

# FILMSS

## GUILLERMO LORCA MARTÍNEZ

DESARROLLO DE APLICACIONES WEB  
"2022/2023"

Alejandro Furriol y Enrique Galduf

Centro Integrado Público CIP FP **cecheste**  
FORMACIÓN PROFESIONAL

# ÍNDICE

RESUMEN.....	3
ABSTRACT .....	3
PLANIFICACIÓN DE LAS TAREAS Y TEMPORALIZACIÓN.....	4
UBICACIÓN DEL PROYECTO DENTRO DE LA EMPRESA.....	8
SECTOR PRODUCTIVO .....	8
ACTIVIDAD PRODUCTIVA DE LA EMPRESA .....	9
ANÁLISIS DE LA ESTRUCTURA ORGANIZATIVA DE LA EMPRESA. ....	10
CONVENIO APLICABLE A LA EMPRESA. ....	11
DETERMINACIÓN DEL TARGET. ....	12
DISEÑO.....	13
DISEÑO DE LA BASE DE DATOS. ....	13
DISEÑO DE LOS OBJETOS.....	17
<i>Diagrama de clases</i> .....	17
<i>Caso de uso</i> .....	19
MAPA WEB.....	20
DRAFTS/BORRADORES/MOCKUP DEL PROYECTO Y LAS INTERFACES .....	21
CODIFICACIÓN .....	26
TECNOLOGÍAS ELEGIDAS .....	26
ENTORNO SERVIDOR .....	28
ENTORNO CLIENTE .....	31
DOCUMENTACIÓN .....	32
MANUAL DE USUARIO .....	32
DOCUMENTACIÓN DEL CÓDIGO .....	32
DESPLIEGUE .....	34
DIAGRAMA O EXPLICACIÓN DEL DESPLIEGUE .....	34
DESCRIPCIÓN DE LA INSTALACIÓN O DESPLIEGUE.....	36
HERRAMIENTAS DE SOPORTE UTILIZADAS .....	38
BIBLIOGRAFÍA .....	39

## RESUMEN

El proyecto consiste en una aplicación web que brinda a los usuarios un buscador de películas, con el cual pueden descubrir información variada de su contenido favorito. Hay características adicionales, como la capacidad de crear una Watchlist, ver la de sus amigos o disfrutar de juegos temáticos.

El proyecto surge como respuesta a la demanda de los consumidores que buscan información sobre la disponibilidad de películas, series y otros contenidos audiovisuales. Además, muchas de estas personas desean contar con una plataforma intuitiva y funcional que les permita organizar el contenido audiovisual de su interés. El objetivo principal es crear una herramienta que facilite la exploración y el seguimiento de todo este contenido, brindando una experiencia atractiva y personalizada. La inclusión de la sección de juegos relacionados con este sector amplía aún más las posibilidades de entretenimiento, permitiendo a los usuarios poner a prueba su conocimiento y disfrutar de una experiencia completa.

Para el desarrollo del proyecto, se emplearon tecnologías como PHP Laravel en el entorno del servidor y Blade, Tailwind CSS, jQuery, JavaScript y Bootstrap para el desarrollo del entorno del cliente. Por otra parte, se sigue la metodología Kanban para la gestión de tareas, la cual ofrece un seguimiento del flujo de trabajo y proporciona una distribución equilibrada de la carga de trabajo. También se emplearon herramientas de control de versiones como Git y plataformas de alojamiento como GitHub.

Se ha logrado desarrollar una aplicación web completa y funcional que cumple con los objetivos establecidos. En ella, los usuarios pueden buscar y explorar películas, series, ... así como tener la opción de crear listas personalizadas y participar en la comunidad a través de comentarios y valoraciones.

En conclusión, este proyecto ha cumplido con las necesidades planteadas de proporcionar una plataforma para la búsqueda y gestión de contenido audiovisual. Esta web sienta las bases para futuras mejoras y expansiones, brindando una base sólida para seguir mejorando la experiencia del usuario y ofreciendo nuevas funcionalidades.

## ABSTRACT

Our web application makes a search engine available to users where they can explore and manage audiovisual content. The website offers different features such as Trending, the Watchlist and various Games.

This project has emerged in response to the growing demand from users, who seek easily accessible information on the availability of movies, TV series, and other audiovisual content. Nowadays users expect a user-friendly and feature-rich platform that not only allows them to access their favorite content but also offers organization and customization options. The primary objective of this project is to create a

dynamic and intuitive tool that simplifies the exploration and tracking of movies, TV series, and various other audiovisual offerings, providing users with an engaging and personalized experience. To achieve this, a range of innovative technologies and methodologies have been employed.

In terms of technology, server-side development takes advantage of PHP Laravel, a robust and efficient framework renowned for its versatility and scalability. Regarding the client side, the project incorporates a combination of Blade, Tailwind CSS, jQuery, JavaScript, and Bootstrap, allowing for visually appealing and responsive user interfaces. The Kanban methodology has been adopted. This methodology enables a streamlined workflow and provides clear visibility into project progress. By visualizing tasks on Kanban boards, we were able to prioritize and manage work efficiently, resulting in timely delivery of project milestones.

The outcome of this project is a fully functional web application that successfully meets the set objectives. Users can effortlessly search, explore, and discover a vast collection of movies, TV series, and additional audiovisual content. Furthermore, they have the ability to curate personalized lists, actively engage with the community through comments and ratings, and delve into the exciting world of gaming related to the entertainment industry.

To conclude, this project not only fulfills the users' demands for a comprehensive audiovisual platform but also paves the way for future enhancements, promising an even more enriching user experience and the introduction of new features.

## PLANIFICACIÓN DE LAS TAREAS Y TEMPORALIZACIÓN

El presente proyecto se comenzó a gestar a comienzos del 2023, en el marco del curso 2022/2023. Su iniciación se adelantó, en consonancia con la decisión adoptada tanto por el profesorado como por el alumnado, con el fin de garantizar el tiempo necesario para su desarrollo, lo que en años anteriores había ocasionado problemas. Se llevó a cabo una búsqueda temprana de ideas y el diseño del proyecto, lo que fue posible gracias a la decisión unánime del alumnado de Desarrollo de Aplicaciones Web de crear una web externa a las empresas de prácticas, permitiendo así realizarlo previamente al comienzo de esta época.

A lo largo de los meses de enero y febrero, se realizó una exhaustiva investigación de las tendencias actuales en internet, centrándose en el uso generalizado de plataformas de streaming como Netflix, HBO y otras similares. Con toda la información recopilada, se comenzó a hacer sesiones de lluvias de ideas con el objetivo de encontrar una página web atractiva para un amplio target, que lograra satisfacer plenamente sus necesidades. Durante estas tandas de generación de ideas surgieron varias propuestas prometedoras, aunque finalmente se acabó descartando algunas de ellas por preferencias personales.

Finalmente, se eligió una temática relacionada con el mundo del cine, en consonancia con la pasión personal del autor por este ámbito y, a su vez, se consideró que tenía un gran potencial para conectar con el público objetivo, teniendo en cuenta que todos somos consumidores habituales y se conocen sus necesidades.

Una vez se definió la temática de la web, se empezó a idear las diferentes funcionalidades y secciones que se incluirían en la misma. La principal era un buscador de contenido audiovisual, el cual proporciona distinto tipo de información como dónde poder verlo, pero, además de esto, se consideraron otras características como la Watchlist o los juegos. Posteriormente, se realizó una reunión con el profesorado, donde se realizaron aportaciones y se debatieron las distintas características, lo que permitió ajustar aún más las funcionalidades de la página y asegurar que estuvieran alineadas con los objetivos del proyecto.

Por lo que, los pasos realizados hasta ahora son los siguientes:

- *Investigación de tendencias de consumo*
- *Lluvias de ideas*
- *Elección de una temática y funcionalidades*
- *Reunión con el profesorado, donde se ajustaron las características*

Después de haber definido una idea muy clara de cómo se quería que fuese el proyecto y que funcionalidades tendría, era necesario fijar unos objetivos claros y seleccionar una metodología adecuada para poder llevarlos a cabo. En este caso, se optó por Kanban, una metodología ágil de desarrollo de software que se basa en la gestión visual a través del uso de tarjetas o notas que representan tareas. Estas se van moviendo a través de diferentes estados que hay en un tablero, lo cual permite visualizar el flujo de trabajo y el progreso. Gracias a Kanban se puede maximizar la eficiencia del tiempo y así garantizar el éxito del proyecto en el tiempo establecido.

#### **Metodología Kanban:**

- El primer paso dentro de la metodología Kanban implica la creación de un tablero en una herramienta de gestión de proyectos adecuada, para lo cual se ha seleccionado Trello. La elección de esta plataforma se debe a su alta eficiencia y gran flexibilidad, así como su capacidad para ofrecer una experiencia de uso altamente visual. Trello permite ver fácilmente el estado de cada tarea a través de su interfaz intuitiva y sencilla de usar.

Una vez seleccionada la herramienta de gestión de proyectos que se va a utilizar, se procede con la creación del tablero, en el que habrá tres columnas: "Por hacer", "En proceso" y "Hecho". A continuación, cada funcionalidad y requisito que se incluirá en la web se convertirá en tareas que

se añadirán a la sección "Por hacer". Asimismo, tras situar estas tareas en la columna, se deben dividir por prioridad. En este caso les se les asignara un color distinto en función de la importancia.

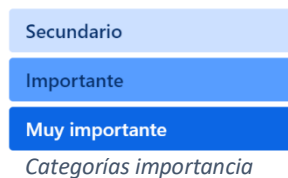
- En otro orden de cosas, es necesario hacer una estimación de los tiempos, es decir de debe poner una fecha límite para completar cada tarea. Esta debe ser una estimación realista y basada en la evaluación de cada trabajo.
- Por lo tanto, una vez completados los pasos anteriores, se podrá visualizar en Trello la siguiente tabla:

Por hacer	En proceso	Hecho
<div> <div></div> <div>Diseñar la web(Logo, Paleta, Tipo de letra)</div> <div>🕒 20 de feb.</div> </div> <div> <div></div> <div>Mapa web (FIGMA)</div> <div>🕒 26 de mar.</div> </div> <div> <div></div> <div>Diseño de la BD (Entidades, Tablas...)</div> <div>🕒 31 de mar.</div> </div> <div> <div></div> <div>Preparar el entorno de programación</div> <div>🕒 6 de abr.</div> </div> <div> <div></div> <div>Creación del frontend</div> <div>🕒 25 de abr.</div> </div> <div> <div></div> <div>Creación de la BD con Laravel (Migraciones, Seeders...)</div> <div>🕒 7 de may.</div> </div> <div> <div></div> <div>Creación de la lógica (Objetos, Clases)</div> </div> <div> <div></div> <div>Realizar el despliegue de la Web</div> </div> <div> <div></div> <div>Testeos de la web</div> </div>	<div>+ Añada una tarjeta</div>	<div>+ Añada una tarjeta</div>

Tabla Trello – Tareas por hacer

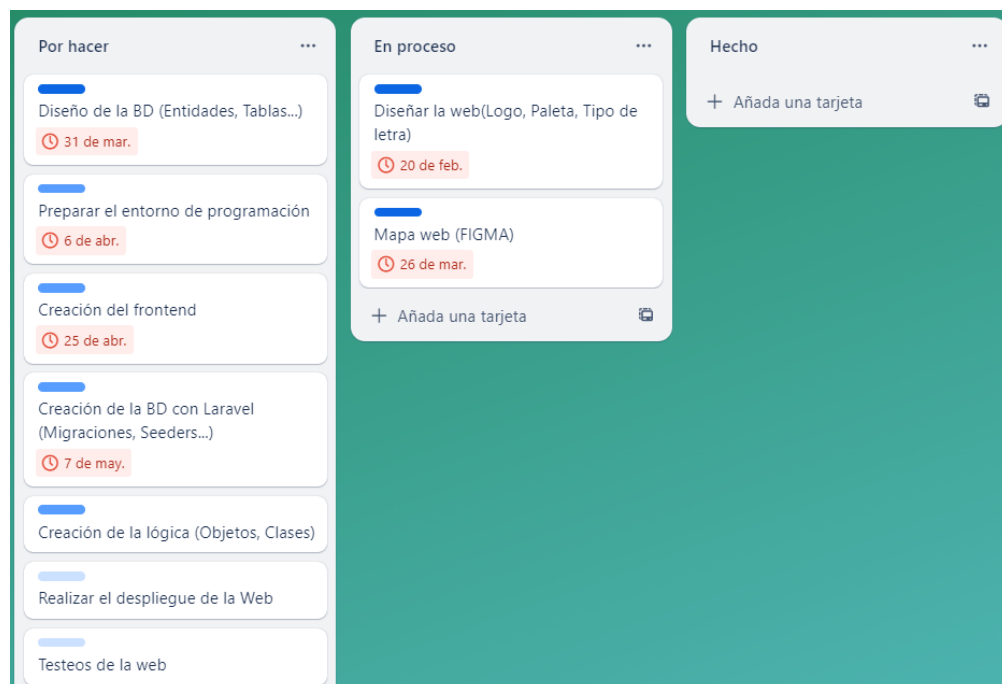
En la siguiente tabla se pueden observar diversas tareas en la columna "Por hacer", ya que esta es la etapa inicial y aún no se ha comenzado ninguna función. Además, las tareas están categorizadas en:

- "Secundario"
- "Importante"
- "Muy importante "



Y, cada grupo está relacionado con un color concreto. Cabe destacar que estas tarjetas tienen una fecha límite que también sirve para determinar su relevancia, dado que aquellas que hay que completar antes son necesarias para poder realizar los otros apartados.

