

T-VER

PATRICIA GÓMEZ HERNÁNDEZ

2º DESARROLLO APLICACIONES WEB

Índice

1.	Análisis previo	2
1.1.	Descripción del producto o proyecto a realizar.	2
1.2.	Descripción de la empresa.	2
1.3.	Justificación del proyecto.	3
1.4.	Planificación del proyecto.	4
2.	Modelado: Análisis y diseño.	5
2.1	Diagramas de casos de uso	5
2.2	Diagramas de ER y Relacional y scripts	11
2.3	Diagrama de Clases	13
2.4	Diseño de Interfaces	14
2.5	Diagrama de Secuencia	15
3	Implementación, despliegue y pruebas realizadas	17
3.1.	Proceso de realización de proyecto en local	17
3.2.	Desarrollo del proyecto	18
3.3.	Proceso de subir el proyecto a servidor	19
4	Memoria Económica	20
4.1.	Recursos materiales	20
4.2.	Recursos humanos	20
4.3.	Coste de proyecto	20
5	Conclusiones	21
5.1.	Abstract de proyecto	21
6.	Bibliografía	22
	ANEXO I	23
	ANEXO II	24
	ANEXO III	27
	ANEXO IV	28

1. Análisis previo

1.1. Descripción del producto o proyecto a realizar.

Nombre del proyecto: T-View

Descripción:

El proyecto consiste en un página web dedicada al cine. En ella, el usuario podrá, mediante un filtrado, obtener una serie de películas sugeridas, de las cuales podrá elegir una para guardar en su lista de “vistas”. El usuario también podrá ver una lista con las películas que tiene marcadas como “vistas”.

Si no se está logueado, no aparecerá la opción para guardar la película en la lista. Pero si que se podrá acceder al filtrado y a su resultado.

Habrà un usuario Administrador, que será el responsable de algunos aspectos como la creación de películas. Dispondrá de las mismas opciones que un usuario normal, pero además podrá hacer una gestión de todos los demás usuarios.

El proyecto se desarrollará con phpmyadmin, php, html y css.

La acción principal de la página es el filtrado. Para el desarrollo más completo también se ha creado una gestión de usuarios, un formulario para el alta de nuevas películas desde el cual además se pueden añadir nuevos países y nuevos directores, desde sus respectivos formularios, y un listado con todas las películas que hay en la base de datos. Estas funcionalidades están sólo disponibles para el usuario con el rol de Administrador, si el rol es de User solo podrá ver un listado con sus películas y realizar el filtrado con la opción de guardar.

Este producto se desarrolla como una ampliación de otro proyecto de gestión de críticas de películas.

Tenemos en cuenta todas las posibilidades que tenemos para la posible ampliación del producto. En la siguiente versión tenemos como objetivo una valoración de los usuarios para las películas, y así poder colocar un ranking con las películas más votadas. También consideramos la adicción de nuevas funcionalidades para los usuarios administradores, como por ejemplo poder modificar películas y modificar usuarios.

1.2. Descripción de la empresa.

La empresa estará ubicada en Zaragoza. Y está formada por dos socios.

Será una Sociedad Limitada (S.L.) como forma jurídica, ya que podemos constituirla con un capital de 3006€ y nos permite, como socios, responder a las deudas tan sólo con el capital aportado a la empresa, evitando responder con nuestro patrimonio personal.

El capital de la SL se dividirá en 100 participaciones, entregando 50 a cada uno de los 2 socios, que aportarán 1503€ por persona.

Los 2 formaremos la junta General de Socios, lo que nos permitirá tomar decisiones de forma igualitaria y democrática. Ambos seremos administradores de la empresa, debiendo darnos de alta en el régimen de autónomos.

Nuestro producto va dirigido a todos los públicos, con el auge de las plataformas digitales de películas hemos querido hacer más fácil la elección entre tanta oferta. Es la solución que hemos querido dar a ese vacío que este tipo de plataformas tienen, ya que sus filtros son bastante pobres.

1.3. Justificación del proyecto.

¿Qué se va a hacer?

Se va a crear una base de datos donde se guardarán todas las tablas que se indican en el esquema de Entidad-Relación.

Se creará una interfaz de usuario para que el filtrado y la utilización de esos datos sea fácil para cualquier tipo de usuario.

Además se crearán una serie de utilidades para el usuario que tenga un rol de Administrador para poder gestionar los usuarios y las películas.

¿Por qué se va a hacer? ¿Para qué se va a hacer?

La creación de este proyecto es para facilitar al usuario la elección de una película cuando esté indeciso por lo que puede ver en cualquier plataforma, dado que las plataformas de visionado de películas tienen unos filtros muy generales y no siempre aciertan cuando haces una búsqueda. Simplemente porque los filtrados son bastante simples.

¿Cómo se va a hacer?

Desarrollo completo en un entorno local para una posterior subida a un servidor real en internet. Se realizará con php, html y css.

Algo parecido:

Sólo he encontrado algo parecido en una página de películas alternativas. (Filmin). Pero dado que va dirigido a películas alternativas, el campo se queda muy reducido. Además que su producto solo tiene la parte del filtrado, no deja guardar a cada usuario su propio filtrado.

1.4. Planificación del proyecto.

En primer lugar se realiza una primera planificación para el desarrollo del proyecto, para ello se dividen las tareas en 4 bloques que se van a ir desarrollando simultáneamente.

El primer bloque es el inicio del proyecto, **Análisis e inicio del proyecto**, donde incluimos este apartado. Es un bloque dedicado a la documentación del proyecto, y la realización del Diagrama de Gantt. (Página siguiente).

El siguiente bloque, **Modelo, análisis y diseño**, está dedicado a los esquemas y diagramas propios del proyecto que ayudan a entender el funcionamiento de un solo vistazo. En el siguiente apartado de la memoria se pueden ver todos estos diseños más detallados y explicados.

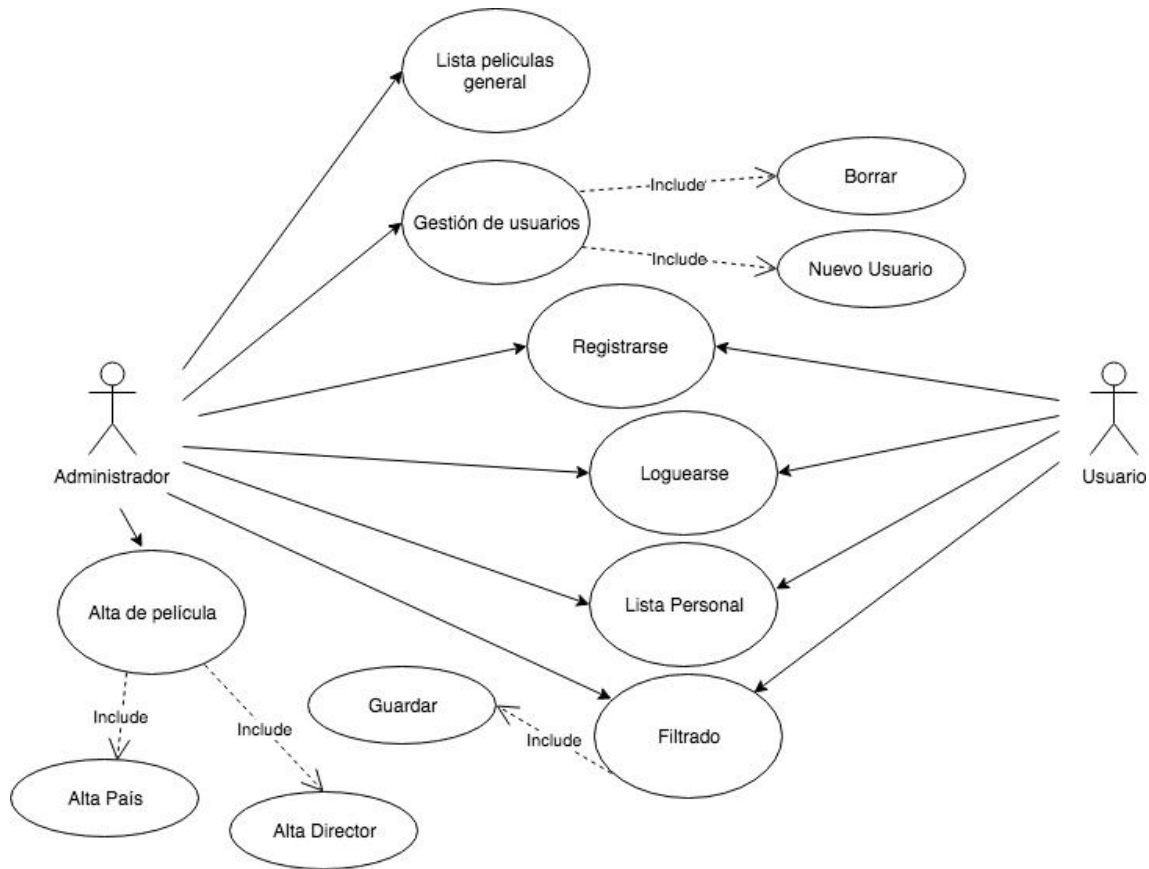
El tercer bloque, **Implementación, despliegue**, se refiere al desarrollo de la aplicación en el entorno. Se tratan los puntos más importantes que se tienen en cuenta a la hora de desarrollar la aplicación en local, para luego subirlo a un servidor web.

