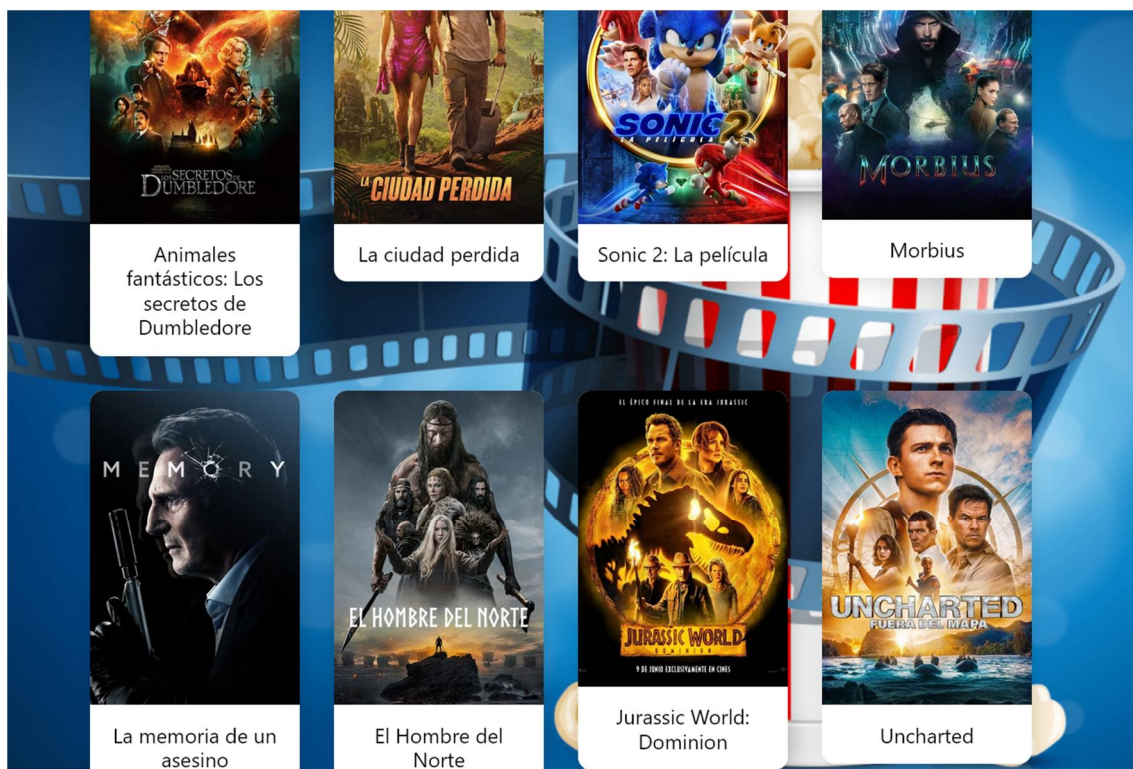


Figura 20 – Test información película

Pruebas unitarias: Las pruebas unitarias se centran en probar piezas/unidades individuales de una aplicación. Cualquier función, procedimiento, método o módulo puede ser una unidad que se someta a pruebas unitarias para determinar su corrección y comportamiento esperado. Las pruebas unitarias son las primeras pruebas que los desarrolladores realizan durante la fase de desarrollo. Para nuestra aplicación haremos la prueba de las distintas funcionalidades obteniendo el siguiente resultado:

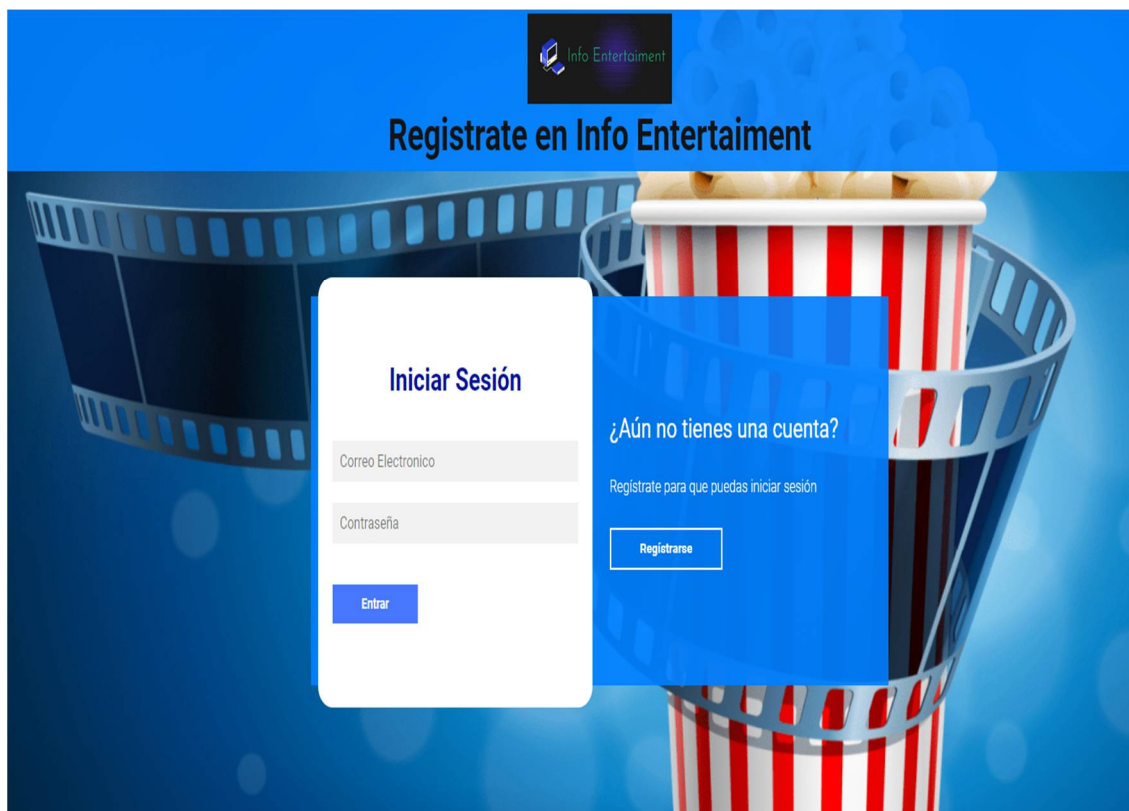




Pruebas no funcionales: Las pruebas no funcionales son como pruebas funcionales; sin embargo, la principal diferencia es que esas funciones se prueban bajo carga para el rendimiento de los observadores, fiabilidad, usabilidad, escalabilidad, etc. Las pruebas no funcionales, como las pruebas de carga y esfuerzo, normalmente se llevan a cabo mediante herramientas y soluciones de automatización, las cuales no disponemos de ellas. Además de las pruebas de rendimiento, los tipos de pruebas no funcionales incluyen pruebas de instalación, pruebas de confiabilidad y pruebas de seguridad.

6. Resultados y conclusiones

EL proyecto está cumpliendo con el principal objetivo del trabajo, que es el diseño y desarrollo de un portal web para la búsqueda de información sobre películas, con añadidos que la hacen diferente a otras plataformas. Sin embargo, los objetivos planteados inicialmente no se han podido cumplir con la precisión que se hubiese gustado (el tiempo es justo y al ser una persona sola encargada de un proyecto, se acaba encontrando con dificultades que se tarda en solventarlas), en este caso faltaría plasmar alguna funcionalidad ya de un prototipo de web pero la falta de experiencia con ciertos lenguajes ha retrasado dicha información, que queda para futuras mejoras en donde ya se puede ver por completo tanto el desarrollo Frontend como el desarrollo Backend necesario para por ejemplo el login de usuario y registro que en el desarrollo FrontEnd nos quedaría así:



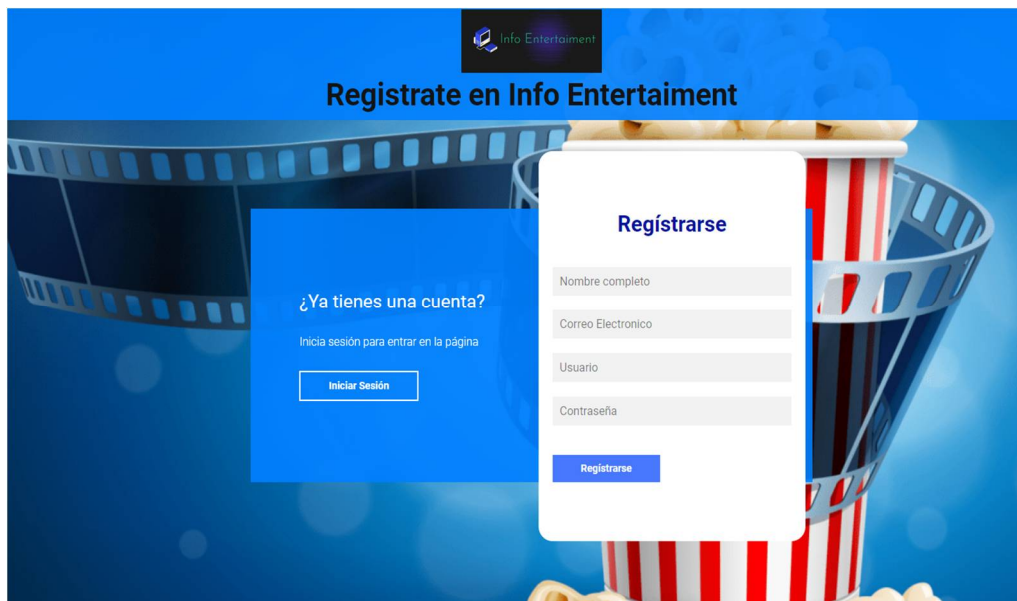


Figura 21 – Registro y Login

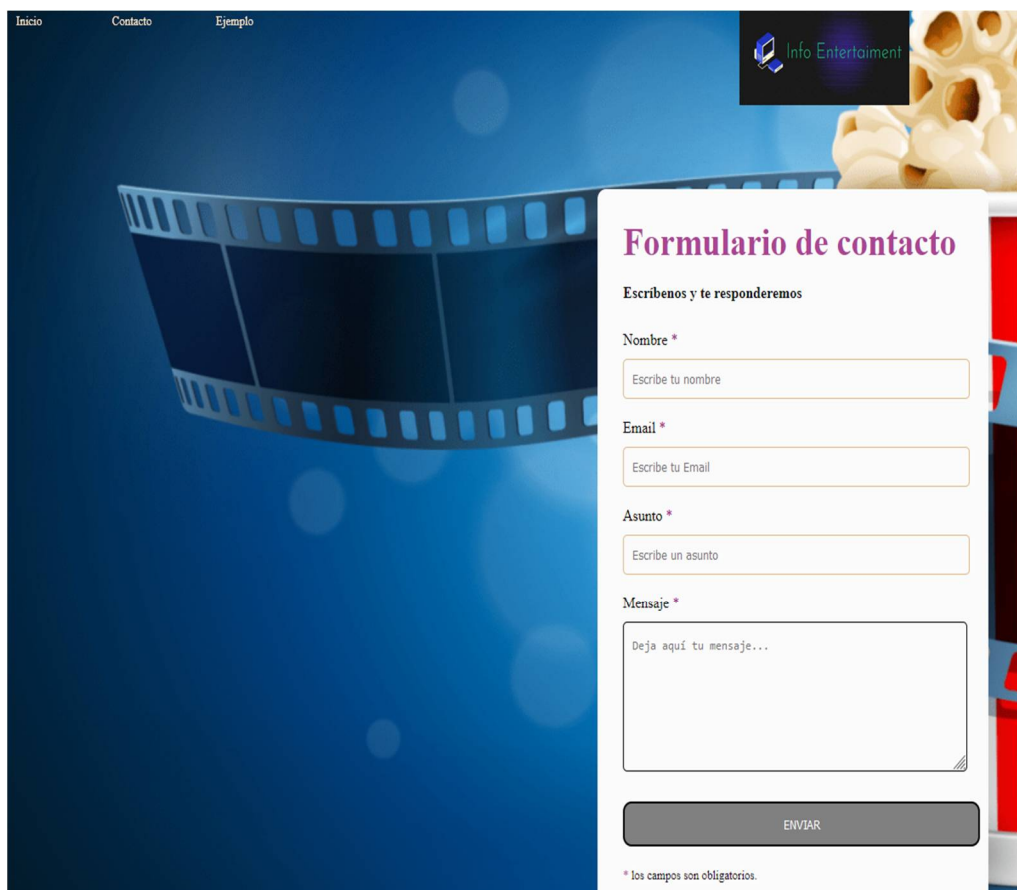


Figura 22 – Formulario de contacto para no registrados

Como se observa en el diseño de la página de contactos, tenemos un formulario para poder solicitar información sobre nuestra aplicación y enviarla a la persona encargada de dicho apartado y que este sea respondido al usuario.