- Para continuar, es necesario comenzar las tareas y asignar una o dos en función de su volumen de trabajo. Una vez asignada, se debe mover a la columna "En proceso", pues no es eficiente trabajar en todas las tareas al mismo tiempo, por lo que, es mejor hacerlas de una en una o de a dos. Para empezar, se ha centrado la atención en el "Diseño de la Web" y en el "Mapa Web", tal y como se puede observar en la siguiente tabla:



*Tabla Trello – Tareas en proceso*

- El siguiente paso es la finalización de las tareas, lo cual implica mover una tarea que está en proceso a la columna "Hecho" una vez que se haya completado. Esta fase es crucial, ya que permite mantener una visión clara del proyecto y garantiza que se están cumpliendo los objetivos establecidos. Como se puede observar a continuación:

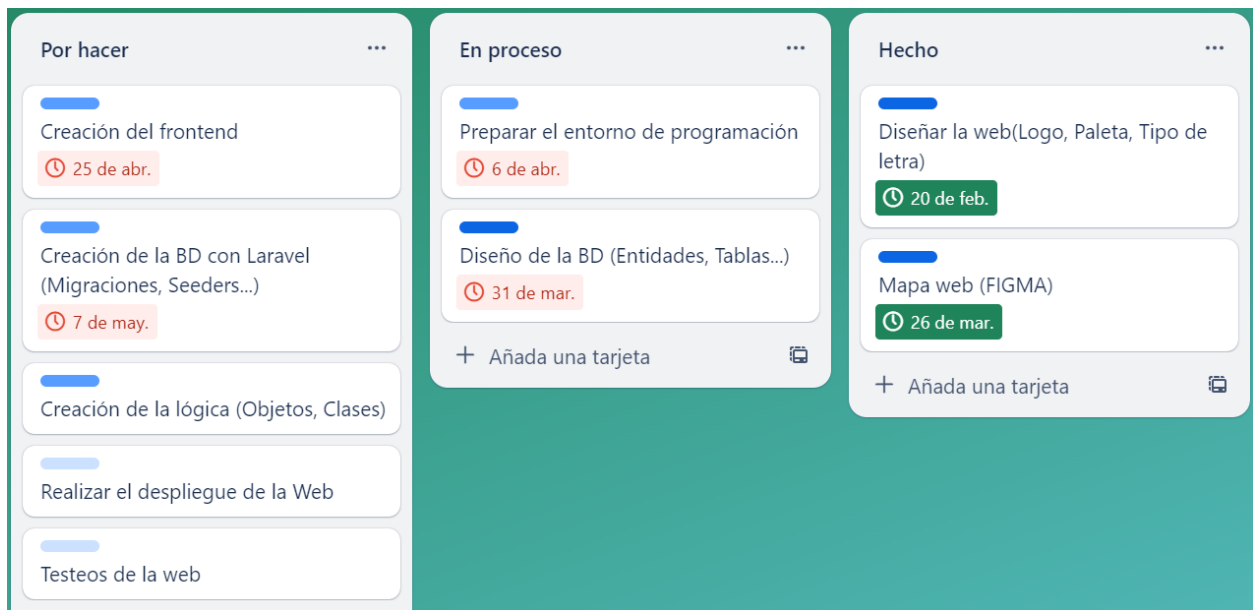


Tabla Trello – Tareas finalizadas

En conclusión, se ha utilizado el sistema Kanban y la herramienta Trello para organizar el proyecto y monitorearlo. Gracias a esto, ha sido posible abordar todas las tareas de manera ordenada y eficiente, y asegurarse de no omitir ninguna de ellas.

## UBICACIÓN DEL PROYECTO DENTRO DE LA EMPRESA

### Sector productivo

El proyecto que se va a desarrollar se trata de una aplicación web cuya principal característica consistirá en un buscador de contenido audiovisual. Además, la aplicación contará con otras funcionalidades, tales como nuestra Watchlist en la que el usuario podrá guardar el contenido que desee ver en un futuro o que le haya gustado. Así como, juegos relacionados con el contenido ofrecido para que los usuarios puedan disfrutar de una experiencia completa en la plataforma.

Por todas estas características, el proyecto se enmarca en el sector del entretenimiento. Concretamente, dentro de los subsectores del cine, series y videojuegos. Este se caracteriza por su dinamismo y constante cambio, ya que los avances tecnológicos y la evolución de las tendencias de consumo transforman continuamente el contenido audiovisual que se produce y la manera de crearlo.

A continuación, se presentan otras de las principales características del sector de entretenimiento. En primer lugar, este sector cuenta con una audiencia amplia, dado que se considera una forma de escapar de la rutina y relajarse, lo que la convierte en una actividad atractiva para todo tipo de personas. Los subsectores como el cine, las series y los videojuegos tienen un gran atractivo para una audiencia variada, desde jóvenes hasta adultos. Asimismo, el sector ha impulsado el surgimiento de distintos tipos de



plataformas para disfrutar del entretenimiento, que van desde plataformas de transmisión en vivo como Twitch, hasta plataformas que ofrecen contenido audiovisual diverso como Netflix o HBO, entre otras.

El sector del entretenimiento tiene un valor inmenso tanto a nivel global como a nivel nacional. Se trata de una industria que está en constante crecimiento y que, en los últimos años, ha experimentado un auge sin precedentes. Concretamente tiene un valor aproximado de 2,9 billones de dólares a nivel global. Según el Anuario de Estadísticas Culturales 2022 del Ministerio de Cultura y Deporte, el valor del sector cultural en España en 2020 alcanzó los 34.106 millones de euros, lo que representa el 3,2% del PIB del país.

En cuanto a la prospección de crecimiento del sector del entretenimiento en la economía española, se espera que siga siendo una industria importante en los próximos años, pues la digitalización y la globalización han abierto nuevas oportunidades para el sector del entretenimiento. Se prevé que la demanda de contenido y experiencias de ocio siga creciendo en el futuro. Además, la pandemia ha acelerado la transición hacia el consumo de contenidos digitales, lo que podría impulsar aún más el crecimiento del sector en los próximos años.

### Actividad productiva de la empresa

Este proyecto se trata de una plataforma web que cubre varias necesidades que tienen los usuarios en relación con el entretenimiento audiovisual. Principalmente, cubre la necesidad de la búsqueda de películas, series, personas...; ya que nuestra web permite a los usuarios encontrar fácilmente contenido de su interés a través de nuestro buscador.

Asimismo, otra necesidad que abordamos es la de tener un lugar donde los usuarios puedan gestionar el contenido que desean ver, que les han recomendado o que están viendo. Por ello, ofrecemos una Watchlist, en la que los usuarios pueden crear una lista personalizada de contenido audiovisual. De esta manera, pueden planificar y ordenar su tiempo de ocio, así como llevar un registro detallado del progreso en su contenido audiovisual.

La web también incluye un apartado que ofrece juegos que permiten al usuario poner a prueba sus conocimientos sobre el mundo del cine. Los juegos ofertados son un “Ahorcado”, otro que consiste en adivinar el contenido a través del poster, y el último, a partir de emojis. Además, esta sección de la aplicación añade un elemento lúdico y divertido a la experiencia de uso de nuestra web.

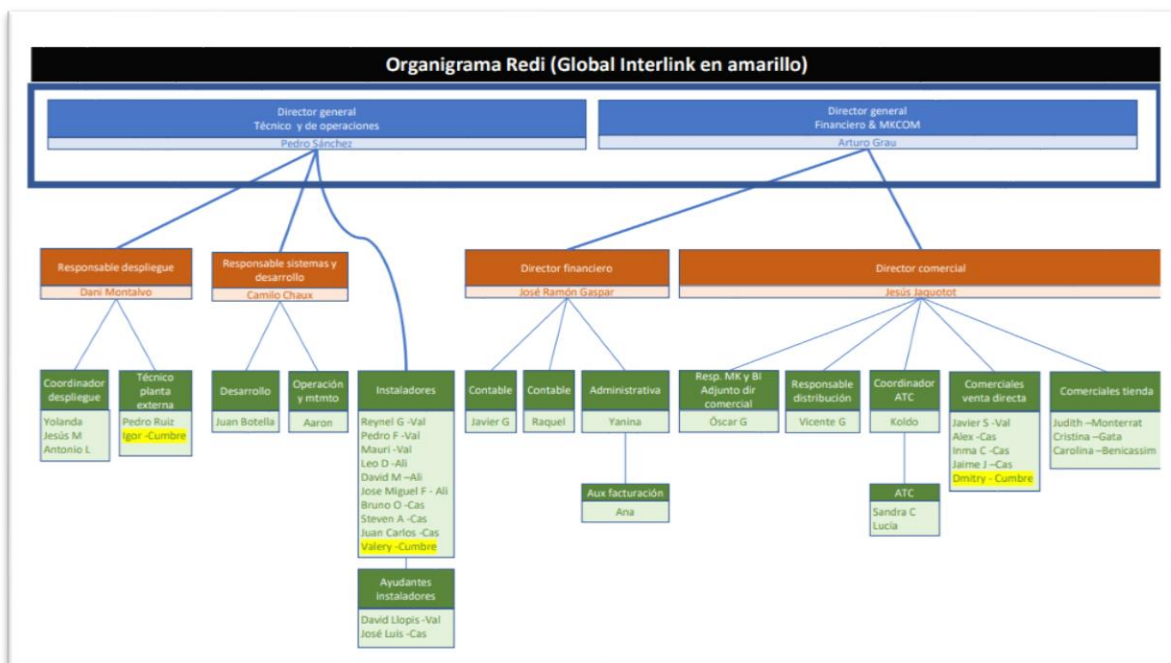
El proyecto va dirigido a usuarios que les interese el mundo del cine y que quieran hacer seguimiento de él, así como disfrutar de juegos relacionados con el mismo. Se desarrolla en 2023, un período en el cual el sector está en auge, por lo que, los usuarios con estas necesidades aumentan.

Respecto a las ventajas que aporta el proyecto frente a productos o servicios similares, se ha realizado un análisis previo de la competencia en el mercado. Y se ha identificado que los productos similares

existentes en el mercado no ofrecen una experiencia completa para los usuarios, ya que carecen de algunas funcionalidades importantes. Además, los productos existentes suelen ser complicados de usar y poco intuitivos para el usuario promedio.

Por lo tanto, nos diferenciamos de los competidores que ofrecen un servicio similar aportando distintas ventajas frente a ellos. La principal diferencia es la combinación de funciones, dado que ofrecemos varias actividades en una sola web. Esto hace que los usuarios ahorren tiempo y tengan todo lo que necesitan o les interesa en un mismo sitio. Asimismo, esta web tiene un uso sencillo e innovador que hace que nuestros clientes puedan utilizarla más fácilmente.

Igualmente, otra de nuestras ventajas que nos diferencia de nuestros competidores, es nuestra Watchlist, pues ofrecemos un servicio personalizado, que es lo que más demandado está últimamente en nuestro



sector. Esto, le da un valor añadido a nuestra web, pues la mayoría de nuestros competidores solo ofrecen el buscador de contenido audiovisual.

También nos caracteriza la accesibilidad, porque al ser una plataforma web los usuarios pueden acceder desde cualquier dispositivo conectado a internet. Ello permite a los usuarios hacer una búsqueda, añadir una película a su Watchlist o jugar su partida diaria desde cualquiera de sus dispositivos y desde cualquier lugar con conexión.

## Análisis de la estructura organizativa de la empresa.

En el siguiente organigrama de Redi, se detallan los distintos departamentos y sus relaciones entre sí, así como de los trabajadores de cada uno y el cargo que tiene cada persona. Este organigrama es de gran

*Organigrama empresa Redi*

relevancia porque detalla muy bien, con un solo vistazo, la estructura de la empresa, así como su tamaño y organización. Se trata de un organigrama vertical e informativo, ya que su función es evidenciar el nivel de jerarquía vertical y se usa como carta de presentación.

En el presente organigrama se puede observar que hay dos directores generales que lideran la empresa dividiéndola en dos partes: la técnica y la financiera. Ambos directores son responsables de establecer la hoja de ruta que va a seguir cada sector de la empresa y de tomar las decisiones importantes en sus respectivas áreas.

En un nivel más abajo se encuentran: por un lado, el responsable de despliegue y el de sistemas y desarrollo, por otro lado, el director financiero y el comercial. Por último, se encuentran los empleados de cada departamento, quienes tienen diferentes funciones en la empresa.

En esta empresa, hay algunos departamentos que se organizan y se comunican a la hora de realizar las diferentes tareas. En el caso de la parte técnica, para llevar a cabo tareas como el despliegue e instalación de una red, se involucra el departamento de despliegue colaborando con los diferentes instaladores que manda el coordinador. Además, estos están en constante comunicación con Camilo Chaux que monitoriza y estudia si la red está correctamente desplegada y en funcionamiento.