El último bloque, **Pruebas y modificaciones**, está dirigido a todas las pruebas que se hacen a lo largo del desarrollo del proyecto, y las modificaciones que se van realizando según van saliendo incidencias y nuevos caminos.

DIAGRAMA DE GANTT _____ Anexo I.

2. Modelado: Análisis y diseño.

2.1 Diagramas de casos de uso



El diagrama de casos de uso nos indica las acciones que puede realizar cada actor. En este sistema encontramos dos actores que son el Administrador y el Usuario, el usuario Administrador tiene las mismas acciones que un usuario normal, pero además tiene la opción de gestionar usuarios, de ver todas las películas que están en la base de datos y un formulario de alta de película, alta director y alta país.

A continuación se presentan los casos de uso para dicho esquema:

T01	Registrarse (Nuevo Usuario)
Descripción	El usuario introduce su nombre y contraseña para que el sistema lo guarde
Actores	Usuario
Precondiciones	Que todos los datos sean correctos.
Curso	<ol style="list-style-type: none"> 1. El usuario introduce su nombre y contraseña 2. El sistema recibe los datos 3. Si los datos son correctos: <ol style="list-style-type: none"> a. El sistema guarda al usuario a su base de datos b. Hace login para el usuario 4. Si los datos no son correctos: <ol style="list-style-type: none"> a. El sistema da un mensaje de error.
Postcondiciones	Ninguna
Alternativa	Utilización de la herramienta sin posibilidad de guardar cualquier consulta

T02	Logging
Descripción	El usuario introduce sus credenciales para poder entrar a su perfil.
Actores	Usuario / Administrador
Precondiciones	Estar registrado.
Curso	<ol style="list-style-type: none"> 1. El usuario introduce su nombre y contraseña 2. El sistema recibe los datos 3. El sistema verifica los datos 4. Si los datos son correctos: <ol style="list-style-type: none"> a. El sistema da acceso al usuario a su sitio 5. Si los datos no son correctos: <ol style="list-style-type: none"> a. El sistema da un mensaje de error de autenticación.
Postcondiciones	Ninguna
Alternativa	Utilización de la herramienta sin posibilidad de guardar cualquier consulta

T03	Filtrado
Descripción	El usuario filtra por una serie de campos para obtener una lista de 4 películas. De las cuales podrá elegir una que se guardará en su lista de películas.
Actores	Usuario / Administrador
Precondiciones	Ninguna
Curso	<ol style="list-style-type: none"> 1. El usuario sigue los distintos filtros que el programa le muestra. 2. El sistema recibe los datos, verifica los datos y devuelve una lista con 4 películas 3. El usuario podrá elegir <ol style="list-style-type: none"> a. Una película: se guardará en su lista b. Volver a filtrar: donde volverá a la primera parte del filtrado
Postcondiciones	Ninguna
Alternativa	Ninguna

T04	Ver Listado Películas
Descripción	El usuario en su sitio puede ver un listado con las películas que ha ido guardando.
Actores	Usuario / Administrador
Precondiciones	Haber guardado alguna película en el listado. De lo contrario aparecerá vacía
Curso	<ol style="list-style-type: none"> 1. El usuario se loguea (T01) 2. El usuario dentro de su perfil elige ver lista
Postcondiciones	Ninguna
Alternativa	Ninguna

T05	Dar de alta películas
Descripción	El administrador tiene una opción para dar de alta nuevas películas en la base de datos a través de un formulario.
Actores	Administrador
Precondiciones	Estar logueado y ser el Administrador
Curso	<ol style="list-style-type: none"> 1. Elige la opción de dar de alta películas 2. Rellena el formulario <ol style="list-style-type: none"> a. si todos los campos son correctos se guarda b. si hay algún campo mal, se marca para que lo vuelva a rellenar 3. Envía el formulario 4. Se guardan los datos en la base de datos
Postcondiciones	Todos los datos han de ser correctos para que se pueda realizar la acción
Alternativa	Ninguna

T06	Dar de alta director
Descripción	El administrador tiene una opción para dar de alta nuevos directores en la base de datos a través de un formulario. Dentro de Alta de películas.
Actores	Administrador
Precondiciones	Estar logueado, ser el Administrador y que el director no esté ya en la base de datos
Curso	<ol style="list-style-type: none"> 1. Elige la opción de dar de alta director, dentro del formulario de alta de película. 2. Rellena el formulario <ol style="list-style-type: none"> a. si todos los campos son correctos se guarda b. si hay algún campo mal, se marca para que lo vuelva a rellenar 3. Envía el formulario 4. Se guardan los datos en la base de datos
Postcondiciones	Todos los datos han de ser correctos para que se pueda realizar la acción
Alternativa	Que el director ya esté dado de alta.

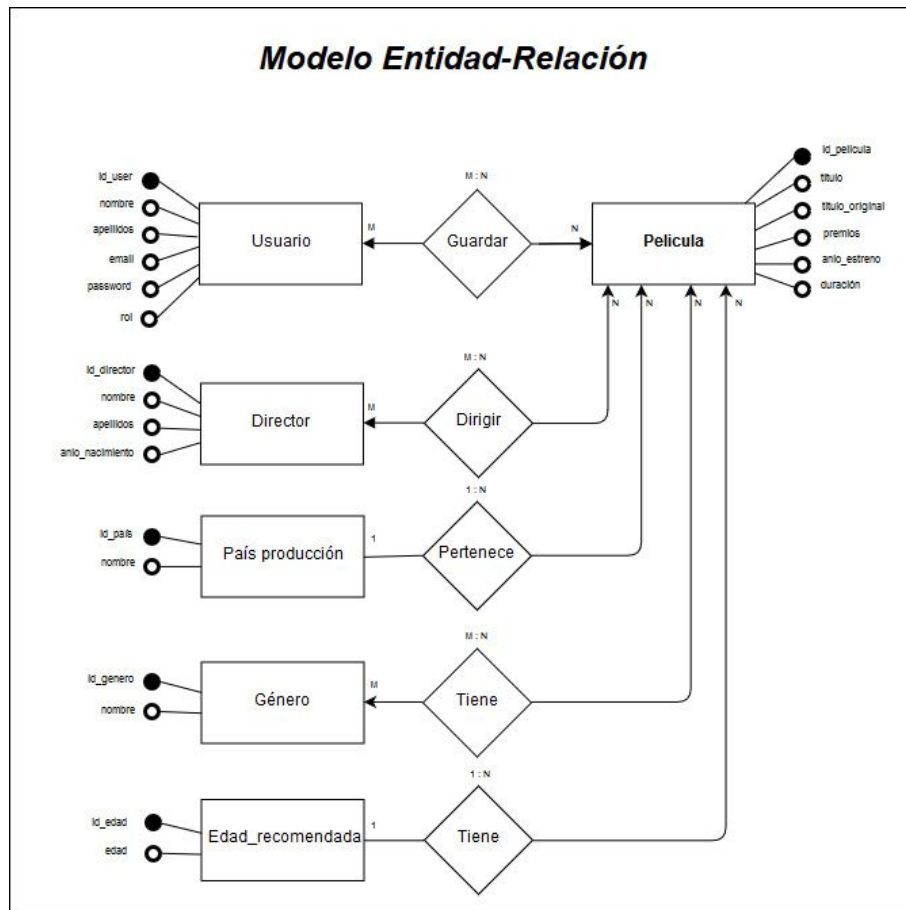
T07	Dar de alta país
Descripción	El administrador tiene una opción para dar de alta nuevos países en la base de datos a través de un formulario. Dentro de Alta de películas.
Actores	Administrador
Precondiciones	Estar logueado, ser el Administrador y que el país no esté dado de alta en la base de datos ya.
Curso	<ol style="list-style-type: none"> 1. Elige la opción de dar de alta país, dentro del formulario de alta de películas. 2. Rellena el formulario <ol style="list-style-type: none"> a. si todos los campos son correctos se guarda b. si hay algún campo mal, se marca para que lo vuelva a rellenar 3. Envía el formulario 4. Se guardan los datos en la base de datos
Postcondiciones	Todos los datos han de ser correctos para que se pueda realizar la acción
Alternativa	Que el país esté ya en la base de datos.

