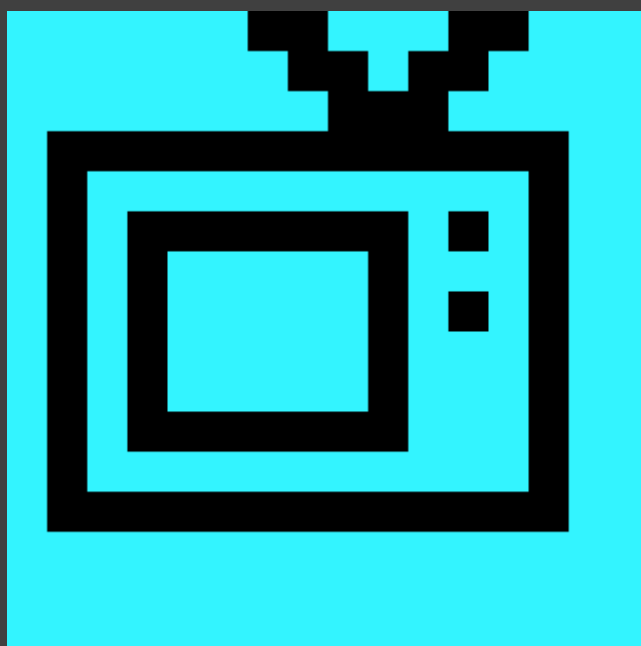


TRABAJO FIN DE GRADO EN DESARROLLO DE APLICACIONES WEB



TO WATCH LIST

AUTOR: JAIME SÁNCHEZ-VILLACAÑAS LUCAS

CURSO:2020-2021

Contenido	
Introducción	3
Motivación	3
Especificaciones. Requisitos, alcance y limitaciones del proyecto	4
Requisitos generales.	4
Requisitos funcionales	4
Alcance y limitaciones.	4
Herramientas y recursos	5
Herramientas.	5
Recursos	5
Planificación, desarrollo, metodología y códigos de estilo	6
Planificación, desarrollo y metodología	6
Códigos de estilo	7
Procedimientos de instalación y prueba.	9
Arquitectura del sistema y modelo de datos	11
Mejoras futuras	12

Introducción

ToWatchList es una aplicación web full stack desarrollada con las tecnologías PHP nativo, Angular JS y MySQL, creada para mantener al día todas las películas y series que vemos a diario. Con ToWatchList tendremos acceso a un catálogo de películas y series a nuestra disposición tanto para encontrar aquellos títulos que hemos visto, tenemos pendientes o que estamos siguiendo en este mismo momento.

Para realizar esta actividad, gracias a la aplicación, podremos crear listas donde podremos mantener guardados, ordenados y con toda la información a manos sobre las series y películas que queramos. Con la información fundamental que se necesita saber sobre una película o serie en un par de clics, respondiendo a las típicas preguntas de ¿En qué plataforma se puede ver este título?, ¿Quién es el director/autor de este título? o incluso ¿Cómo se llamaba la última serie/película que vi?

ToWatchList va dirigido a todas aquellas personas cinéfilas y “seriefilas” que a lo largo del año ven una gran cantidad de títulos y que no tienen una aplicación sencilla y fácil de utilizar para mantener un listado de dichos títulos. Además, de que es una gran herramienta para hacer recomendaciones a otras personas/amigos que no ven tantas películas o series al año, pero que en un momento puntual quieren disfrutar de un buen título.

Motivación

La motivación para realizar esta aplicación web vino de la necesidad de tener acceso a una lista donde pudiera consultar las últimas series y películas que había visto en el último año. Debido a mi naturaleza despistada y poco organizada, siempre acababa con las mismas dudas. Nunca recordaba con certeza cual era aquella película que vi hace un par de días o en que plataforma la había visto.

Debido al aumento de tiempo libre que la cuarentena trajo, con ella también vino el aumento del número de series y películas que vi durante ese año. Tras los largos meses de cuarentena y haber visto un sinfín de nuevos títulos acabe llegando a una conclusión; no sabía ni el número ni los nombres exactos de todas las películas y series que había visualizado.

Tras varios intentos de organizar y mantener un listado de todo lo que veía con programas como Excel o simples bloc de notas, todos sin excepción, acabaron siendo muy tediosos y difíciles de llevar al día y acabé fracasando en organizarme.