Por otra parte, aunque esta empresa no se centra en el sector de mercado del entretenimiento, se podría estudiar la incorporación en el ciclo de desarrollo de software de la Redi, en el cual hay varias fases. En primer lugar, diseño y planificación, en la que trabajaríamos tanto yo como el equipo de desarrollo Juan Botella y Camilo Chaux, el responsable. Los tres nos pondríamos de acuerdo y acordaríamos la línea de estilo que vamos a seguir en el proyecto, para después repartir las tareas. En segundo lugar, este mismo equipo se encargaría de la fase de desarrollo, donde se crearía el software y se le haría distintos tests. Por último, la fase de implementación, en la que también participarían los desarrolladores, pero donde se involucraría el departamento de despliegue. Este departamento, liderado por Dani Montalvo, llevaría a cabo la puesta en marcha del software en un ambiente de producción.

### Convenio aplicable a la empresa.

El proyecto se enfoca en la creación de una aplicación web que proporciona varios servicios, entre ellos la búsqueda de contenido audiovisual, una Watchlist o diferentes juegos relacionados con el cine. Por ello, el convenio colectivo que resulta más adecuado es el XVII Convenio colectivo estatal de empresas de consultoría, y estudios de mercados y de la opinión pública. La resolución de la aprobación de este convenio fue el 22 de febrero de 2018 y fue aprobado por el Ministerio de Empleo y Seguridad Social.

Este convenio colectivo regula las relaciones laborales entre las empresas y los trabajadores del sector. Y, en él, se abordan cuestiones como la jornada laboral, los salarios, la formación, la movilidad funcional, el tiempo libre, la seguridad e higiene en el trabajo, entre otras.

A pesar de que el proyecto no se enfoca directamente en una empresa de consultoría y estudios de mercado, para la realización de este proyecto se necesita un perfil profesional muy concreto: un técnico superior en desarrollo de aplicaciones web, ya que es quien se encargará de diseñar, desarrollar y mantener la aplicación web. Dicha titulación capacita a la persona para optar a puestos de Programador senior o junior, dependiendo de su experiencia laboral. En este caso, el perfil requerido sería de Programador junior, lo cual lo ubica en el grupo E-1 del convenio colectivo.

Asimismo, es importante destacar que este perfil profesional resulta esencial para garantizar el correcto funcionamiento, la seguridad y la accesibilidad de la web en cuestión. Por lo tanto, resulta indispensable contar con un técnico superior en desarrollo de aplicaciones web para llevar a cabo con éxito este proyecto.

Para concluir, el convenio colectivo mencionado anteriormente es el más adecuado, ya que es el que mejor se ajusta a las necesidades y características de la actividad profesional del proyecto. Pues permite establecer las condiciones laborales más idóneas para un técnico superior en desarrollo de aplicaciones web.

### Determinación del target.

El proyecto está dirigido a un segmento de mercado tanto joven como adulto, con un fuerte interés por el cine, las series y demás contenido audiovisual. El target se caracterizaría por una edad media que oscila entre los 18 y los 35 años, aunque puede haber variaciones. Este grupo de consumidores es tecnológicamente avanzado y hace un uso bastante asiduo de herramientas digitales y dispositivos móviles.

Nuestro público objetivo busca una plataforma que les brinde una experiencia completa y satisfactoria en cuanto a información y entretenimiento relacionado con el mundo del cine. Por esta razón, el proyecto se centra en cubrir esta necesidad con una plataforma con un acceso rápido, fácil e intuitivo a información actualizada y de interés.

Además de buscar la información, los usuarios también buscan un servicio personalizado en el que puedan plasmar sus gustos e intereses. Por este motivo, también les ofrecemos una Watchlist, una plataforma que les da la posibilidad de crear listas personalizadas de contenido audiovisual. También podrán compartirlas con otros usuarios, lo que les permitiría interactuar y descubrir nuevas películas, series...

En cuanto a los hábitos de consumo, este segmento de mercado destaca por su digitalización y por el uso de la mayoría de las plataformas digitales. Muchos de ellos prefieren realizar muchas de sus acciones cotidianas a través de sus dispositivos digitales, desde acciones tan comunes como hacer la compra hasta las más lúdicas como ver películas o series en plataformas de streaming.

## DISEÑO

### Diseño de la Base de datos.

En esta sección, se llevará a cabo el diseño de la Base de Datos del proyecto. La BD tiene una peculiaridad, pues se ha decidido emplear una API para extraer datos sobre películas, series y otro contenido multimedia, los cuales se utilizarán para el buscador. Cabe destacar que la base de datos no almacena los datos de los contenidos multimedia, sino que se guardaran comentarios, valoraciones y Watchlist en los que se hará referencia a los nombres de, por ejemplo, una película.

En la base de datos, existen diversas entidades con sus respectivos atributos y cada entidad representa una tabla de la BD y, sus atributos, las columnas de dichas tablas. Estas entidades están relacionadas, por lo que las tablas también lo están. A continuación, se presenta un esquema Entidad / Relación:

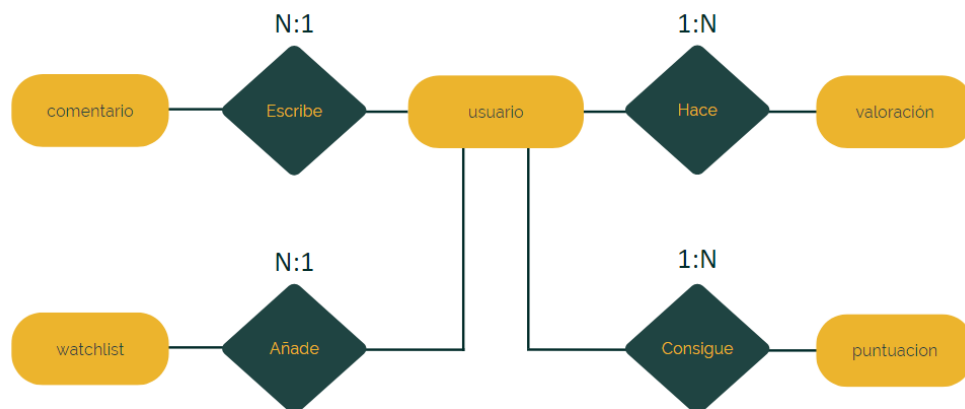


Diagrama ER

Un diagrama entidad-relación es una herramienta fundamental en la gestión de las BD, ya que permite ver visualmente la estructura y las relaciones de los datos. En el caso de este proyecto, se han identificado varias entidades que son relevantes para la gestión de la página web, como son los usuarios, las watchlists, los comentarios, las puntuaciones y las valoraciones.

La entidad principal es *usuario*, dado que son ellos quienes realizan todas las acciones dentro de la web y, cada una de esas acciones, se plasma en las otras entidades. Una de estas acciones sería crear una

*watchlist* personalizada. Que, como se puede observar en el diagrama, un usuario puede añadir distinto contenido a su lista (N) y ese contenido es añadido por un único usuario (1). Otra de las entidades es el *comentario*, en el que un usuario puede escribir lo que opina sobre el contenido que busca, así como de sus personas de interés. Además de poder dejar estos comentarios también se pueden hacer *valoraciones* de él, las cuales irán comprendidas del 0 al 10. Por otra parte, el usuario también se puede jugar a juegos y, una vez acabada una partida, conseguir una puntuación en función de los intentos y de otros parámetros. Además de todas estas funcionalidades, los usuarios podrán ser amigos, es decir podrán enviarse peticiones de amistad y ver las Watchlist de sus amigos.

Una vez expuestas las entidades y sus relaciones, se procederá a representarlas mediante un diagrama UML, el cual nos permitirá visualizar las tablas, sus columnas y las claves que las definen. De esta manera, se podrá comprender con mayor claridad la estructura de la base de datos y la forma en que se relacionan las distintas entidades.

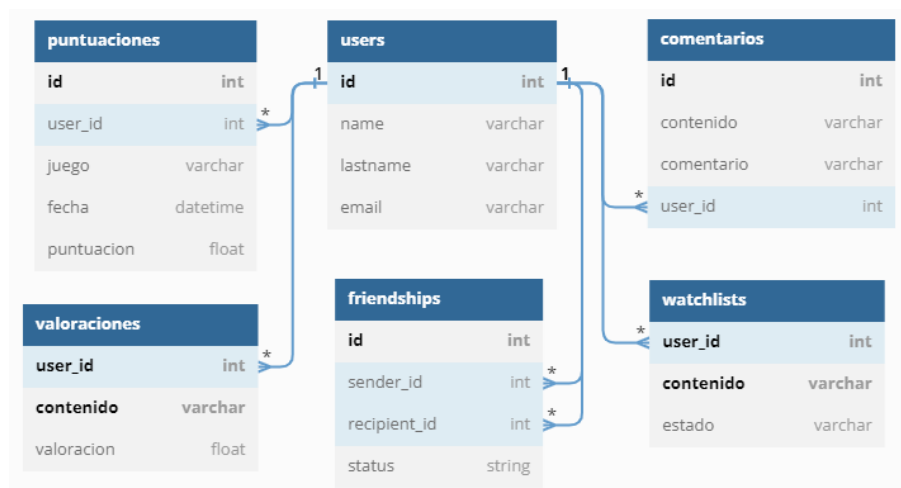


Diagrama UML

- **Users:** Esta tabla representa a los usuarios de la página web y las distintas funcionalidades a las que podrán acceder. Estas, se encuentran representadas por las tablas que apuntan al ID del usuario, tales como: “*puntuaciones*”, “*comentarios*”, “*valoraciones*”, “*watchlist*”.

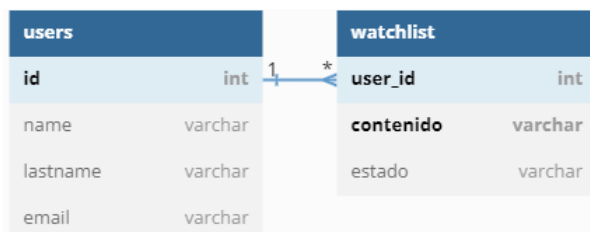
De cada usuario, se almacena sus datos personales, específicamente su “*nombre*”, “*lastname*” y “*email*”. Asimismo, cada uno tendrá asignado un ID único que se generará automáticamente y se incrementará en orden secuencial.

users	
id	int
name	varchar
lastname	varchar
email	varchar

Tabla users

- **WatchList:** Esta tabla representa una lista personalizada que tendrá cada usuario de la web. En esta lista, los usuarios serán capaces de guardar películas, series y otro tipo de contenido. Se podrá guardar contenido que se haya visto, que haya gustado y contenido que se quiere ver en un futuro y se haya descubierto en la web.

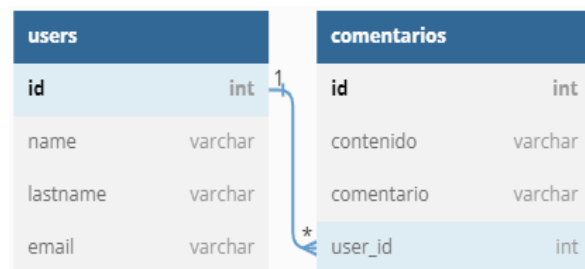
La tabla cuenta con una clave primaria compuesta por dos columnas: el id del usuario "*user\_id*" y el nombre del contenido. De esta manera, se evita que un mismo usuario tenga contenido repetido en su Watchlist. Además, la tabla incluirá un campo llamado "estado" para que el usuario pueda indicar qué quiere hacer con cada contenido de su lista.



*Relación entre las tablas Watchlist y Users*

Se puede observar, que la tabla tiene una clave ajena en la que se relaciona el "*user\_id*" de la misma con el id de la tabla usuarios. Se trata de una relación uno a muchos, ya que un usuario puede tener muchos contenidos en su lista, pero una Watchlist solo puede pertenecer a un usuario.

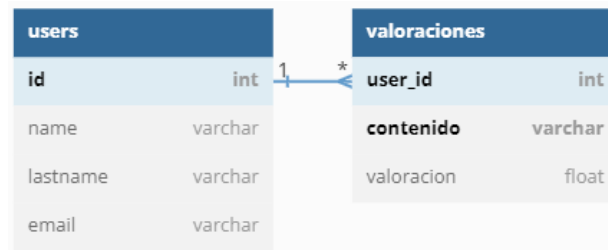
- **Comentarios:** Esta tabla permitirá a los usuarios de la web publicar comentarios sobre todo tipo de contenido multimedia. Por ejemplo, si les ha gustado una película, se podrán dejar un comentario al respecto, o si desean elogiar a una persona por su actuación. La tabla "*comentarios*" almacenara el nombre del contenido sobre el que se ha hecho el comentario, el usuario que lo ha publicado y, por supuesto, el comentario en sí.



*Relación entre las tablas Comentarios y Users*

Como se puede observar, la tabla tiene una clave primaria compuesta por un id generado automáticamente que identifica cada comentario. Igualmente, esta tabla se relaciona con la tabla "Usuarios" a través de una clave ajena, lo que permite identificar al usuario que ha publicado el comentario.