T08	Ver listado completo de la base de datos
Descripción	Visualizar el listado completo de películas que hay en la base de datos
Actores	Administrador
Precondiciones	Estar logueado y ser el Administrador
Curso	<ol style="list-style-type: none"> 1. Elige la opción ver listado completo 2. El sistema busca en la base de datos y devuelve el listado completo de películas que hay guardadas.
Postcondiciones	Puede eliminar las películas que desee con el botón eliminar
Alternativa	Ninguna

T09	Gestión de usuarios (BORRAR)
Descripción	Gestión de usuarios, borrar a un usuario de la base de datos
Actores	Administrador
Precondiciones	Estar logueado y ser el Administrador
Curso	3. Elige la opción de gestión de usuarios 4. Aparece una lista con todos los usuarios y las opciones:
Postcondiciones	El usuario borrado ya no aparecerá en la base de datos y no podrá loguearse
Alternativa	No borrar usuarios.

T10	Guardar Filtrado
Descripción	Guardar la selección de película en la lista que tiene el usuario.
Actores	Administrador y usuario
Precondiciones	Estar logueado y haber realizado el filtrado (T03)
Curso	5. Una vez realizado el filtrado (T03) 6. Elegir una de las opciones que el programa presenta y darle a 'Guardar' 7. El sistema guarda la opción en la lista de películas de cada usuario
Postcondiciones	Ninguna
Alternativa	Volver a filtrar las películas para que salgan otras opciones.

2.2 Diagramas de ER y Relacional y scripts

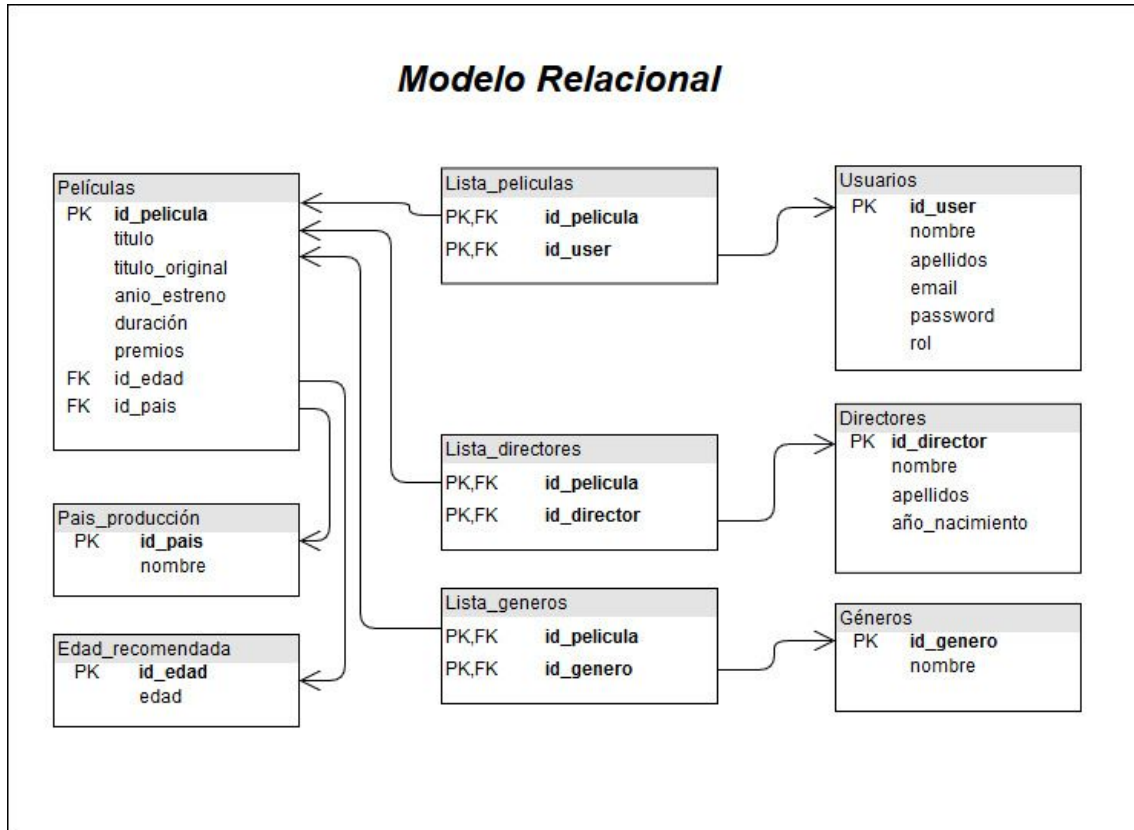


Las dos entidades más importantes del diagrama son **PELÍCULA Y USUARIO**.

La entidad **PELÍCULA**, que contiene varios atributos como son el **título**, **título original**, **premios**, **año de estreno**, y la **duración** que podrán ser distintos en cada una. Su identificador de clave primaria será el **id_película**. Tiene varias claves foráneas, como son **id_edad**, que viene de la tabla **Edad_Recomendada**, **id_país** que viene de la tabla **País_producción**.

La entidad **USUARIO** contiene varios atributos como el **nombre**, **apellido**, **email** y **contraseña**. Su clave primaria será **id_user**. Además tiene un atributo que es **rol**, que tiene dos valores, **Admin** para el administrador y **Usuario**, para todos los demás usuarios.

Además hay unas tablas intermedias como: **Lista_Directores**, con clave primaria compuesta por el **id_película** proveniente de la tabla **PELÍCULA** y el **id_director** proveniente de la tabla **Director**. **Lista_generos**, con clave primaria compuesta por el **id_película** que viene de la tabla **PELICULA** y el **id_género** que proviene de la tabla **Género**. Y por último, **Lista_películas** con clave primaria compuesta por el **id_película** proveniente de la tabla **PELICULA** y el **id_user** que proviene de la tabla **USUARIO**.