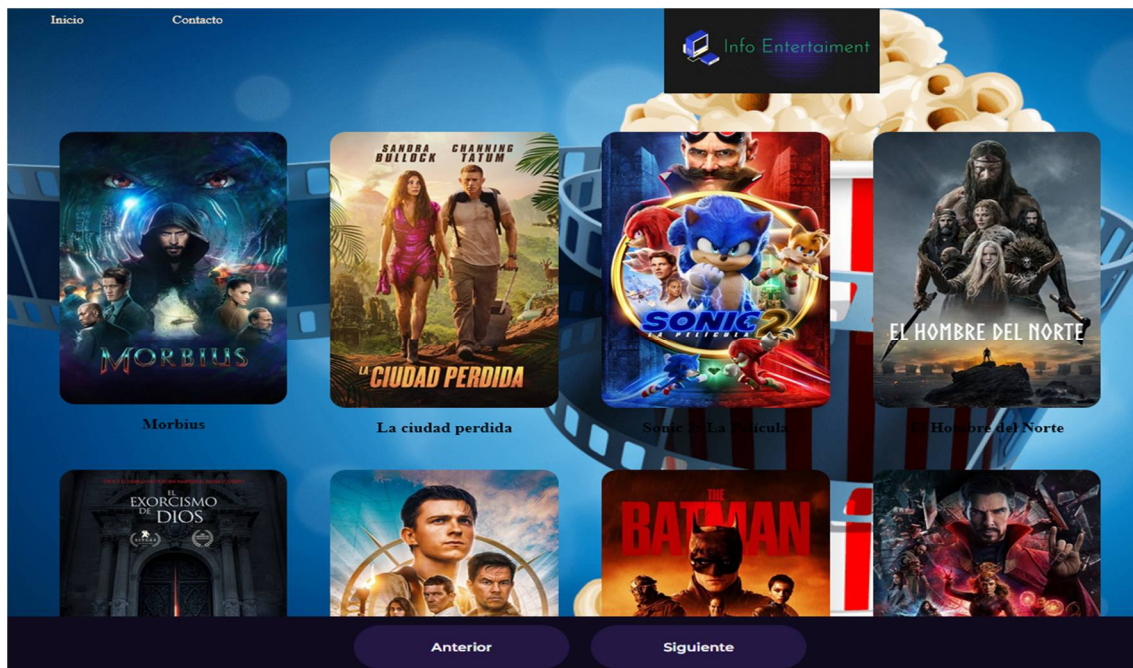


Figura 23– Ejemplo de funcionamiento para no registrados

En cuanto al diseño, se logra una experiencia de usuario óptima y sencilla, con un diseño acorde al planteado al principio. El diseño web adaptable o responsive está dando algunos problemas en dispositivos pequeños (teléfonos móviles), que se pretende sean resueltos en breve dada la falta de experiencia, en lo que respecta a navegadores en PC funciona perfectamente.

El desarrollo es quizás la parte más compleja, teniendo que realizar numerosas pruebas y cambios sobre la marcha, teniendo que posiblemente eliminar algunas funcionalidades que se tenían previstas implementar (la opción de votación por ejemplo y mostrar el tráiler de una película/serie).

La planificación al principio se desarrolló correctamente siguiendo la planificación inicial (modelado, diagramas, casos de uso), pero cuando fue necesario la instalación de las herramientas para el desarrollo fue retrasando la planificación, al tener que dedicar más tiempo del planificado para intentar cumplir con los objetivos e ir haciendo un diseño óptimo, pero a la vez dedicarle más tiempo a la preparación de las herramientas y formación en los lenguajes de programación web necesarios para el desarrollo del proyecto, descartando React y Redux y que con los expuestos anteriormente fue suficiente para lograr el objetivo propuesto inicialmente.

Hasta el momento aparte de las dificultades lógicas cuando se afrontan nuevas tecnologías, una de las más grandes hasta ahora en este proyecto ha sido tener que afrontar todas las fases del proyecto una misma persona, lo que provoca la falta de visión de errores que se podrían haber solventado de manera más temprana y evitar pérdida de tiempo, y la más importante el poco tiempo para formarse en varias tecnologías que a su vez son muy distintas y con múltiples opciones de formación y uso.