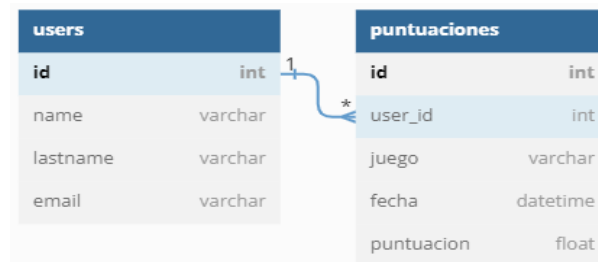
- **Valoraciones:** Esta tabla tiene como objetivo guardar las valoraciones de los contenidos multimedia que se obtengan de la API. Lo que se pretende con esto es conocer las preferencias de los usuarios y así, crear una comunidad en la que solo los usuarios registrados en la web puedan valorar el contenido. Para ello, se ha diseñado una tabla en la que se guardarán los datos del contenido que se valora, la valoración en sí misma y el usuario que ha realizado dicha valoración.



*Relación entre las tablas Valoraciones y Users*

Tal como se puede analizar, el “*user\_id*” de esta tabla está relacionado mediante una clave ajena con la tabla “*usuarios*”. Asimismo, la clave primaria está compuesta por el id del usuario y el contenido, asegurando así que un mismo usuario solo pueda valorar un contenido una vez.

- **Puntuaciones:** Esta entidad se refiere al apartado de juegos en la web, donde se ofrecen distintos juegos relacionados con películas, series, etc. Esta tabla almacena las puntuaciones de los usuarios en cada juego, junto con el nombre de dicho juego y la fecha en la que jugó esa partida. Como todas las tablas anteriores, esta también se relaciona mediante una clave ajena con la tabla “*Users*”, ya que se necesita saber que usuario ha realizado cada puntuación.

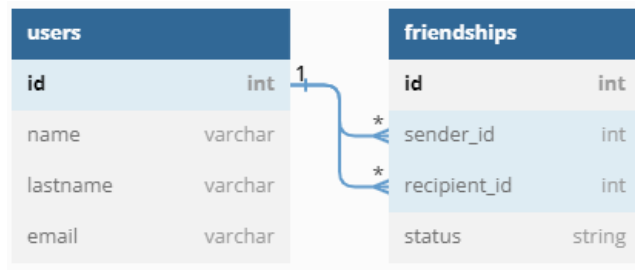


*Relación entre las tablas Puntuaciones y Users*

- **Friendships:** esta tabla almacena las relaciones entre los usuarios, contiene las siguientes columnas:
  - **id:** Identificador único de la relación de amistad.
  - **sender\_id:** ID del usuario que envía la solicitud de amistad.
  - **recipient\_id:** ID del usuario que recibe la solicitud de amistad.



- **status**: Estado de la amistad, que puede ser "pending" (pendiente), "accepted" (aceptada) o "denied" (denegada).



Tablas User y Friendships

## Diseño de los objetos.

### Diagrama de clases

En este apartado, se va a elaborar un diagrama de clases, el cual consiste en una representación visual que muestra las clases del sistema, sus atributos, métodos y las relaciones entre ellas. Esta herramienta desempeña un papel esencial en el diseño orientado a objetos y en el proceso de desarrollo de software. Además, facilita la creación de un diseño efectivo, la documentación del sistema y otros aspectos, lo que contribuye a la creación de un software bien estructurado, flexible y de calidad.

El diagrama de este proyecto en cuestión es el siguiente:



Diagrama de clases

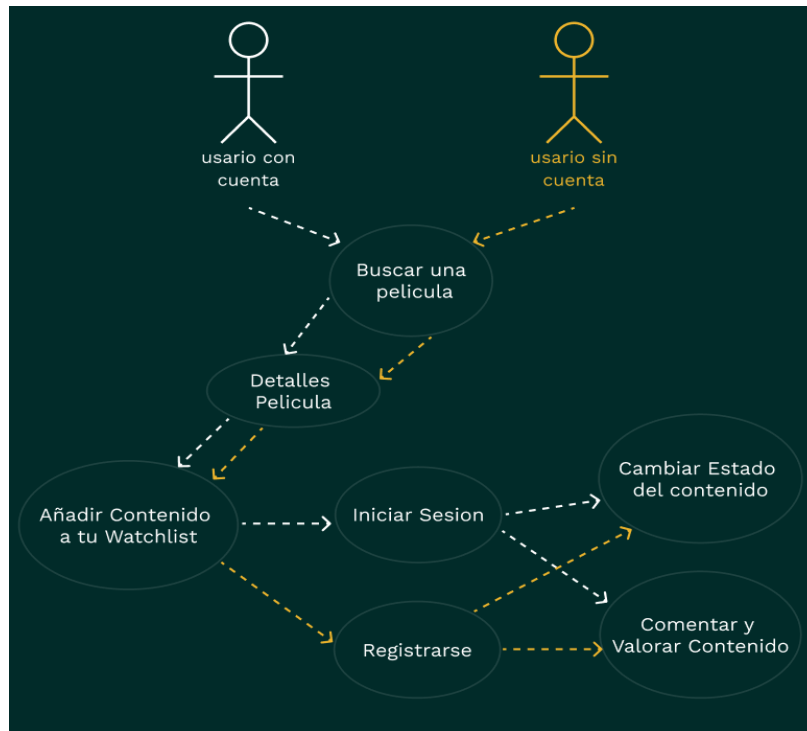
En el diagrama de clases anterior, se pueden observar las siguientes clases principales:

- **Usuario:** Representa a un usuario del sistema, con atributos como id, nombre, correo y contraseña. También tiene métodos para registrarse, iniciar y cerrar sesión. User también tiene otros métodos que nos proporciona la biblioteca laravel-friendships, con esta hacemos al user Friendable y conseguimos varios métodos:
  - befriend
  - acceptFriendRequest
  - denyFriendRequest
- **Watchlist:** Clase que gestiona la lista de seguimiento correspondiente a cada usuario, con atributos como user\_id, y contenido y estado. Tiene métodos para agregar y eliminar contenido de la lista, así como para cambiarles el estado.
- **Comentario:** Clase que maneja los comentarios realizados por un usuario, con atributos como id, user\_id, contenido y comentario. Tiene métodos para añadir y eliminar comentarios.
- **Puntuación:** Representa una puntuación conseguida por un usuario, con atributos como id, user\_id, juego y puntuación. Tiene un método en el que se agrega una puntuación.
- **Valoración:** Esta clase se encarga de las valoraciones realizadas por un usuario, con atributos como user\_id, contenido y valoración. Tiene métodos para agregar y eliminar valoraciones.

En este diagrama de clases, se establecen las relaciones entre las clases mediante asociaciones. Por ejemplo, se puede observar una relación de uno a muchos entre la clase Usuario y la clase Watchlist, lo que significa que un usuario puede tener una única Watchlist, pero una Watchlist puede contener múltiples películas, series o personas.

## Caso de uso

Un caso de uso es una técnica utilizada en la ingeniería de software para describir la interacción entre los usuarios y el sistema en desarrollo. Proporciona un ejemplo de las acciones y comportamientos que ocurren en un escenario específico.



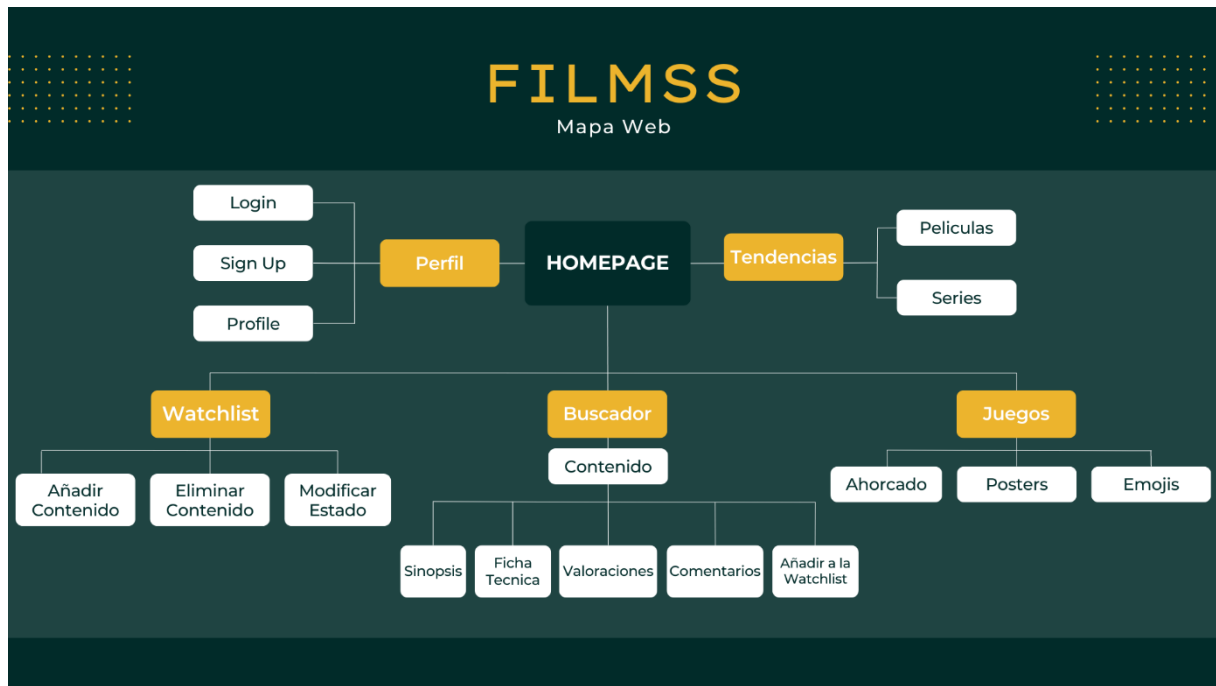
*Caso de uso FILMSS*

En este caso de uso se puede observar cómo los dos usuarios utilizan el buscador de la web y, después de encontrar lo que deseaban, específicamente una película, ambos intentan añadirla a su Watchlist. El usuario que ya tiene una cuenta debe iniciar sesión, mientras que el otro debe registrarse. Una vez añadido el contenido a su respectiva Watchlist, tienen la capacidad de modificar el estado de la película y realizar otras acciones, como comentarla o valorarla.

Los casos de uso son una herramienta esencial en el desarrollo de software, dado que ayudan a capturar una descripción detallada de las interacciones entre los usuarios y la aplicación. El caso de uso presentado anteriormente ha proporcionado una comunicación clara y efectiva entre los diferentes roles del proyecto, centrando el enfoque en el usuario. Esto contribuye a la comprensión y al éxito en el desarrollo del sistema.

## Mapa web

El mapa web es una representación visual de la estructura y organización de una página web, que permite a los usuarios y desarrolladores comprender de manera clara y precisa la distribución de las secciones y subsecciones del sitio. A continuación, se presenta el mapa web para el proyecto en cuestión:



Mapa Web

Como se puede observar, la página principal del sitio web, conocida como *"Homepage"*, es la primera imagen que reciben los usuarios al entrar. Esta sección contiene el menú de navegación que permite acceder a otras secciones, así como un buscador para explorar diferentes tipos de contenido. En esta página, se ofrece una selección de películas y series populares. Los usuarios tienen la opción de aplicar diversos filtros para encontrar el contenido que deseen.

El buscador es una herramienta fundamental en el sitio web, pues los usuarios pueden realizar búsquedas de todo tipo de contenido: personas, películas y series. Al realizar una búsqueda, se muestra una página con todas las coincidencias encontradas, las cuales pueden ser filtradas según su tipo.

Al hacer clic en un contenido específico, los usuarios son dirigidos a una página dedicada a ese contenido en particular. En esta página, se proporciona información detallada como la sinopsis, ficha técnica, valoraciones, comentarios, entre otros datos relevantes. Además, los usuarios tienen la opción de agregar el contenido a su lista de "Watchlist" u otras acciones disponibles.

Como se ha comentado anteriormente, en la página principal hay una barra de navegación que proporciona acceso a varios apartados: Perfil, Tendencias, Watchlist, Juegos. A continuación, se va a describir cada una de estas secciones:

- **Perfil:** A esta sección solo tienen acceso los usuarios registrados y que han iniciado sesión. Si ya se tiene una cuenta en FILMSS, se puede acceder al perfil iniciando sesión en la opción "*login*". Si aún no se tiene una cuenta, se puede crear una nueva cuenta en la opción "*sign\_up*". Una vez que se haya iniciado sesión con la cuenta, se podrá acceder a la sección privada en la que se encontrará los datos personales, como nombre, dirección de correo electrónico y contraseña. Igualmente, en esta sección se pueden ver las plataformas seleccionadas como predeterminadas para las búsquedas.
- **Tendencias:** En esta sección, se ofrece contenido popular y de tendencia. Aquí se podrá encontrar una selección de películas y series que son populares en el momento. Esta sección se actualiza regularmente para mostrar las últimas tendencias en el mundo del entretenimiento.
- **Watchlist:** La sección de Watchlist permite crear una lista personalizada de contenido que interesa a los usuarios. Se puede agregar películas, series o cualquier otro contenido a la Watchlist para tener un seguimiento de lo que se desea ver en el futuro o lo que se está viendo actualmente. Esta sección también permite administrar y organizar el contenido guardado, pudiendo cambiar el estado del contenido.
- **Juegos:** Esta sección ofrece diversos juegos relacionados con las películas y series, concretamente: un Ahorcado, un Posterdle y Emojis. Con esta sección los usuarios pueden disfrutar de momentos divertidos mientras ponen a prueba sus conocimientos sobre películas y series.