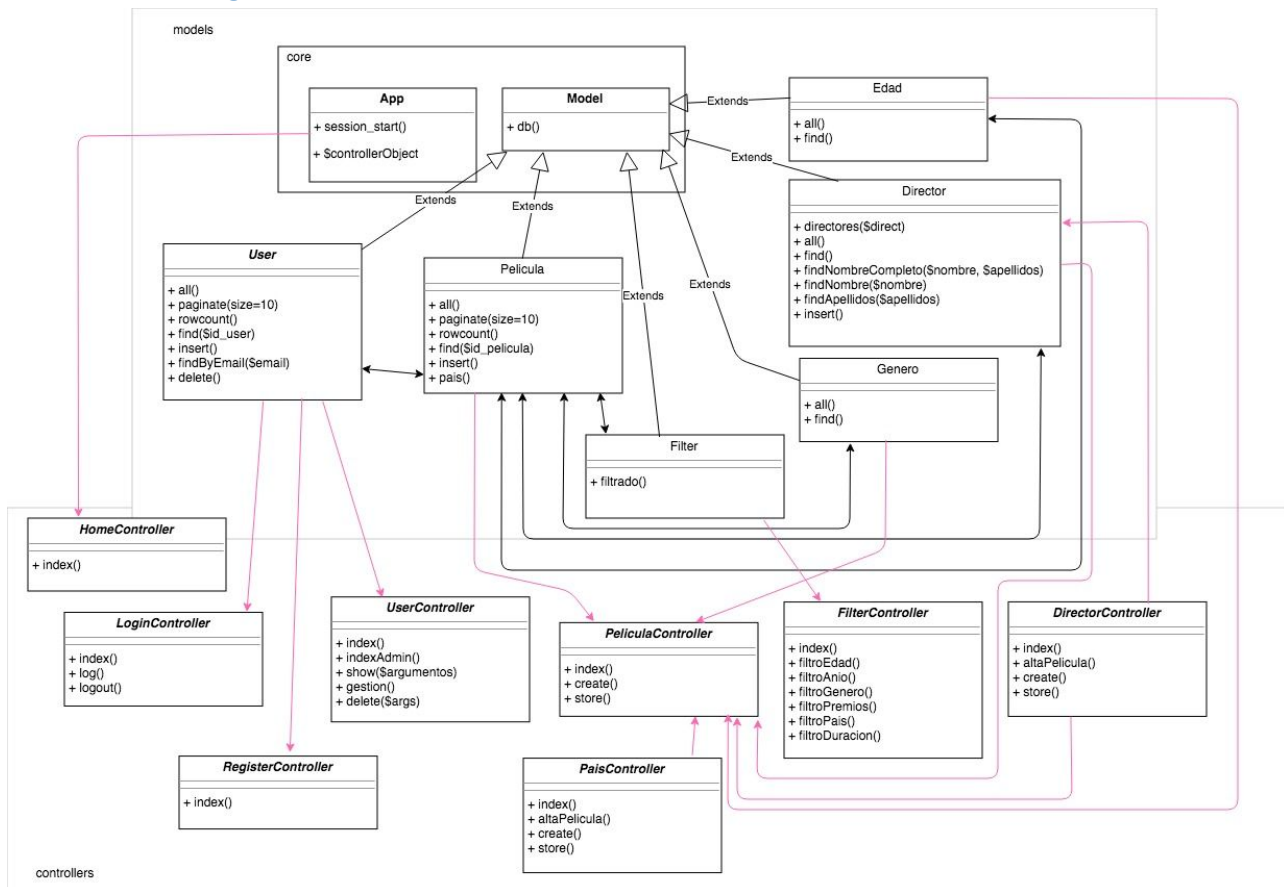
Del esquema Entidad-Relación podemos sacar las tablas que nos harán falta para nuestra base de datos, y las claves que las conformarán.

Tenemos unas tablas que vemos que estarán a simple vista, como son las tablas de **USUARIOS, PELICULAS, PAIS_PRODUCCION, DIRECTORES, GENEROS o EDAD_RECOMENDADA**, las otras tres tablas, **LISTA_PELICULAS, LISTA_DIRECTORES y LISTA_GENEROS**, aparecen de la relación que tienen las demás tablas. Al ser relaciones de muchos a muchos necesitamos estas tablas intermedias para guardar la información y que no se pierda nada.

Scripts utilizados para la generación de las tablas _____ ANEXO II.

Scripts utilizados para consultar en la aplicación _____ ANEXO II.

2.3 Diagrama de Clases



El proceso comienza en la clase App, que es donde se crea la nueva aplicación. Aunque la clase principal será la clase Model, es la que conecta con la base de datos y por ello todas las clases de Model extienden a esta. Después todas las clases del Controllers, están relacionadas con sus respectivas clases model, a excepción de algunas como PaisController que está vinculada a PeliculaController, y RegisterController y LoginController que están desarrolladas sobre la clase model User.

Dependiendo de las necesidades de cada controlador y de cada modelo, se van necesitando unas a otras. Las flechas rosas marcan las relaciones que hay entre los modelos y los controladores. Y las relaciones entre los modelos se marcan con flechas negras.

2.4 Diseño de Interfaces



El inicio de la aplicación comienza en la página presentación donde se incluye el logo, un cono para el filtrado y el login/ sitio, dependiendo si ya se está logueado o no. La primera acción que se puede realizar de este modo es ir al filtrado (si se quiere hacer sin estar logueado) o loguearse. Si es la primera vez que se entra y no está registrado el usuario, se puede registrar. En el caso de registrarse, se va al formulario, se rellena y al aceptar se redirige a la página principal del usuario que es la lista con sus películas seleccionadas.

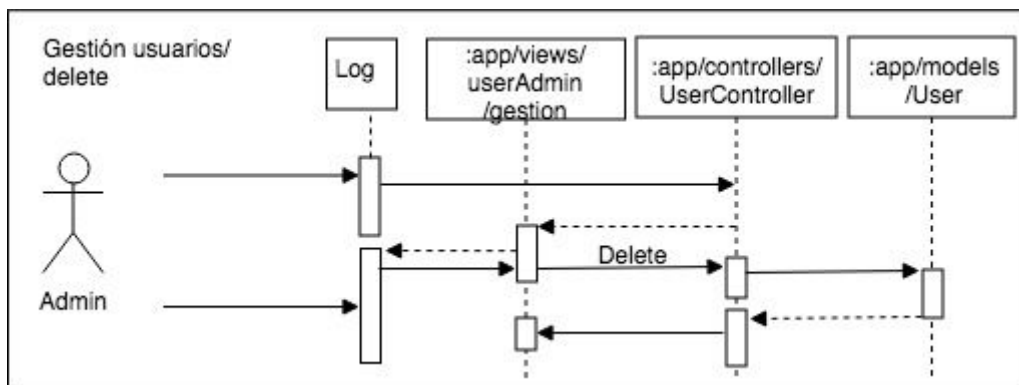
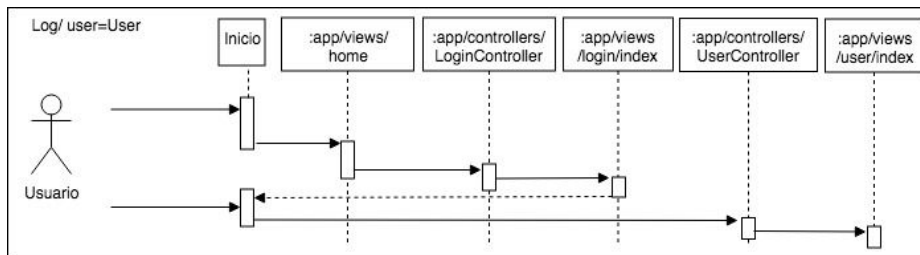
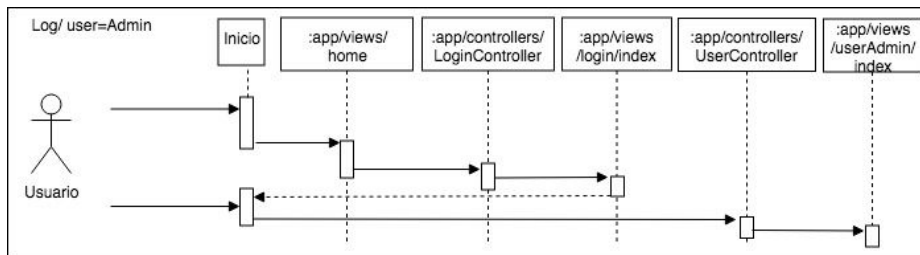
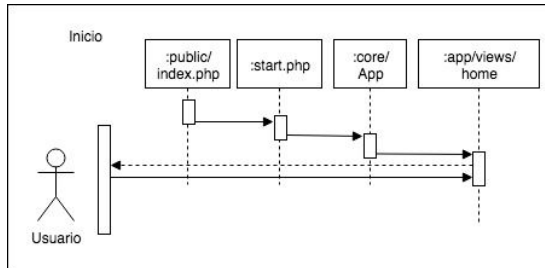
Si el usuario es Administrador las opciones se amplían, el Admin tiene un menú principal más extenso, donde puede ver las películas que ha seleccionado él, puede gestionar los usuarios (ver y borrar usuarios), dar de alta películas, dentro de esta opción puede dar de alta países, y directores. Y también puede ver una lista general con todas las películas que se encuentran en la base de datos.