Como conclusión, puedo afirmar que la aplicación tiene un gran potencial de desarrollo, cumpliendo las condiciones necesarias para facilitar su usabilidad. Por último, se puede afirmar que es un gran reto personal este proyecto y que a día de hoy sigo entusiasmado con él y con ganas de poder dedicarle más tiempo para poder completar las funcionalidades que faltan y adentrarnos en el mundo Backend que ofrece infinitud de opciones.

Debido a la limitación temporal y los problemas encontrados hay ciertos aspectos del desarrollo de la aplicación que se han quedado pendientes por tanto hay que incluirlos en cambios futuros. Además el proyecto puedes ser modificado y mejorado en diferentes aspectos.

Este proyecto final de carrera ha supuesto la adquisición de una gran cantidad de conocimientos. Por un lado tenemos los conocimientos profesionales, que seguro podrán ser aplicados en la vida laboral como profesional del sector, como son asentar algunos conocimientos aprendidos durante la carrera sobretodo relacionados con el desarrollo software. También se ha comprobado la complejidad que conlleva la realización de un proyecto, desde el análisis hasta la implementación y pruebas.

El tiempo que lleva el estudio de una nueva tecnología a utilizar y más si solo es desarrollado por una persona y no por un equipo.

Y por otro lado en lo personal, la posibilidad de hacer un proyecto relacionado con uno de tus hobbies, el mundo del cine, ha mejorado la experiencia.

El descubrimiento de nuevas páginas web relacionadas con el sector para obtener información tanto para uso y disfrute personal como para la utilización dentro de la aplicación.

En general ha sido una gran experiencia que ayuda a consolidar los conocimientos adquiridos a la vez que te ayuda a enfocar tu vida profesional.

7. BIBLIOGRAFÍA

2022. *Varios Materiales*. Grado en Ingeniería Informática. UOC.

2022. *Gestión de Proyectos*. Grado en Ingeniería Informática. UOC.

Openaccess.uoc.edu. 2022. *L'Oberta en Obert: Bachelor thesis, research projects, etc.*: <<http://openaccess.uoc.edu/webapps/o2/handle/10609/47986>> [Acceso 2 Junio 2022].

Sansagara, V., 2022. *Los pasos de la metodología de desarrollo 'Agile' y sus responsables - Snippets y Más. Snippets y Más.*: <<http://blog.leonelatencio.com/los-pasos-de-la-metodologia-de-desarrollo-agile-y-sus-responsables/>> [Acceso 9 Marzo 2022].

Kranio.io. 2022. *Blog: ¿Que es una API? Introducción practica con AWS API Gateway.*: <<https://www.kranio.io/blog/que-es-una-api-introduccion-practica-con-aws-api-gateway>> [Acceso 22 Febrero 2022].

[1] Apache.org. 2022. *Welcome to The Apache Software Foundation!.*: <<https://www.apache.org/>> [Acceso 5 June 2022].

[2] Apachefriends.org. 2022. *XAMPP Installers and Downloads for Apache Friends.*: <<https://www.apachefriends.org/>> [Accessed 9 June 2022].

[3] Hipertextual. 2022. *380 fondos de pantalla para cinéfilos.*: <<https://hipertextual.com/2017/03/380-fondos-pantalla-cinefilos>> [Acceso 14 Marzo 2022].

Es.wikipedia.org. 2022. *Wikipedia - Wikipedia, la enciclopedia libre.* [: <<https://es.wikipedia.org/wiki/Wikipedia>> [Acceso 1 June 2022].

2022:<<https://www.udemy.com/course/full-front-end-web-development-course/>> [Acceso 1 Mayo 2022].

Aprendejavascript.org. 2022. *Aprende JavaScript.*: <<https://aprendejavascript.org/>> [Acceso 5 Mayo 2022].

Anexo I - Entregables del proyecto

Carpeta llamada “Antelo_Pereira_Manuel_TFG” comprimida de manera adecuada en formato ZIP. La carpeta contiene por una parte la memoria del proyecto, y por otra parte la presentación, el video y el test de autoevaluación.

Anexo II – Acrónimos

- API: Application Programming Interface
- HTML: HyperText Markup Language
- CSS: Cascading Style Sheets
- SQL: Structured Query Language
- VPS: Private Server
- JS: JavaScript language source code file