Por lo cual, de ahí surgió la idea de crear mi propia aplicación que resolviera todos los problemas previos que había tenido anteriormente y así resolver de una manera sencilla y rápida todos ellos.

Especificaciones. Requisitos, alcance y limitaciones del proyecto.

Requisitos generales.

Los requisitos generales del trabajo era crear una aplicación web funcional con una parte servidor o backend que fuese capaz de resolver las peticiones generadas por la parte cliente o frontend y una base de datos relacional que fuese capaz de almacenar toda la información necesaria para el uso y funcionamiento de la aplicación web.

Requisitos funcionales.

Las funcionalidades que ofrece la aplicación web son:

- Un sistema de usuarios para registrarse y autenticarse en la aplicación con la opción de crear su propia lista personal.
- Mostrar información de un catálogo de series y películas almacenados en la base de datos.
- Mostrar una serie de listas, que por defecto que la aplicación trae, como por ejemplo “Películas mejor valoradas”, “Títulos en Netflix” o “Títulos en HBO”.
- Ofrece un buscador donde el usuario podrá buscar los títulos mediante una búsqueda por plataformas de visualización, títulos, títulos originales, o incluso por actores, directores y autores.
- La posibilidad de crear una lista con tus series y películas de tu elección, pudiéndolas guardarlas como “vistas”, “pendientes” y en el caso de series “siguiendo”, además de poder también filtrarlas por dichos estados.

Alcance y limitaciones.

El desarrollo del proyecto pretende alcanzar una aplicación web full stack capaz de visualizar información sobre títulos de series y películas, mostrar una colección de listas de títulos creadas previamente, la posibilidad de crear tu propia lista de títulos vistos, pendientes y en seguimiento y que contenga un sistema de autenticación mediante usuarios.

Las limitaciones que se ha encontrado a la hora de la realización de este trabajo han sido principalmente con los datos necesarios para crear una base de datos, lo suficiente amplia y rica en contenido relacionado con cada uno de los títulos.

En un primer caso, con la utilización de la API TMDb, el problema del contenido y riqueza de cada uno de los títulos se solucionaba hasta cierto punto. De cada título se podía extraer una gran cantidad de información como los actores, directores, compañías involucradas ... a cambio de tener menos control sobre los datos.

El segundo acercamiento a solucionar este problema fue crear, en una pequeña escala, una base de datos relacional con un diseño relacional parecido al que tiene IMDB. Con esta solución me daba un control total sobre los datos que guardaba en la base de datos, pero sacrificando toda la información extra que me daba la API.

Con este acercamiento la principal limitación se convirtió en el enorme esfuerzo que suponía rellenar la base de datos a mano, recopilando la información de páginas como IMDB, FilmAfinity, etc.

Herramientas y recursos

Herramientas.

Para la creación de esta aplicación web full stack los programas utilizados fueron Visual Studio Code, Google Web Developers Tools, NetBeans, phpmyadmin, Postman y GitHub.

Para la parte del frontend las herramientas utilizadas fueron Visual Studio Code, el IDE elegido para programar en Angular y las herramientas de desarrollo web de Google, las cuales me sirvieron para depurar el código de la parte cliente, ver los posibles errores en consola y observar las llamadas que se hacían desde la página al backend.

En la parte del backend, el IDE escogido fue NetBeans para programar la parte de PHP, phpMyAdmin para crear la base de datos, introducir y guardar los datos; y Postman para ayudar a visualizar los resultados que se obtenían en cada una de las llamadas al servidor, sin la necesidad de tener una vista que mostrará los datos. Y por último para el control de versiones de la aplicación se ha utilizado GitHub.

Recursos

Los recursos utilizados para el desarrollo del proyecto han sido páginas como Stack Overflow, la documentación oficial de Bootstrap, Angular, PHP y MySQL para la resolución de dudas relacionadas con el código. Páginas como Reddit, Pinterest, Coolors y Google Imágenes para encontrar inspiración en el diseño, desarrollo, metodología y códigos de estilos para la aplicación. Y IMDB, FilmAfinity o TMDb para recaudar toda la información necesaria para la creación de la base de datos.