El mapa web ha sido diseñado cuidadosamente para ofrecer una navegación intuitiva y eficiente a los usuarios, permitiéndoles acceder fácilmente a la información y las funcionalidades que buscan. Asimismo, busca brindar una experiencia atractiva y completa con el fin de satisfacer las necesidades y expectativas de los usuarios.

### Drafts/borradores/mockup del proyecto y las interfaces

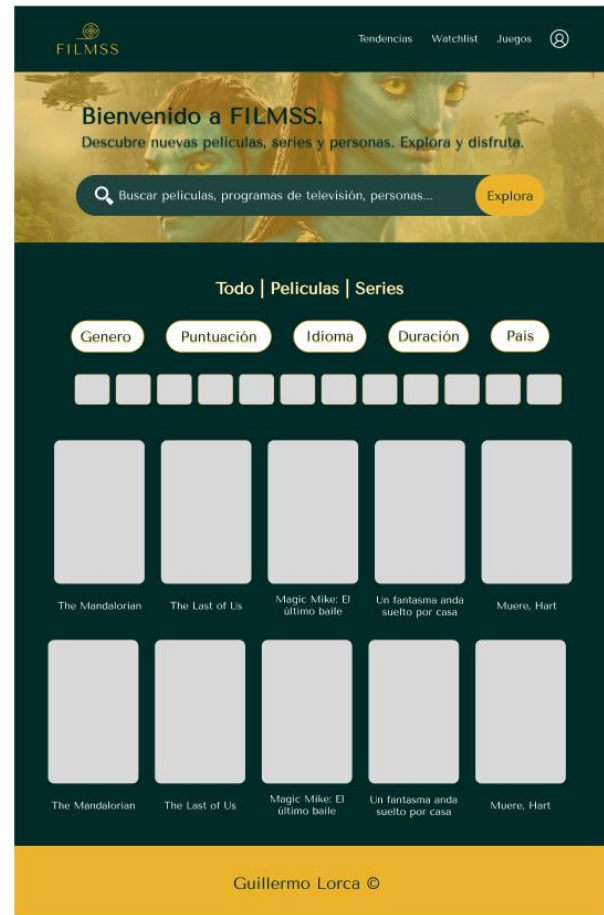
En el proceso de desarrollo de una página web, una de las etapas iniciales es la de diseño. Esta desempeña un papel fundamental para definir la apariencia y la disposición de las interfaces con las que el usuario interactuará.

En esta fase, es habitual realizar un borrador con alguna herramienta de diseño y prototipado, en este caso se ha decidido usar Figma. Esta es una aplicación gratuita basada en la nube que nos permite crear mockups interactivos en tiempo real. Además, ofrece una interfaz intuitiva y amigable, lo que facilita su aprendizaje y uso.

Un mockup es una representación fiel del diseño final, incluyendo la disposición de los elementos, el tamaño y el estilo de las tipografías, los colores y otros aspectos visuales. Permite visualizar cómo se verá y funcionará una página web antes de su implementación real. A continuación, se muestra el mockup del proyecto, este se divide en diferentes interfaces:

#### Página de inicio:

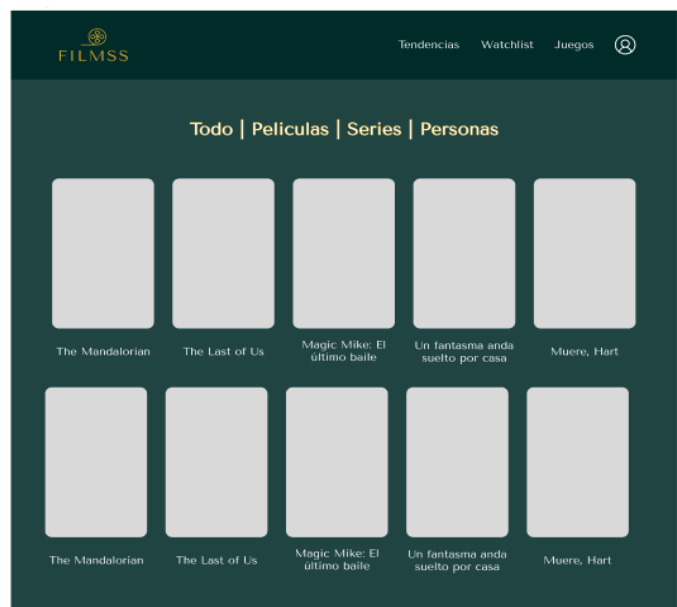
- **Header:** en él, se encuentra el logo de la web y la barra de navegación con las diferentes secciones.
- **Buscador:** buscador, con un fondo que va cambiando.
- **Contenido:** en esta sección se muestran pelis y series, las cuales se pueden filtrar con las distintas opciones.
- **Footer:** es la última sección de la web y es donde irían las redes sociales o el copyright.



Página de Inicio

#### Resultados de búsqueda:

- **Filtros:** cuando se busca algo se puede separar por personas, películas o series.
- **Resultados:** una vista previa de los resultados en la que se muestra el poster y el título de cada uno.



Resultados de búsqueda

## Detalle del contenido:

- **Imagen del contenido:** la aporta la API .
- **Portada y acciones:** foto de la portada y las diferentes acciones.
- **Sinopsis**
- **Datos:** páginas donde se puede ver, tipo, duración y más datos.
- **Contenido relacionado**
  - **Película**
  - **Serie**

**Película**  
**Criminales a la vista**

**Ver Ahora**  
Netflix HBO

**Tipo**  
Película

**Calificación**  
7.5

**Duración**  
1h 28min

**Edad**  
+12

**Director**  
Jeremy Canelik

**Año**  
2023

**Géneros**  
Comedia, Romance, Acción & Aventura, Crimen

**Cast**  
Adam Sandler Jennifer Aniston Mark Strong  
Nick Audrey Miller

**Trailer**  
[Murder Mystery 2 | Official Trailer | Netflix](#)

**Contenido Relacionado**

Audrey y Nick deben descubrir quién secuestró a su amigo antes de que sea demasiado tarde.

Nick y Audrey están de vuelta. Ahora, como verdaderos investigadores privados, se quedan boquiabiertos cuando secuestran a su amigo el Maharaja en su propia boda. Con un rastreo de pistas, la pareja de detectives debe unirse para resolver este misterio internacional y rescatar a su amigo indefenso.

Detalle de Película

**Serie**  
**The Mandalorian**

**Ver Ahora**  
Disney +

**Tipo**  
Serie

**Calificación**  
8.1

**Duración**  
3 Temporadas

**Año**  
2019

**Director**  
Jon Favreau

**Géneros**  
Ciencia ficción, Acción & Aventura, Drama, Fantasía

**Cast**  
Pedro Pascal Katee Sackhoff  
The Mandalorian Bo-Katan Kryze

**Videos: Trailers y Avances**  
[The Mandalorian | Segunda Temporada | Trailer Oficial Doblado | Disney +](#)  
[The Mandalorian Trailer \[HD\] Subtitulado en Español](#)

**Contenido Relacionado**

Tus las historias de Jungo y Boba Fett, otro guerrero emerge en el universo de Star Wars. 'The Mandalorian' se sitúa después de la caída del Imperio y antes de la aparición de la Primera Orden. Seguimos las aventuras de un pistolero solitario en los confines de la galaxia, lejos de la autoridad de la Nueva República.

Detalle Serie

## • Persona

**Pedro Pascal**

**Biografía**  
Pascal nació en Santiago de Chile, de madre psicóloga infantil y padre, José Balmaceda, que es médico de fertilidad. Es nieto de un matrimonio español, una abuela nacida en la isla de Mallorca y un abuelo nacido en el País Vasco. [Leer Mas >](#)

**Información Personal**  
**Conocido Por** **Fecha de nacimiento**  
Interpretación 02-04-1975 (48 años)  
**Lugar de nacimiento**  
Santiago, Chile  
**Redes Sociales**

**Sus obras**

**The Mandalorian** **The Last of Us** **Narcos** **Wonder Woman 1984**

## Watchlist:

**Inicia Sesión:** aparece cuando el usuario no ha hecho login.



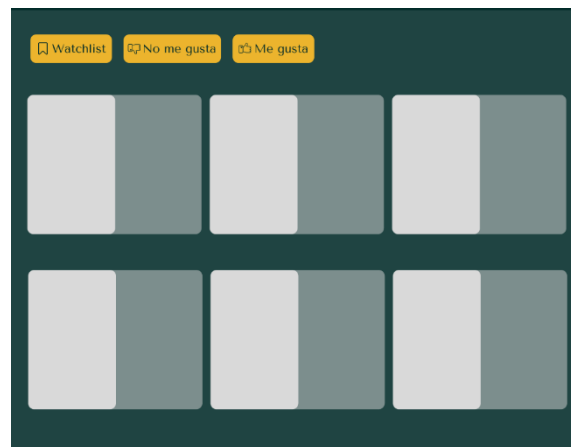
Antes de Iniciar Sesión

**Sin contenido:** cuando no se ha añadido nada a la WatchList.



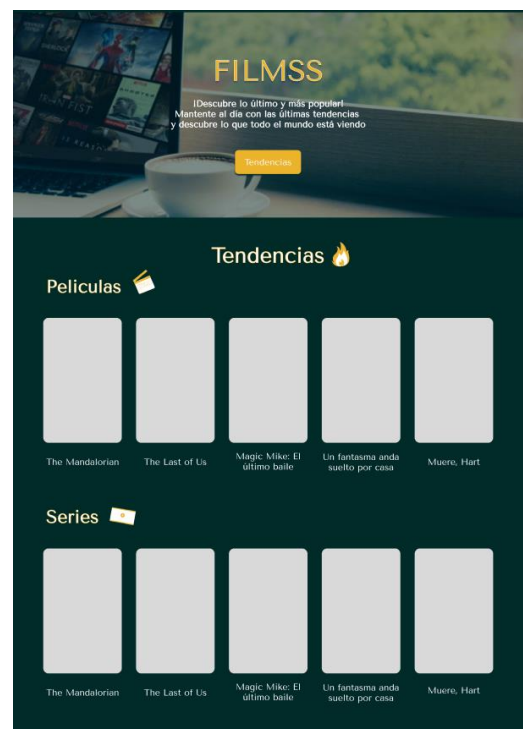
Watchlist sin contenido

## Con contenido:



## Tendencias:

- **Imagen:** esta va variando entre dos opciones
- **Botón:** este botón lleva hacia abajo donde están todas las tendencias.
- **Películas y series:** contenido más relevante actualmente.





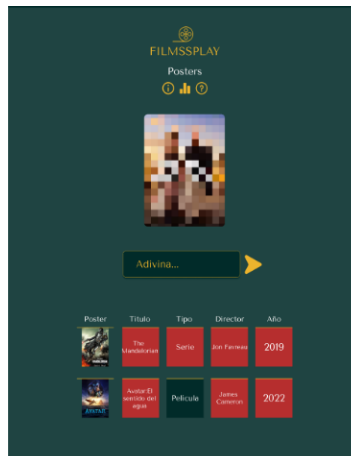
## Juegos:

- **Ahorcado**



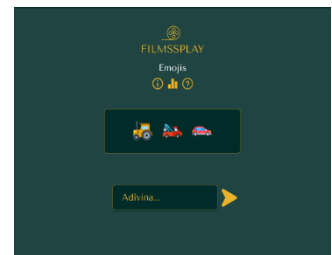
Juego del ahorcado

- **Posters**



Juego de los posters

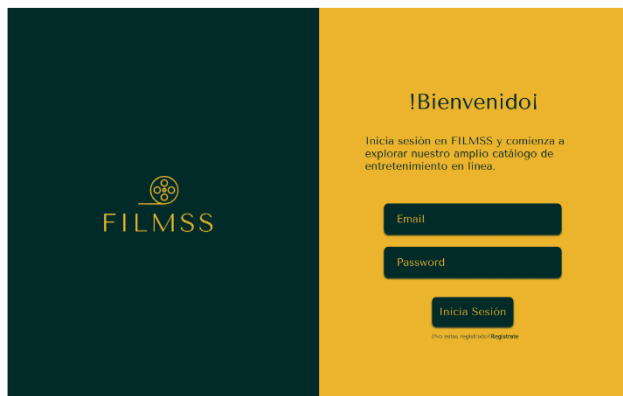
- **Emojis**



Juego de emojis

## Perfil:

- **Login**



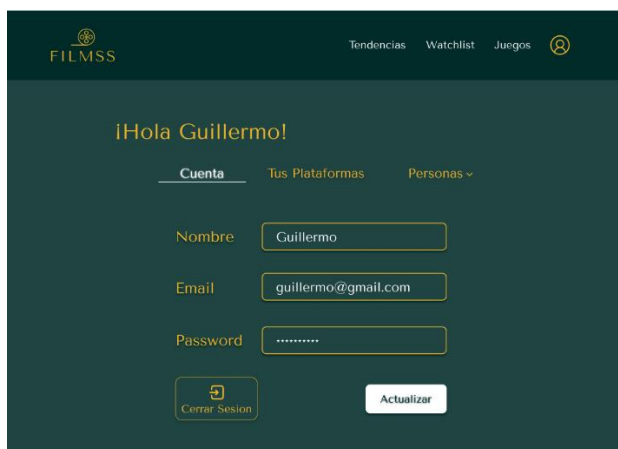
Login

- **Registrarse**

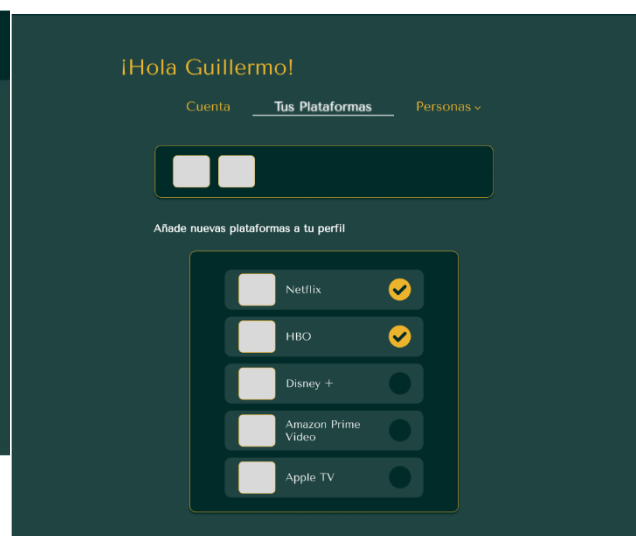


Registro

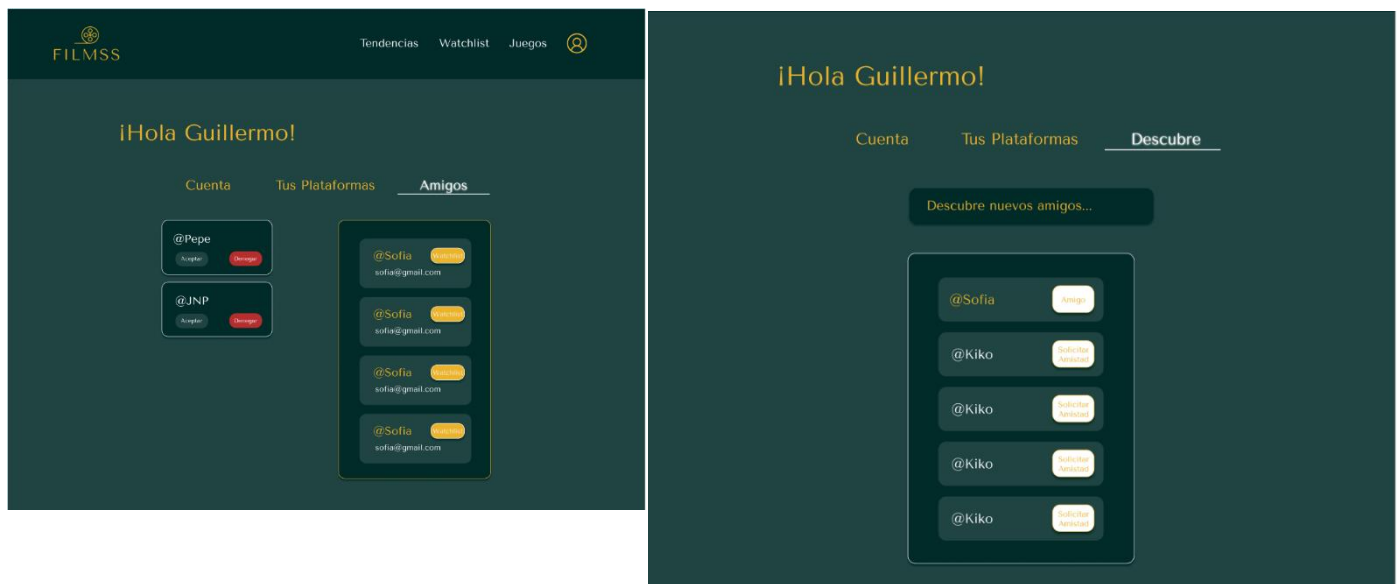
- **Cuenta**



- **Tus Plataformas**



## Personas



## CODIFICACIÓN

### Tecnologías elegidas

En este apartado, se presentarán las tecnologías elegidas para el desarrollo del proyecto. La selección de las tecnologías adecuadas es fundamental para garantizar un desarrollo eficiente, un rendimiento óptimo y una experiencia de usuario satisfactoria. A lo largo de este apartado, se explicará detalladamente cada una de las tecnologías utilizadas, incluyendo los lenguajes de programación, frameworks, librerías y otras herramientas utilizadas tanto en el entorno servidor como en el entorno cliente.

El objetivo principal al elegir estas tecnologías es aprovechar sus características y capacidades específicas para cumplir con los requisitos y objetivos del proyecto. Además, se ha considerado la familiaridad y experiencia previa con algunas de estas tecnologías, lo que permite un desarrollo más rápido y eficiente, así como la confianza en su rendimiento y escalabilidad. Las tecnologías son las siguientes:

- **PHP con Laravel:** se ha seleccionado PHP como lenguaje de programación debido a la experiencia previa con él y el conocimiento de su sintaxis, lo que ha hecho más sencillo el comienzo de este proyecto. Igualmente, también se había usado junto a Laravel, conociendo así su arquitectura y capacidades. Tanto PHP como Laravel son tecnologías de código abierto que ofrecen diversas ventajas en comparación con otras tecnologías.
  - **PHP:** Se trata de un lenguaje de código abierto ampliamente utilizado que cuenta con mucha documentación, así como con una sintaxis sencilla y funcionalidades para la interacción con bases de datos.

- **Laravel:** Es un framework de PHP que proporciona una estructura robusta y eficiente, que ha facilitado la implementación de diversas funcionalidades, como la gestión de rutas, controladores, modelos y vistas.
- **GuzzleHttp\Client:** se ha utilizado esta biblioteca de PHP para acceder y consumir una API externa (TMDB). Esta biblioteca proporciona una interfaz sencilla y eficiente para realizar peticiones HTTP utilizando métodos como GET y POST, entre otros. Además, ofrece beneficios adicionales, como su gran gestión de repuestas, su manejo de errores y su flexibilidad, los cuales han garantizado una obtención de datos confiable y eficiente. La integración de GuzzleHttp en el proyecto ha sido de gran importancia, pues ha permitido establecer comunicación con TMDB para obtener todos los datos necesarios.
- **API TMDB (The Movie Database):** se trata de una base de datos en línea que proporciona información completa y actualizada sobre películas, series de televisión y celebridades. La integración de la API TMDB ha permitido dotar la aplicación con datos relevantes y brindar a los usuarios una experiencia más completa y atractiva. Esta API tiene muchas ventajas frente a otras disponibles en el mercado, entre las cuales destacan su amplio conjunto de datos, que incluye información variada sobre diversos contenidos, como imágenes y vídeos. Además, permite hacer una búsqueda con un filtrado avanzado y mantiene una actualización constante de su contenido, lo que evita que los datos se vuelvan obsoletos.
- **hootlex/laravel-friendships:** es un paquete creado para Laravel, este proporciona funcionalidades predefinidas relacionadas con las amistades entre usuarios. Lo que significa que no es necesario escribir todo el código desde cero para implementar las amistades. Con este paquete, se ha implementado características como agregar amigos, aceptar solicitudes de amistad o denegarlas. En el caso de esta web los usuarios podrán descubrir usuarios, hacerse amigos y ver las Watchlist de sus amigos.
- **Blade:** la elección de Blade como motor de plantillas se debe a que está completamente integrado en el Laravel. Este ofrece una sintaxis intuitiva y legible que facilita la generación de vistas dinámicas. Asimismo, su capacidad para incrustar código PHP en las plantillas permite aprovechar la potencia de Laravel y su motor de enrutamiento, controladores y modelos.
- **Tailwind CSS:** la selección de Tailwind CSS como framework CSS se ha fundamentado en su capacidad para simplificar la creación de estilos, dado que ofrece una amplia gama de clases predefinidas que se pueden aplicar directamente a los elementos HTML, sin la necesidad de escribir CSS personalizado. También permite personalizar los estilos de manera precisa y adaptar la aplicación a los requisitos específicos del proyecto.

- **jQuery y JavaScript:** la inclusión de jQuery y JavaScript se debe a su capacidad para mejorar la interactividad y la experiencia del usuario en el entorno cliente. jQuery simplifica la manipulación del DOM y el manejo de eventos, lo que permite agregar funcionalidades dinámicas y realizar solicitudes asíncronas al servidor. Además, JavaScript es un lenguaje de programación versátil que permite implementar lógica personalizada y mejorar la usabilidad de la aplicación.
- **Bootstrap:** la elección de Bootstrap se debe a su amplia variedad de estilos predefinidos que facilitan crear un diseño atractivo y responsive de manera eficiente. Además, la compatibilidad de Bootstrap con Laravel agiliza la integración de componentes y la personalización del estilo de la aplicación.

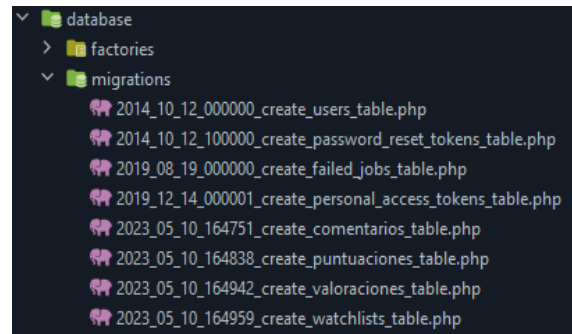
## Entorno servidor

En el entorno servidor, se ha utilizado PHP en combinación con Laravel, estos han proporcionado una base sólida para el desarrollo del proyecto, gracias a su arquitectura MVC (Modelo-Vista-Controlador) y su amplia gama de características y funcionalidades. Se han creado controladores, modelos y vistas siguiendo las convenciones de Laravel, lo que ha facilitado la codificación estructurada. Igualmente, Laravel ofrece una capa de abstracción para la interacción con la base de datos, lo que ha simplificado las consultas y operaciones relacionadas con los datos. Tanto a la hora de crear el MVC como de crear la BD Laravel ofrece distintos comandos y funcionalidades:

- **MVC:** este sistema divide una aplicación web en tres componentes:
  - **Modelo:** se encarga del acceso a los datos de la BD y la lógica necesaria para la interacción y manipulación de los datos.
  - **Vista:** representa la interfaz de usuario y se encarga de presentar los datos de manera estructurada y visualmente atractiva.
  - **Controlador:** actúa de intermediario entre los anteriores componentes, gestionando las interacciones del usuario y actualizando los datos en consecuencia.

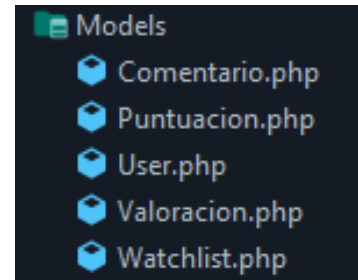
Laravel ofrece un comando específico para crear simultáneamente los archivos correspondientes al Modelo, Controlador y Vista de un recurso en particular. Por ejemplo, al utilizar el comando: ``php artisan make:resource Comentario --model --controller --views``, se generan automáticamente los archivos necesarios para el recurso "Comentario". Este recurso nos resultó útil a la hora de crear las clases, como "Comentario", "Puntuación", "Watchlist" y "Valoración".

- **BD:** para crear las tablas y relaciones entre ellas Laravel ofrece distintas herramientas, la principal son las migraciones. Estas, son archivos que definen la estructura de las tablas y sus relaciones y al ejecutar las, Laravel se encarga de crear las tablas correspondientes en la base de datos.



*Migraciones de la aplicación*

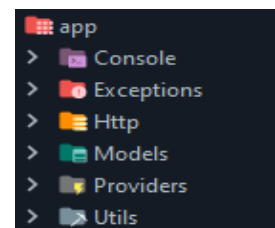
Además, Laravel ofrece Eloquent, con él se pueden definir modelos que representan a las tablas de la base de datos. Estos modelos permiten acceder y manipular los datos de manera orientada a objetos, lo que facilita la interacción con la base de datos y la realización de consultas.



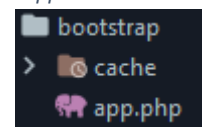
*Modelos de la aplicación*

Por otra parte, Laravel ofrece una estructura de carpetas organizada para facilitar el desarrollo y la administración de aplicaciones:

- **app:** esta carpeta contiene la mayoría de los archivos relacionados con la lógica de negocio de la aplicación. Aquí se encuentran los modelos, controladores y clases auxiliares.
- **bootstrap:** esta carpeta contiene los archivos necesarios para arrancar la aplicación. Aquí se encuentra el archivo `app.php` que configura la aplicación y carga las dependencias necesarias.
- **config:** en esta carpeta se encuentran los archivos de configuración de la base de datos, servicios externos, middleware, y otras opciones aquí.