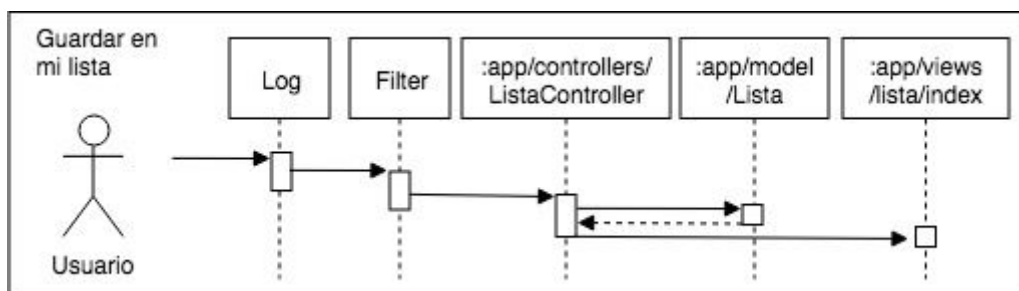
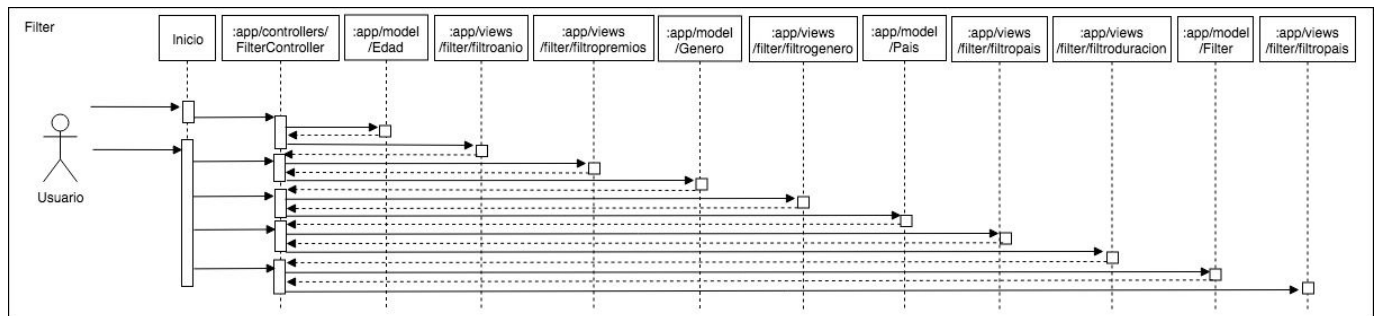
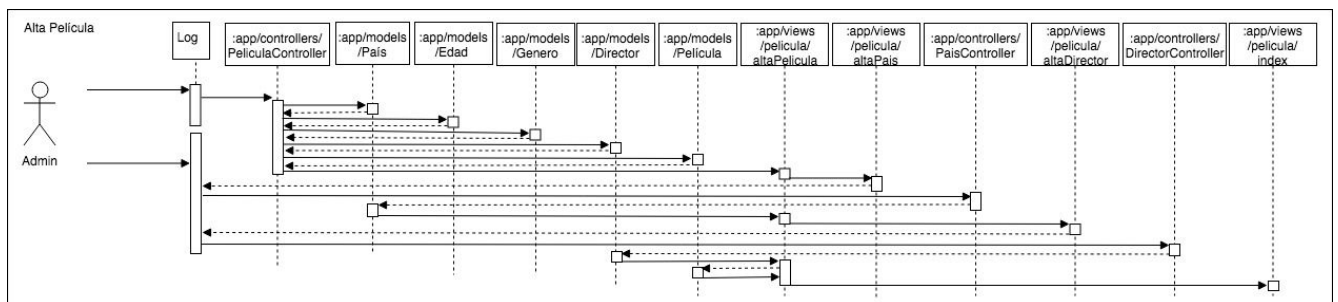
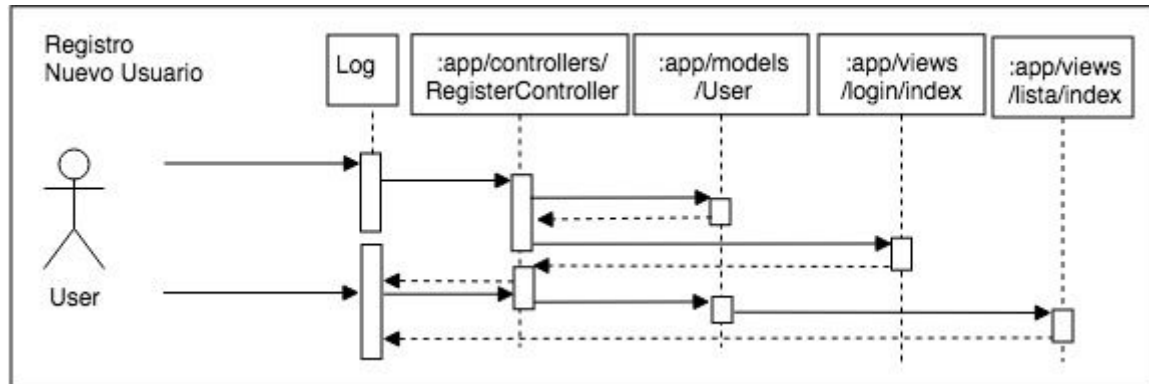
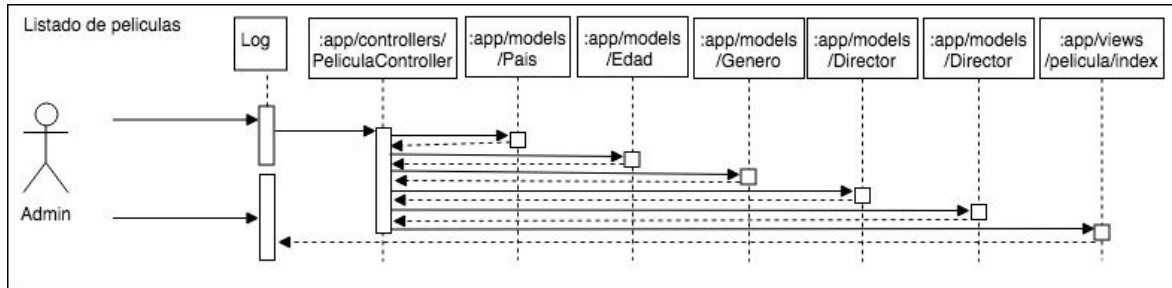
El filtrado lo pueden hacer todos los usuarios que visiten la página, estén o no estén logueados. Consiste en una serie de pantallas, donde se van mostrando unas opciones de filtrado para que el usuario vaya eligiendo, año de estreno, edad recomendada, si la película tiene premios, el género, el país y la duración. Después de esta serie de filtros, al usuario se le muestran cuatro opciones de las cuales puede elegir una y si está logueado la podrá guardar en su lista de películas. Si no le gustan las opciones tiene la opción de volver a filtrar.

Imagen ampliada del diseño _____ ANEXO III

2.5 Diagrama de Secuencia

En los siguientes diagramas se explica secuencialmente el trayecto que lleva la ejecución del programa. Lo he dividido en varias partes para que sea más fácil la lectura y comprensión. Además de que cada proceso es distinto.





3 Implementación, despliegue y pruebas realizadas

3.1 Proceso de realización del proyecto en local

El proyecto se realiza desde una máquina virtual de Ubuntu.

Creación de la base de datos en phpMyAdmin:

Debemos de entrar con nuestro usuario y contraseña, que deberemos recordar para colocarlas en la clase Model de nuestro proyecto. Una vez iniciada la sesión simplemente le damos a 'Nueva', y le damos un nombre.



Cuando ya está creada la base de datos se ejecutan los scripts para la creación de todas las tablas y para rellenar algunos campos que son esenciales para el funcionamiento del proyecto.