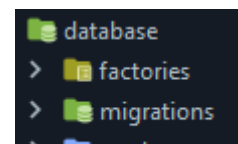


*Carpeta App*



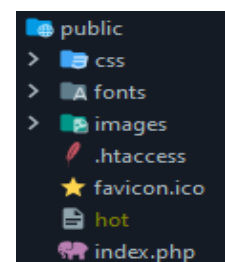
*Carpeta Bootstrap*

- **database:** aquí se encuentran las migraciones (definen los esquemas de la base de datos), las seeders (insertan datos de prueba) y los factories (definen los datos de prueba).



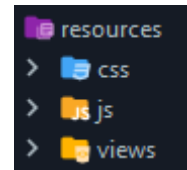
*Carpeta Database*

- **public:** la carpeta *public* es la raíz del servidor web. Aquí se encuentran los archivos accesibles públicamente, como el archivo `index.php` que actúa como punto de entrada para todas las solicitudes HTTP.



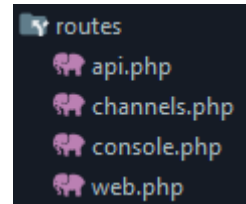
*Carpeta Public*

- **resources:** En esta carpeta se encuentran las *views*, archivos de lenguaje y archivos de *assets* como CSS, JavaScript, imágenes, etc.



Carpeta Resources

- **routes:** Aquí se definen las rutas de tu aplicación, con archivos como `api.php` para las rutas de la API y `web.php` para las rutas web.



Carpeta Routes

Laravel también ofrece archivos importantes para la realización del proyecto:

- **.env:** El archivo `.env` es un archivo de configuración crucial en Laravel. Contiene variables de entorno que se utilizan para configurar aspectos como la conexión a la base de datos, las credenciales de correo electrónico, la configuración de cache, entre otros.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:GbIsayijLD2FH/SHZHRNSE9DJfQE3Jw0P3Wt6pV0Jvk=
APP_DEBUG=true
APP_URL=http://127.0.0.1:8000

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=filmss
DB_USERNAME=guille
DB_PASSWORD=guille
```

- **composer.json y composer.lock:** Estos archivos son parte de la gestión de dependencias de Composer en Laravel. El archivo `composer.json` contiene la lista de paquetes y sus versiones requeridas por tu proyecto, mientras que el archivo `composer.lock` guarda la información específica de las versiones instaladas. Al ejecutar el comando `composer install` o `composer update`, Composer leerá estos archivos y descargará/actualizará las dependencias necesarias en la carpeta `vendor`.

```
{
  "name": "laravel/laravel",
  "type": "project",
  "description": "The Laravel Framework.",
  "keywords": ["framework", "laravel"],
  "license": "MIT",
  "require": {
    "php": "^8.1",
    "guzzlehttp/guzzle": "^7.5",
    "laravel/framework": "^10.0",
    "laravel/sanctum": "^3.2",
    "laravel/tinker": "^2.8"
  }
}
```

Composer.json

- **routes/web.php:** Este archivo define las rutas de la aplicación web, especificando los métodos HTTP correspondientes. Divide en varios tipos de rutas, como las que necesitan autenticación, las que nos llevan a la view o a las que enviamos información al controlador.

```
Route::middleware(['auth'])->group(function () {
    Route::get('/perfil', function () {
        return view('pages.perfil');
    })->name('perfil');
    Route::post('/guardarContenido',[WatchlistController::class,'guardarContenido'])->name('guardarContenido');
    Route::post('/quitarContenido',[WatchlistController::class,'quitarContenido'])->name('quitarContenido');
});

Route::get('/', [ApiController::class,'home'])->name('/');
Route::post('/pelicula/{slug}', [ContenidoController::class,'contenido'])->name('movie');
Route::post('/serie/{slug}', [ContenidoController::class,'contenido'])->name('tv');
Route::post('/person/{slug}', [ContenidoController::class,'contenido'])->name('person');
Route::get('/tendencias', [ApiController::class,'tendencias']);
Route::get('/watchlist',[WatchlistController::class,'index'])->name('watchlist');
Route::get('/login', function () { return view('pages.login');})->name('login');
Route::get('/sign_up', function () { return view('pages.sign_up');});
Route::get('/juegos', function () { return view('pages.juegos');});

Route::get('/obtenerMasResultados', [ApiController::class,'obtenerMasResultados'])->name('obtenerMasResultados');
Route::get('/buscar', [ApiController::class,'search'])->name('explorar.home');
Route::post('/validar-registro',[UserController::class,'register'])->name('registro');
Route::post('/iniciar-sesion', [UserController::class,'login'])->name('inicio.sesion');
Route::get('/logout',[UserController::class,'logout'])->name('logout');
```

*Rutas del proyecto*

## Entorno cliente

En el entorno cliente, se ha utilizado Blade como motor de plantillas, aprovechando su integración con Laravel. Blade permite generar las vistas de manera eficiente y flexible, al permitir incrustar código PHP en los archivos de plantilla.

```
<form method="POST" action="{route($result['media_type'],['slug' => $slug])}" >
```

*Ejemplo de código PHP incrustado en html*

En Laravel, la carpeta "views" es el lugar donde se almacenan los archivos de plantilla Blade:

- **layouts:** esta carpeta contiene archivos actúan como diseños base para las páginas.
  - **master.blade.php** representa el diseño principal del sitio web, el cual incluye elementos comunes:
    - **Nav:** como la barra de navegación
    - **Footer:** el pie de página.
- **pages:** en esta carpeta se encuentran los archivos individuales para cada página. Cada archivo representa una página específica y extiende el archivo "master.blade.php".

Esta estructura permite reutilizar componentes y mantener una estructura coherente en todas las páginas.

Para la estilización de los elementos HTML, se ha utilizado Tailwind CSS, ya que puede evitar crear CSS personalizados e incrustar el estilo directamente en el HTML. Además, se ha podido crear unos estilos predeterminados basándose en la imagen que se quería dar a la web.

```
theme: {
  extend: {
    colors: {
      'green': '#012b29',
      'yellow': '#ecb42d',
      'green2': '#1f4442',
      'blanco': '#fff',
    },
    fontFamily: {
      'sans': ['MiFuente', 'sans-serif'],
    },
  },
},
```

*Estilos definidos en tailwind*

Asimismo, se ha utilizado jQuery junto a JS para capturar eventos y ejecutar diferentes métodos que han ayudado a aligerar la parte del servidor y han mejorado la experiencia del usuario. Por ejemplo, la función de scroll Infinito, pues se ha añadido que mientras se baja se hace una petición Ajax que va cargando las distintas páginas de contenido y las va añadiendo al DOM.

```
$(window).scroll(function () {
  if ($(window).scrollTop() + $(window).height() >= $(document).height() - 1000 && !cargando) {
    pagina++;
    cargando = true;
    $.ajax({
      url: '{{ route("obtenerMasResultados") }}',
      type: 'GET',
      data: {pagina: pagina},
      success: function (response) {
        const nuevosHTML = crearHTMLResultados(response);
        $('#div[id="resultados"]:last').append(nuevosHTML);
      },
      complete: function () {
        cargando = false;
      }
    });
  }
});
```

*Función de Scroll Infinito hecha en JS y jQuery*

Bootstrap también ha sido utilizado en el proyecto para agilizar el diseño y poder hacer responsive parte de él. Ha permitido hacer un sistema de rejilla responsive de manera muy rápida e intuitiva.

```
<div class="col-sm-6 col-md-4 col-lg-2 flex-col flex align-items-center">
```

*Columna responsive hecha con bootstrap*

## DOCUMENTACIÓN

### Manual de usuario

El manual de usuario es una guía que proporciona información detallada sobre cómo utilizar la aplicación, brindando instrucciones claras y concisas para que los usuarios puedan aprovechar al máximo sus funcionalidades. En este caso se ha dividido en diferentes tipos de páginas: las de Autenticación, las públicas y las privadas. A continuación, se adjunta un enlace al [Manual de Usuario.pdf](#).

### Documentación del código

La documentación del código es un aspecto fundamental en el desarrollo de cualquier proyecto de software. Consiste en la creación de comentarios a lo largo de los archivos del proyecto, que describen y



explican el funcionamiento interno del código, su estructura y otros detalles relevantes. Esta documentación tiene como objetivo principal facilitar el entendimiento y mantenimiento del código a lo largo del tiempo, tanto para los desarrolladores que participan en el proyecto como para aquellos que puedan incorporarse en el futuro.

Para incorporar los comentarios se utilizará la funcionalidad “Insert Doc Block” en PhpStorm. Esta característica permite generar un bloque de documentación básico para un método. La funcionalidad opera de la siguiente manera:

- justo arriba del método se ha de escribir `/**`
- después se presiona Enter y se generará automáticamente un bloque de documentación. En él se especificará los parámetros que se le introducen al método y lo que este devuelve.
- el resultado sería algo similar a este método:

```
/**
```

```
/**  
 * @param Request $request  
 * @return \Illuminate\Http\RedirectResponse  
 */
```

```
/**  
 * @param Request $request  
 * @return \Illuminate\Http\RedirectResponse  
 */  
public function quitarContenido(Request $request)  
{  
    try {  
        Watchlist::where('user_id', Auth::id())  
                    ->where('contenido', $request->input( key: 'contenido' ) )  
                    ->delete();  
        return redirect()->route( route: 'watchlist');  
    } catch (QueryException $e){  
        $errorCode = $e->getCode();  
        $mensaje='Se produjo un error al eliminar el contenido.';  
        return redirect()->route( route: 'watchlist')->with('error', $mensaje);  
    }  
}
```

Otros comentarios creados con esta funcionalidad:

```
/**
 * Registra a un usuario en la base de datos.
 *
 * @param Request $request Datos del usuario.
 * @return \Illuminate\Http\RedirectResponse
 */
public function register(Request $request)
{
    $user = new User();
    $user->name = $request->input(key: 'nombre');
    $user->email = $request->input(key: 'email');
    $user->password = encrypt($request->input(key: 'password'));
    $user->save();
    Auth::login($user);
    return redirect(route(name: 'perfil'));
}

/**
 * Realiza el inicio de sesión del usuario.
 *
 * @param Request $request Datos del usuario.
 * @return \Illuminate\Http\RedirectResponse
 */
public function login(Request $request)
{
    $credentials = [
        'email' => $request->input(key: 'email'),
        'password' => $request->input(key: 'password')
    ];
    if (Auth::attempt($credentials, remember: true)) {
        $request->session()->regenerateToken();
        return redirect()->intended(route(name: 'perfil'));
    } else {
        return redirect(to: 'login');
    }
}

/**
 * Cierra la sesión del usuario conectado.
 *
 * @param Request $request Datos del usuario.
 * @return \Illuminate\Http\RedirectResponse
 */
public function logout(Request $request)
{
    Auth::logout();
    $request->session()->invalidate();
    $request->session()->regenerateToken();
    return redirect(to: '/');
}
```

Métodos de UserController

```
/**
 * Genera una vista de la página 'pages.tendencias' con una imagen aleatoria de la matriz $imagenes.
 *
 * @return \Illuminate\Contracts\View\Factory \Illuminate\Contracts\View\View
 */
public function tendencias()
{
    $imagenes = array('images/tendenciasAmigos2.jpg', 'images/tendenciasCafe.jpg');
    // Genera un índice aleatorio dentro del rango de la matriz $imagenes
    $indice = mt_rand(0, count($imagenes) - 1);
    return view(view: '/pages.tendencias', ['imagen_aleatoria' => $imagenes[$indice]]);
}