Instalación de Apache:

- sudo apt-get update
- sudo apt-get install apache2

Creación del fichero /etc/apache2/sites-available/cinecito.conf para poder trabajar en local en el sitio indicado en dicho fichero.

```
<VirtualHost *:80>
    ServerName cinecito.local
    ServerAdmin webmaster@localhost
    DocumentRoot /home/alumno/cinecito/public
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    <Directory /home/alumno/cinecito/public>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Activar el servicio y reiniciar apache:

- `sudo a2ensite cinecito`
- `sudo service apache2 restart`

Editar el fichero `/etc/hosts` y añadir :

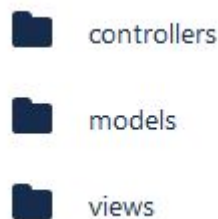
- `127.0.0.1 cinecito.local`

Utilización de Sublime text 3 como programa principal para el desarrollo del código.

Una vez creado el proyecto con su conexión, empezamos con el desarrollo de la estructura del proyecto.

3.2 Desarrollo del proyecto

Creación del modelo básico de directorios para MVC.



Creación de la vista `home.php` que será la página principal del sitio. Cada página principal de cada vista específica irá dentro de una carpeta con su `index` y sus vistas correspondientes dependiendo de las funcionalidades. En casi todos los casos irá acorde con un `controller` y con un `modelo`.

1. Desarrollo del modelo `USER`, con su respectivo controlador. Se crean las vistas de `user` y de `userAdmin`, pero lo que se desarrolla es la vista `login`, con un controlador para ello.
2. Creación del registro de usuarios. (El registro para el administrador se crea directamente en la base de datos). Se crea la vista en `register` y el controlador para el registro.
3. Creación del `index` de película donde se visualizará la lista con todas las películas que hay en la base de datos. En esta parte se crea el modelo `Película` y su controlador.
4. Creación del formulario para la inserción de películas. Esta parte es la más laboriosa, dado que el formulario tiene bastantes campos. Además dentro de este formulario se incluyen dos enlaces que nos llevan a otros formularios. La inserción de país nuevo, que conlleva un nuevo modelo `País` y un nuevo controlador.

También podemos insertar Directores, creamos un nuevo modelo Director y su controlador. Además creamos un modelo Género para recoger los géneros guardados en la base de datos. Una vez introducidas varias películas en la base de datos, seguimos con el filtrado.

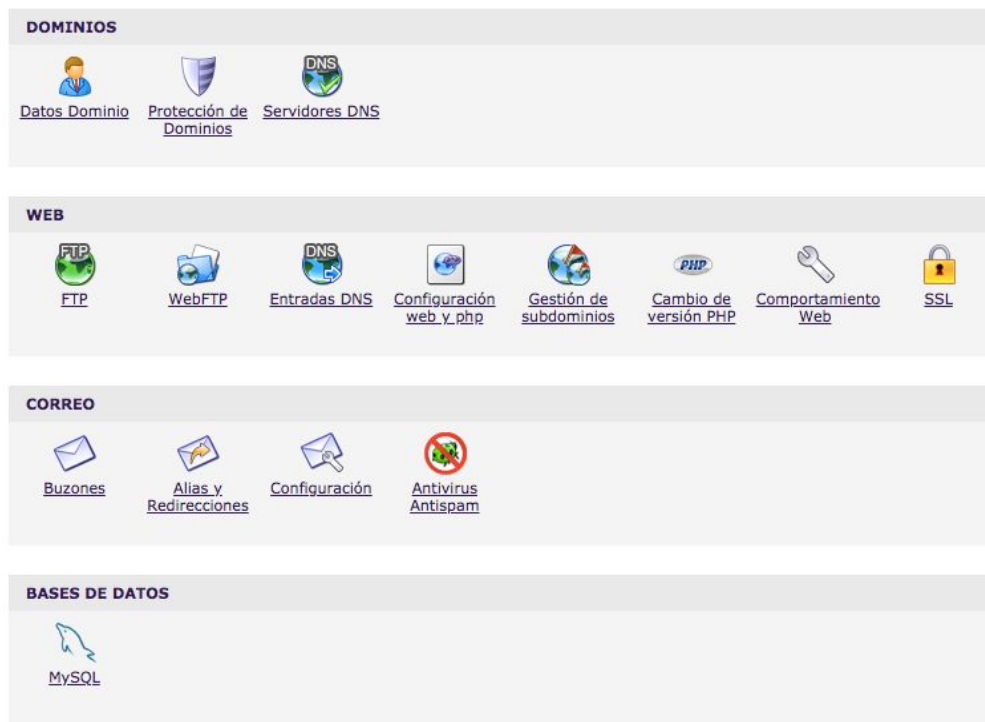
5. Filtrado de películas. Creación de las diferentes vistas para el filtrado. También se crea un modelo Filter y su correspondiente controlador. Desde el controlador se van recogiendo todos los datos que elige el usuario y los guarda en un array para después compararlos con lo que hay en la base de datos y sacar al final el resultado con 4 soluciones.
6. Si se está logueado y se desea guardar uno de los resultados del filtrado, está la opción para guardar en la lista. Se crea una vista para la lista personal de cada usuario, también se crea un modelo Lista y su correspondiente controlador. Que busca en la base de datos si la película seleccionada por el usuario ya está en su lista o no.

3.3 Proceso de subir el proyecto a servidor

Para subir el proyecto a un servidor web, lo primero es contratar un hosting y un dominio, yo los tenía contratados ya.

Mi hosting se encuentra en : www.pinsasolutions.com

Y el dominio es: pathgohe.com



La carpeta raíz de mi sitio es html, como en el proyecto la carpeta raíz es public, se debería de cambiar la raíz a ese directorio, pero al no ser posible, lo que hago es cambiar el index a la carpeta html y desde ahí montar la ruta para que comience la ejecución.

También se exporta la base de datos que está en local y se importa en el servidor. Hay una opción en el panel de PhpMyAdmin que genera un archivo que es el tenemos que importar.

Dificultades:

Al no ser el mismo directorio que el proyecto en local, no me ejecuta el proyecto, se me abre una página en blanco. Tras la consulta con varios profesores, se soluciona parte del problema quitando todas las rutas internas que estaban dirigidas desde la carpeta public y colocando la ruta desde html.

Se ve la página principal, pero no se ejecuta todo correctamente, posiblemente sea porque la conexión con la base de datos no esté bien realizada.

4 Memoria económica

4.1 Recursos materiales

Para el desarrollo del proyecto es necesario un ordenador con un sistema operativo linux, o una máquina virtual que nos permita implementarlo. Tendremos que instalar Apache para la simulación del servidor y configurarlo conforme a lo ya descrito en el punto 3.

4.2 Recursos humanos

El proyecto se puede realizar entre una o dos personas, dado que es una parte pequeña de un proyecto más grande. Al ser dos socios en la empresa, el trabajo se dividiría entre ambas partes. De momento no se necesitaría la incorporación de nuevos trabajadores a la empresa, aunque se tendría en cuenta si la cantidad de trabajo fuese aumentando.

4.3 Costes del proyecto

Tipo de acción	Coste en horas	Coste en euros
Análisis e inicio del proyecto	40	800€
Modelo, análisis y diseño	20	400€
Implementación y despliegue	50	1000€
Pruebas y modificaciones	30	600€
Total:		2800€

Según los puntos desarrollados en el diagrama de Gantt se ha creado una tabla orientativa con las horas invertidas en cada punto y el coste de su realización.

5 Conclusiones

Para la realización del proyecto en primer lugar me planteé la posibilidad de hacerlo con diferentes lenguajes, el primero fue laravel, pero lo descarté porque para la implementación del proyecto en la web, me iba a resultar más compleja e incluso imposible. Así que al final me incliné por realizar el proyecto con php, html y phpmyadmin, porque pensé que sería más sencillo de subir a un servidor web.

Todo el desarrollo ha sido conforme los diagramas realizados. Mientras ha ido avanzando el proyecto se han ido incorporando cosas para mejorar el funcionamiento y la interfaz. Como por ejemplo los formularios para introducir directores y películas.

Se tienen en cuenta diferentes mejoras para que sea más funcional y para ampliar su usabilidad. Por ejemplo un sistema de puntos para las películas, que así se pueda crear un ranking. Una mejora en la gestión de usuarios, que permita la modificación de datos por parte del administrador.

Me hubiese gustado colocar el filtrado con javascript, pero no he sabido desarrollarlo junto con el php. Esta parte podría ser una posible mejora también para la aplicación. Porque nos podríamos ahorrar varias vistas e incluso varias funciones.

Dificultades:

Al crear scripts para los géneros o para directores e insertarlos directamente en la base de datos. Los acentos no salían bien, me dí cuenta de que las inserciones desde la aplicación las hacía bien, así que cree un formulario nuevo para poder guardar todos los géneros y que viesan bien con sus acentos. También es una forma de que se puedan añadir nuevos géneros en cualquier momento.

Dificultad para incorporar diferentes lenguajes que no fuesen php, y html.

Todo el proyecto se ha ido subiendo a bitbucket conforme se ha ido desarrollando para evitar que si la máquina no funcionaba, se perdiese todo el código.

Estructura de ficheros del proyecto _____ ANEXO IV

5.1 Abstract de proyecto

The project is about filtering movies. We have an option in the main menu that takes us directly to the filtering. If we are not logged, once the filtering is done we will not be able to save the chosen option. However, if we are logged, we can save and have the option to see the list with the movies we have been watching.

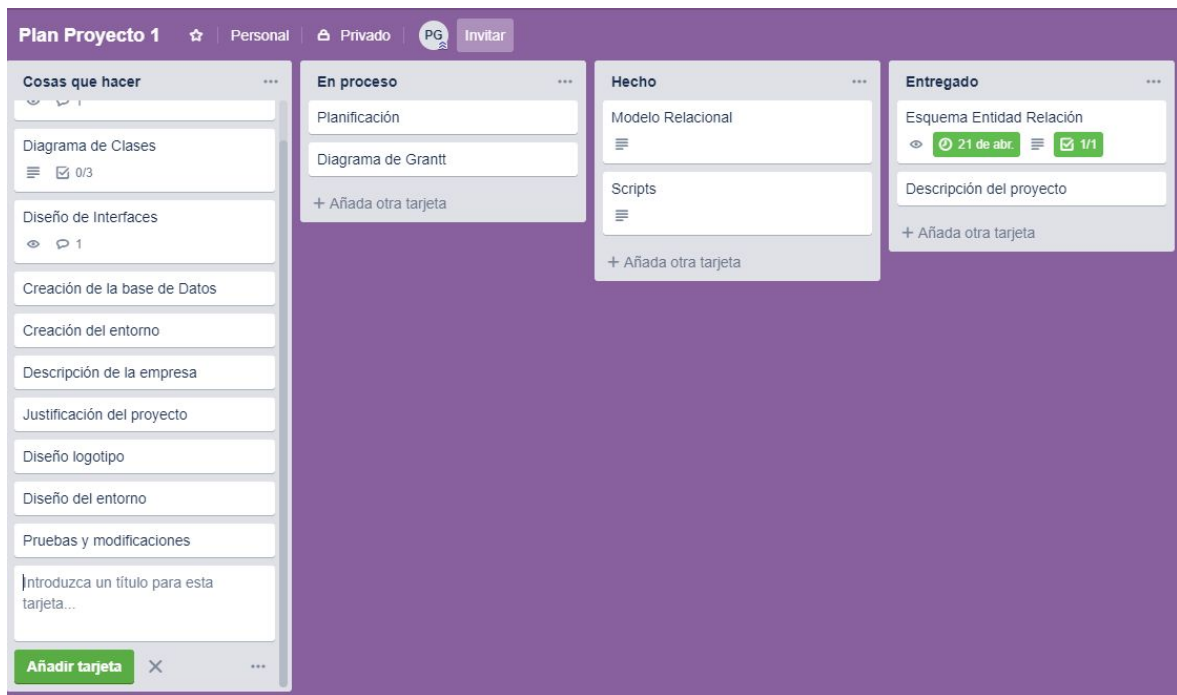
The administrator user also has the options to insert movies, countries and directors. And it has user management. Being the administrator you can also see a complete list of all the movies that are in the database.

The final result is directed so that the user registers and makes filters and keeps the options that he / she visualizes.

6 BIBLIOGRAFÍA

HERRAMIENTAS UTILIZADAS

- Trello: Para una planificación más clara de las tareas que se han de realizar.



- Draw.io : Para la creación de los distintos diagramas.
- Excell : Para la creación del diagrama de Gantt.
- Word : Para el desarrollo del proyecto. Documentación y documentos adjuntos.

Páginas utilizadas:

- <https://www.w3schools.com/bootstrap/>
- <https://rafacabeza.gitbooks.io/dws/content/>

ANEXO II

CREATE DATABASE Aplicación;

```
CREATE TABLE PaisProduccion
(id_pais INT(10) PRIMARY KEY AUTO_INCREMENT NOT NULL,
nombre VARCHAR(25) NOT NULL);
```

```
CREATE TABLE Generos
(id_genero INT(10) PRIMARY KEY AUTO_INCREMENT NOT NULL,
nombre VARCHAR(25) NOT NULL);
```

```
CREATE TABLE EdadRecomendada
(id_edad INT(10) PRIMARY KEY AUTO_INCREMENT NOT NULL,
nombre VARCHAR(3) NOT NULL);
```

```
CREATE TABLE Directores
(id_director INT(10) PRIMARY KEY AUTO_INCREMENT NOT NULL,
nombre VARCHAR(25) NOT NULL,
apellidos VARCHAR(25) NOT NULL,
anio_nacimiento INT(4));
```

```
CREATE TABLE Users
(id_user INT(10) PRIMARY KEY AUTO_INCREMENT NOT NULL,
nombre VARCHAR(25) NOT NULL,
apellidos VARCHAR(25) NOT NULL,
email VARCHAR(25) NOT NULL,
password VARCHAR(25) NOT NULL,
id_rol INT(10) not null);
```

```
CREATE TABLE Peliculas
( id_pelicula INT(10) PRIMARY KEY NOT NULL AUTO_INCREMENT,
titulo VARCHAR(25) NOT NULL,
titulo_original varchar(25),
id_edad INT NOT NULL,
anio_estreno INT(4) NOT NULL,
duracion INT(4) NOT NULL,
premios INT(1) NOT NULL,
id_pais INT NOT NULL,
FOREIGN KEY (id_pais) REFERENCES PaisProduccion(id_pais),
FOREIGN KEY(id_edad) REFERENCES EdadRecomendada(id_edad));
```

```
CREATE TABLE ListaPeliculas
(id_pelicula INT NOT NULL,
id_user INT NOT NULL,
```

```

PRIMARY KEY (id_pelicula, id_user),
CONSTRAINT id_pelicula FOREIGN KEY(id_pelicula) references
Películas(id_pelicula),
CONSTRAINT id_user FOREIGN key(id_user) REFERENCES Users(id_user));

```

```

CREATE TABLE ListaDirectores
(id_pelicula INT NOT NULL,
id_director INT NOT NULL,
FOREIGN KEY(id_pelicula) references Películas(id_pelicula),
FOREIGN KEY(id_director) REFERENCES Directores(id_director),
PRIMARY KEY (id_pelicula, id_director));

```

```

CREATE TABLE ListaGeneros
(id_pelicula INT NOT NULL,
id_genero INT NOT NULL,
FOREIGN KEY(id_pelicula) references Películas(id_pelicula),
FOREIGN key(id_genero) REFERENCES Generos(id_genero),
PRIMARY KEY (id_pelicula, id_genero));

```

INSERCIONES

```

INSERT INTO `Generos` (`id_genero`, `nombre`) VALUES
(1, 'Comedia'), (2, 'Drama'), (3, 'Aventuras'), (4, 'Animación'), (5, 'Terror'), (6, 'Acción'), (7,
'Romántico'), (8, 'Musical'), (9, 'Ciencia-Ficción'), (10, 'Fantástico'), (11, 'Western'), (12,
'Thriller'), (13, 'Documental'), (14, 'Intriga'), (15, 'Cine Negro'), (16, 'Histórico'), (17,
'Bélico'), (18, 'Parodia'), (19, 'Juvenil');

```

```

INSERT INTO `EdadRecomendada` (`id_edad`, `nombre`) VALUES
(1, '+7'), (2, '+13'), (3, '+18'), (4, 'R');

```

SCRIPTS REALIZADOS DENTRO DE LA APLICACIÓN

Model/User

```

SELECT * FROM Users;
SELECT * FROM Users LIMIT :pagesize OFFSET :offset;
SELECT count(id_user) as count FROM Users;
SELECT * FROM Users WHERE id_user=:id_user;
INSERT into Users(nombre, apellidos, email, password,rol)
values(:nombre, :apellidos,:email, :password, :rol);
SELECT * FROM Users WHERE email=:email;
DELETE FROM Users WHERE id_user LIKE :id_user;

```

Model/Pelicula

```

SELECT * FROM Películas;
SELECT * FROM Películas WHERE id_pelicula=:id_pelicula;
SELECT * FROM Películas LIMIT :pagesize OFFSET :offset;

```