/**
 * Obtiene una imagen aleatoria utilizando la API de The Movie DB.
 *
 * @param mixed $cliente Objeto cliente para realizar la solicitud a la API.
 * @return string null Ruta de la imagen seleccionada aleatoriamente o null si no se encontraron.
 */
public static function home($cliente)
{
    $imagenes = array();
    $genero = mt_rand(0, 1) == 0 ? 'tv' : 'movie';
    // Inicializa el contador de páginas y construye la URL de la API
    $pages = 1;
    $apiUrl = 'https://api.themoviedb.org/3/discover/' . $genero . '?include_adult=false&language=es&reg
do {
    // Realiza una solicitud GET a la API utilizando el cliente proporcionado
    $response = $cliente->request('GET', $apiUrl . $pages, [
        'headers' => [
            'Authorization' => 'Bearer eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJmZDRmNDE2NWlON2YzN0kyZGNmMDE3
            'accept' => 'application/json',
        ],
    ]);
    // Decodifica la respuesta JSON obtenida
    $responseData = json_decode($response->getBody(), associative: true);
    // Recorre los resultados y agrega las imágenes con popularidad mayor a 7 a la matriz $imagenes
    foreach ($responseData['results'] as $result) {
        if ($result['popularity'] > 7) {
            array_push($imagenes, $result);
        }
    }
    $pages++;
} while ($pages <= $responseData['total_pages']);
$indiceAleatorio = mt_rand(0, count($imagenes) - 1);
return $imagenes[$indiceAleatorio]['backdrop_path'];
}
```

Métodos de ImagenAleatoriaController

Para concluir, la documentación del código es una práctica que contribuye a la calidad, mantenibilidad y colaboración en el desarrollo de software. Es una herramienta clave para comprender y mejorar el código a lo largo del tiempo, y se considera su elaboración una parte integral del proceso de desarrollo.

## DESPLIEGUE

### Diagrama o explicación del despliegue

El despliegue es una etapa fundamental en el ciclo de vida de un proyecto de software, en la cual se implementa y pone en funcionamiento la aplicación en un entorno de producción. En el caso de este

proyecto, se ha optado por utilizar Docker para facilitar el despliegue y asegurar la consistencia y portabilidad del entorno.

Docker es una herramienta de virtualización que permite crear y gestionar contenedores ligeros y autónomos que encapsulan las dependencias y configuraciones necesarias para ejecutar una aplicación. Con Docker, se pueden crear contenedores con imágenes que contengan todas las dependencias del proyecto, incluyendo el sistema operativo, bibliotecas, frameworks y configuraciones.

Los contenedores realizados con Dockers son los siguientes:

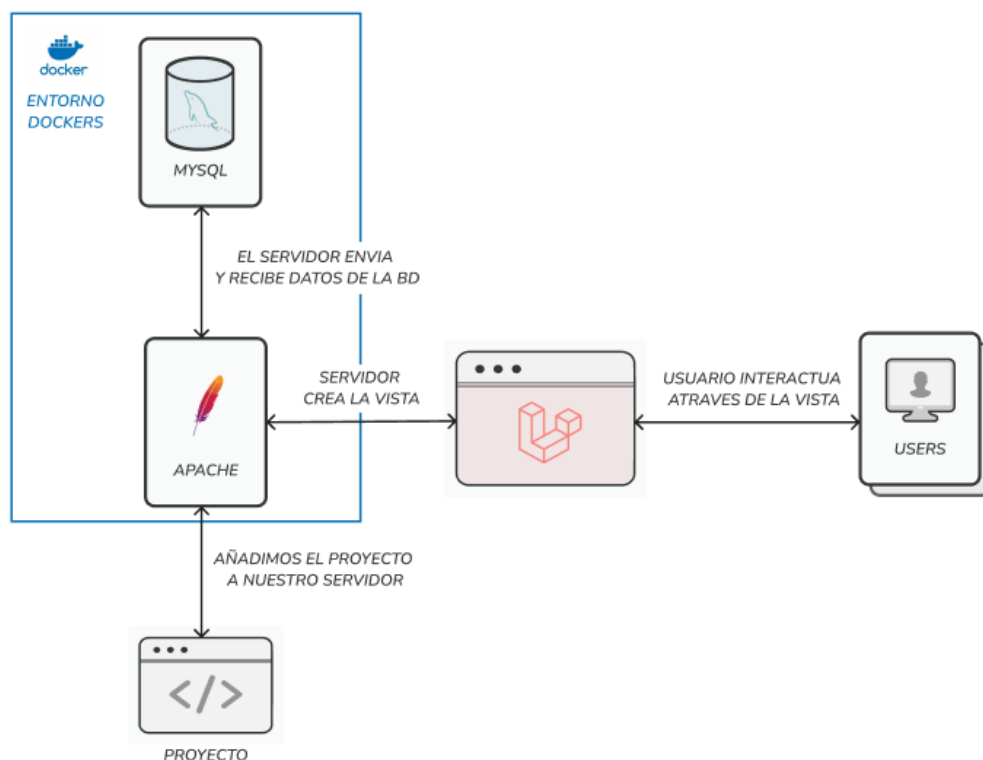
- **Servidor Apache:**

- este contenedor alojará el proyecto y lo ejecutará
- tiene dependencias tanto de PHP como de Laravel
- se crea con un Dockerfile que después se llama desde el docker-compose

- **Base de datos MySQL:**

- se configurará con distintas credenciales para que el servidor pueda entrar
- el servidor se conectará a este contenedor y creará las tablas
- se crea directamente en el docker-compose

Por lo que, el despliegue consiste en dos contenedores Dockers independientes creados mediante la ejecución de un archivo docker-compose.yml. La estructura de contenedores se vería de la siguiente manera:



La utilización de Docker para el despliegue del proyecto ofrece varias ventajas. En primer lugar, la portabilidad. Los contenedores son independientes del entorno de ejecución, lo que significa que se pueden instalar y ejecutar en diferentes sistemas operativos sin problemas de compatibilidad. Esto facilita la escalabilidad y la migración del proyecto.

## Descripción de la instalación o despliegue

A continuación, se detallan los pasos generales realizados a la hora de desplegar la aplicación en Dockers:

- **Requisitos previos:**
  - instalación de dockers en el sistema operativo donde se realice el despliegue
  - inicialización de dockers desktop
- **Archivos:**
  - *Dockerfile*
    - En este archivo se detalla la configuración del contenedor Apache, el servidor
    - Baja una imagen apache junto con php
    - Se instalan las dependencias de php y se habilita el módulo, también se instala composer ya que el proyecto es Laravel
    - Se copia el proyecto a la carpeta de trabajo
    - Se hacen configuraciones adicionales de apache

```
FROM php:apache
# Instalar dependencias de PHP
RUN apt-get update && apt-get install -y nodejs npm \
    git \
    unzip \
    zlib1g-dev \
    libzip-dev
ENV COMPOSER_ALLOW_SUPERUSER=1

# Instalar y habilitar el módulo de zip para PHP
RUN docker-php-ext-install zip
RUN docker-php-ext-install pdo_mysql

RUN php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');" \
    && php composer-setup.php --install-dir=/usr/local/bin --filename=composer \
    && php -r "unlink('composer-setup.php');"

# Copiar archivos de la aplicación y dependencias de Composer
COPY ./filmss /var/www/html/TFG/filmss
WORKDIR /var/www/html/TFG/filmss

# Configurar permisos de la carpeta
RUN find /var/www/html/TFG/filmss -type d \( -name vendor -o -name node_modules \) -prune -o -exec chown www-data:www-data {} +

# Copia el archivo de configuración de Apache
COPY apache-config.conf /etc/apache2/sites-available/000-default.conf

# Habilita el módulo rewrite de Apache
RUN a2enmod rewrite

# Configura el dominio virtual en Apache
RUN echo "ServerName filmss.local.com" >> /etc/apache2/apache2.conf

# Reiniciar Apache
RUN service apache2 restart
```

Archivo Dockerfile

- *docker-compose.yml*

- Este archivo se encarga de lanzar el Dockerfile y del contenedor mysql
- Apache:
  - Lanza el dockerfile y le pone apacheFilmss de nombre al contenedor
  - Enlaza el puerto 80 del servidor con el 8083 de la máquina real
  - Lo introduce en la redWeb
- MySQL:
  - Baja la imagen mysql y le pone mysqlFilmss de nombre al contenedor
  - Le introduce el archivo my.cf, este tiene configuración necesaria
  - Crea la BD filmss y le pone al usuario root la contraseña root
  - Enlaza los puertos 3306 y lo mete en la redWeb

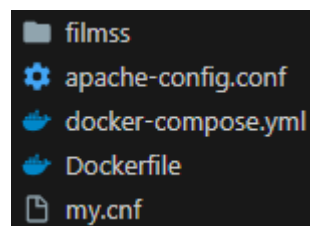
```
version: "3.9"
services:
  #Apache y MySQL
  apache:
    build: #
      context: .
      dockerfile: Dockerfile
    container_name: apacheFilmss
    restart: always
    environment:
      - MYSQL_HOST=mysql
    ports:
      - '8083:80'
    expose:
      - '80'
    networks:
      - redWeb
  mysql:
    container_name: mysqlFilmss
    image: mysql
    volumes:
      - ./my.cnf:/etc/my.cnf
    environment:
      - MYSQL_DATABASE=filmss
      - MYSQL_ROOT_PASSWORD=root
    ports:
      - '3306:3306'
    expose:
      - '3306'
    networks:
      - redWeb
networks:
  redWeb:
```

*Archivo docker-compose.yml*

- **Estructura final:**

- todos los archivos necesarios se encontrarían dentro de la misma carpeta:

- Proyecto (filmss)
- Configuración apache
- docker-compose.yml
- Dockerfile
- Configuración BD



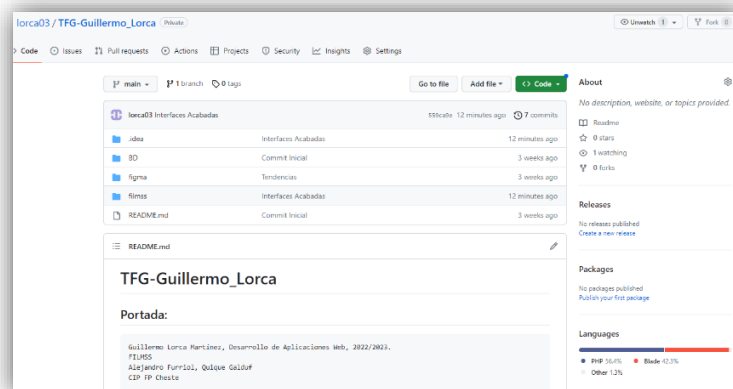
- **Comandos necesarios:** todos dentro de la carpeta donde están los archivos
  - Primero bajar el repositorio e instalar las dependencias del proyecto:
    - “git clone [https://github.com/lorca03/TFG-Guillermo\\_Lorca](https://github.com/lorca03/TFG-Guillermo_Lorca)”
    - “composer install”
    - “npm install”
  - “docker-compose build” crea las imágenes de dockers
  - “docker-compose up” crea los contenedores a partir de las imágenes

Ya se puede acceder a la aplicación a través de <https://localhost:8083> , dejando el puerto libre.

## HERRAMIENTAS DE SOPORTE UTILIZADAS

En el desarrollo de este proyecto, se han utilizado diversas herramientas de soporte que han facilitado la gestión del código, el trabajo remoto y el control de versiones. A continuación, se describirán las principales herramientas utilizadas:

- **Git:** Es un sistema de control de versiones, ampliamente utilizado en el desarrollo de software. Con el que se ha realizado un seguimiento preciso de los cambios efectuados en los archivos del proyecto, permitiendo mantener un historial completo de revisiones. En este caso se ha utilizado Git tanto por línea de comandos, como a través de las funcionalidades que proporciona PHPStrom. Estas incluyen la realización de *commits* para registrar los cambios o la ejecución de diferentes *push* los cuales suben el código a GitHub.  
Git tiene unos archivos importantes que cabe destacar como son el .gitignore o el README. El .gitignore asegura que solo los archivos necesarios y relevantes sean controlados por Git ya que en este archivo se especifica lo que debe ignorar Git. Mientras que el README, proporciona una visión general del proyecto y orientación para los usuarios y colaboradores. Su inclusión en el repositorio ayuda a mantenerlo organizado, facilita la colaboración y proporciona una documentación inicial para su uso y comprensión.
- **GitHub:** Es una plataforma en línea basada en Git que permite alojar y compartir repositorios de código. Además del control de versiones, ofrece características como seguimiento de problemas, revisión de código y colaboración en proyectos de código abierto. Se ha utilizado GitHub como un repositorio remoto para alojar el código del proyecto. Esto ha permitido la revisión de cambios y la integración continua con otras herramientas.



Repositorio del proyecto - [https://github.com/lorca03/TFG-Guillermo\\_Lorca](https://github.com/lorca03/TFG-Guillermo_Lorca)

La elección de estas herramientas se ha basado en su amplia adopción en la comunidad de desarrollo de software, su funcionalidad robusta y su capacidad para mejorar la colaboración y el control de versiones. El uso de Git ha permitido un seguimiento preciso de los cambios realizados en el código, mientras que GitHub ha brindado una plataforma centralizada para alojar y gestionar el repositorio.

## BIBLIOGRAFÍA

- @guillermolorcamartinez | Linktree. (s.f.). Linktree. [@guillermolorcamartinez | Árbol de enlaces \(linktr.ee\)](#)
- de febrero de 2018, de la Dirección General de Empleo, por la que se registra y publica el XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública. (s/f).
- *Laravel - the PHP framework for web artisans*. (s/f). Laravel.com. Recuperado el 19 de mayo de 2023, de [Laravel - El framework PHP para artesanos web](#)
- *Manual de PHP*. (s/f). Php.net. Recuperado el 19 de mayo de 2023, de [PHP: Manual de PHP - Manual](#)
- PricewaterhouseCoopers. (s/f). *Global entertainment & media Outlook 2022–2026*. PwC. Recuperado el 19 de mayo de 2023, de [Global Entertainment & Media Outlook 2022–2026: TMT | PwC](#)

- *Rapidly build modern websites without ever leaving your HTML.* (s/f). Tailwindcss.com. Recuperado el 19 de mayo de 2023, de <https://tailwindcss.com>
- *Stack Overflow - where developers learn, share, & build careers.* (s/f). Stack Overflow. Recuperado el 19 de mayo de 2023, de [Desbordamiento de pila: donde los desarrolladores aprenden, comparten y construyen carreras \(stackoverflow.com\)](#)
- (2000). Gob.es. [Anuario de Estadísticas Culturales 2022. Principales resultados \(culturaydeporte.gob.es\)](#) Pág. 44(18)