```

SELECT count(id_película) as count FROM Películas;
INSERT into Películas(título, títuloOriginal, id_edad, añoEstreno, duración,
premios, id_pais) values(:título, :títuloOriginal, :id_edad, :añoEstreno, :duración, :premios,
:id_pais);
INSERT into ListaGeneros (id_película , id_genero) values (:id_película, :genero);
INSERT into ListaDirectores (id_película , id_director) values (:id_película,
:director1), (:id_película, :director2);
INSERT into ListaDirectores (id_película , id_director)values(:id_película,
:director1);
SELECT * FROM PaisProduccion WHERE id=:id;

```

Model/Pais

```

SELECT * FROM PaisProduccion WHERE id_pais = :id;
SELECT * FROM PaisProduccion;
SELECT * FROM PaisProduccion WHERE nombre=:nombre;
INSERT into PaisProduccion(nombre) values(:nombre);

```

Model/Lista

```

SELECT * FROM ListaPelículas;
SELECT * FROM ListaPelículas WHERE id_user=:id_user;
INSERT into ListaPelículas(id_película,id_user) values(:id_película, :id_user);
SELECT * FROM Películas WHERE id_película=:id_película;

```

Model/Genero

```

SELECT * FROM Generos;
SELECT * FROM Generos WHERE id_genero=:id_genero;
INSERT into Generos(nombre) values(:nombre);

```

Modelo/Filter

```

SELECT * FROM ListaGeneros WHERE id_genero=:genero;

```

Model/Edad

```

SELECT * FROM EdadRecomendada;
SELECT * FROM EdadRecomendada WHERE id_edad=:id_edad;

```

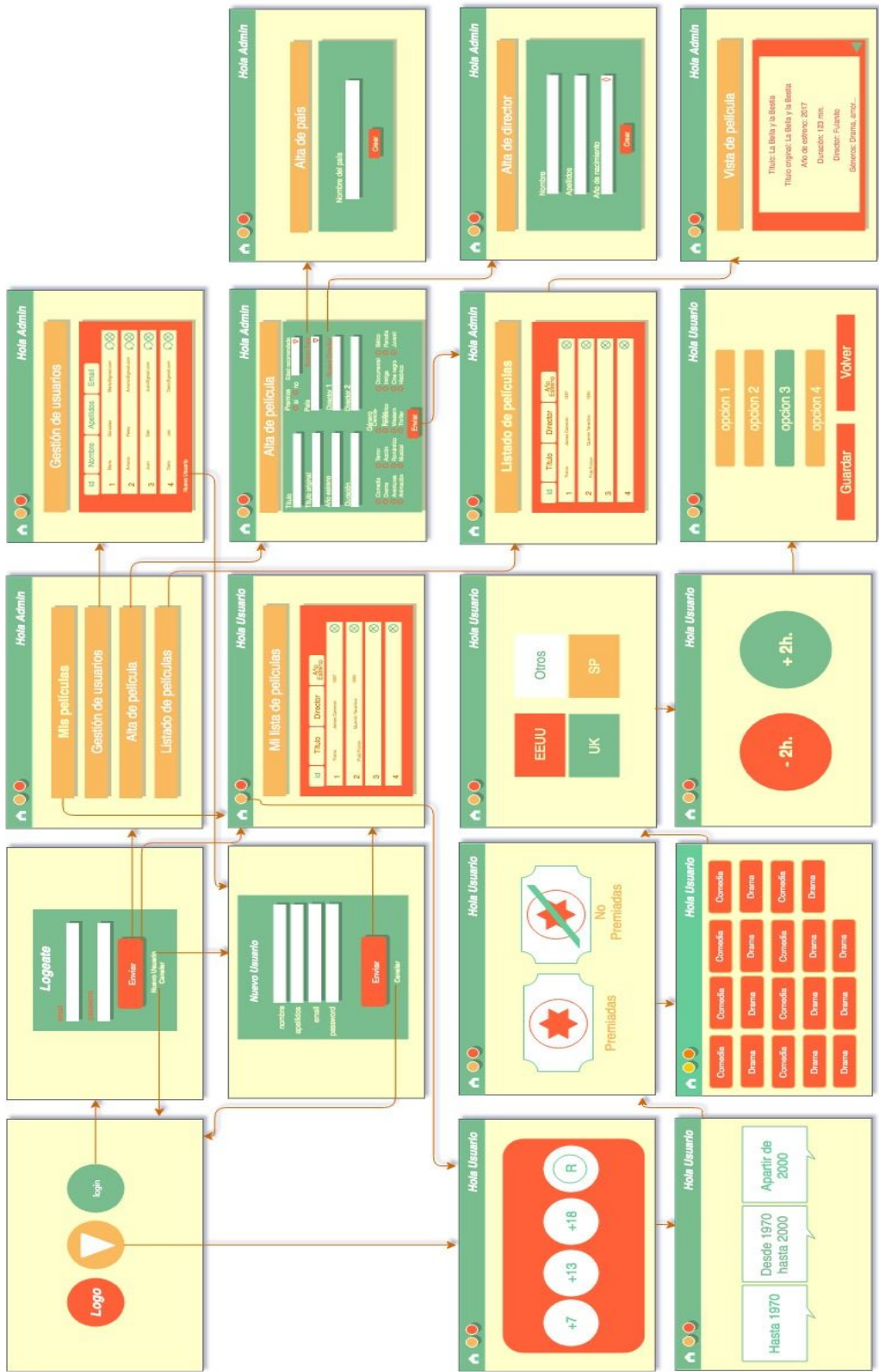
Model/Director

```

SELECT * FROM Directores WHERE id_director = :id_director;
SELECT * FROM Directores;
SELECT * FROM Directores WHERE id_director=:id_director;
SELECT * FROM Directores WHERE nombre=:nombre and apellidos=:apellidos;
SELECT * FROM Directores WHERE nombre=:nombre;
SELECT * FROM Directores WHERE apellidos=:apellidos;
INSERT into Directores(nombre, apellidos, año_nacimiento)
values(:nombre, :apellidos, :año_nacimiento);

```

ANEXO III



ANEXO IV

	FOLDERS	
<ul style="list-style-type: none"> ▼ cinecito <ul style="list-style-type: none"> ▼ Documentacion <ul style="list-style-type: none"> <> documentacion.md ▼ app <ul style="list-style-type: none"> ▼ controllers <ul style="list-style-type: none"> DirectorController.php FilterController.php HomeController.php ListaController.php LoginController.php PaisController.php PeliculaController.php RegisterController.php UserController.php ▼ models <ul style="list-style-type: none"> Director.php Edad.php Filter.php Genero.php Lista.php Pais.php Pelicula.php User.php ▼ views ▼ core ▼ public <ul style="list-style-type: none"> .gitignore .htaccess start.php 	<ul style="list-style-type: none"> ▼ cinecito <ul style="list-style-type: none"> ► Documentacion ▼ app <ul style="list-style-type: none"> ► controllers ► models ▼ views <ul style="list-style-type: none"> ▼ filter <ul style="list-style-type: none"> filtroanio.php filtrouracion.php filtrogenero.php filtropais.php filtropremios.php index.php resultado.php ▼ lista <ul style="list-style-type: none"> index.php ▼ login <ul style="list-style-type: none"> index.php ▼ parts <ul style="list-style-type: none"> footer.php head.php header.php scripts.php ▼ pelicula <ul style="list-style-type: none"> altaDirector.php altaPais.php altaPelicula.php index.php insetaGeneros.php show.php ► register ► user ► userAdmin home.php 	<ul style="list-style-type: none"> ▼ cinecito <ul style="list-style-type: none"> ► Documentacion ▼ app <ul style="list-style-type: none"> ► controllers ► models ▼ views <ul style="list-style-type: none"> ► filter ► lista ► login ► parts ► pelicula ▼ register <ul style="list-style-type: none"> index.php ▼ user <ul style="list-style-type: none"> index.php ▼ userAdmin <ul style="list-style-type: none"> gestion.php index.php home.php ▼ core <ul style="list-style-type: none"> App.php Model.php ▼ public <ul style="list-style-type: none"> ▼ css <ul style="list-style-type: none"> /* estilos.css ▼ img <ul style="list-style-type: none"> filtro.png filtro2.png inicio.png logo.png .htaccess index.php .gitignore .htaccess start.php