Planificación, desarrollo, metodología y códigos de estilo

Planificación, desarrollo y metodología

El primer paso tanto para el desarrollo como la planificación del proyecto fue la idea. Las primeras jornadas de trabajo se centraron en encontrar una idea con la que estuviese cómodo y me diera una buena motivación para completar el trabajo.

El siguiente paso, y con el una gran parte de las primeras jornadas de trabajo fue el diseño de la aplicación. El diseño tenía que ser “mobile friendly” además de ser “responsive”. Con estos requisitos, el segundo paso fue crear un diseño que se adaptará a estas pautas. Teniendo todo esto en cuenta, llegue a un diseño que era sencillo, limpio y no muy recargado que es pauta la cual he intentado seguir a lo largo del desarrollo del proyecto.

Con la idea y el diseño terminados, la estructura de datos, los patrones y la arquitectura fueron lo siguiente en la lista para desarrollar y planificar. Para el frontend utilice el patrón modelo-vista vista-modelo (MVVM) que es el patrón que Angular utiliza para crear sus componentes. Este patrón de diseño sincroniza los datos entre la vista y el modelo-vista de manera totalmente dependiente, es decir, en la vista se puede modificar el modelo y en el modelo podemos modificar la vista.

En cuanto al Backend, el patrón utilizado es el patrón modelo controlador (MC). El patrón divide la aplicación en partes que están interconectadas, separando las partes internas de la representación de los datos, estos son mandados al frontend, el cuál será el encargado de mostrar la información.

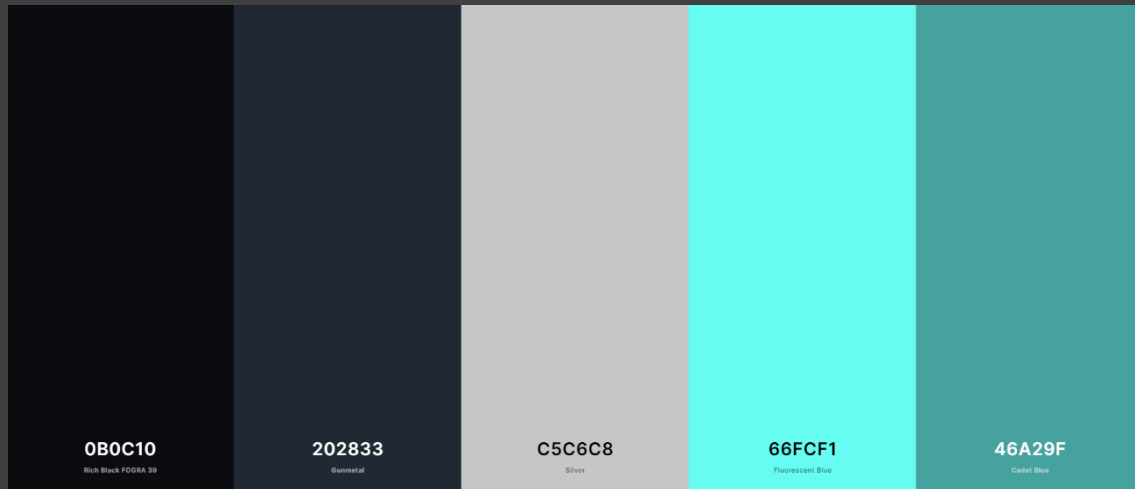
Una vez con el diseño y los patrones elegidos, dio paso al desarrollo del código fuente de la aplicación. Para su desarrollo he seguido una metodología en espiral, es decir, dividí toda la carga del proyecto en secciones más pequeñas para proporcionar una mayor facilidad a la hora de solucionar los problemas que fueron apareciendo.

La primera división fue dividir las tareas que tenía realizar en backend y frontend. Dentro de cada una de ellas, a su vez, se subdividían en subsecciones tales como gestión de usuarios, gestión de títulos y gestión de listas. El proceso de desarrollo se basaba en crear cada uno de esos segmentos y comprobar su correcto funcionamiento antes de introducir cualquier otro elemento en la aplicación.

Por último, y con ello la finalización del proyecto fue la creación de la documentación, así como la tarea de organizar lo que había hecho a lo largo de estos meses y recopilarlos en un documento de texto que resumiera todo el proceso.

Códigos de estilo

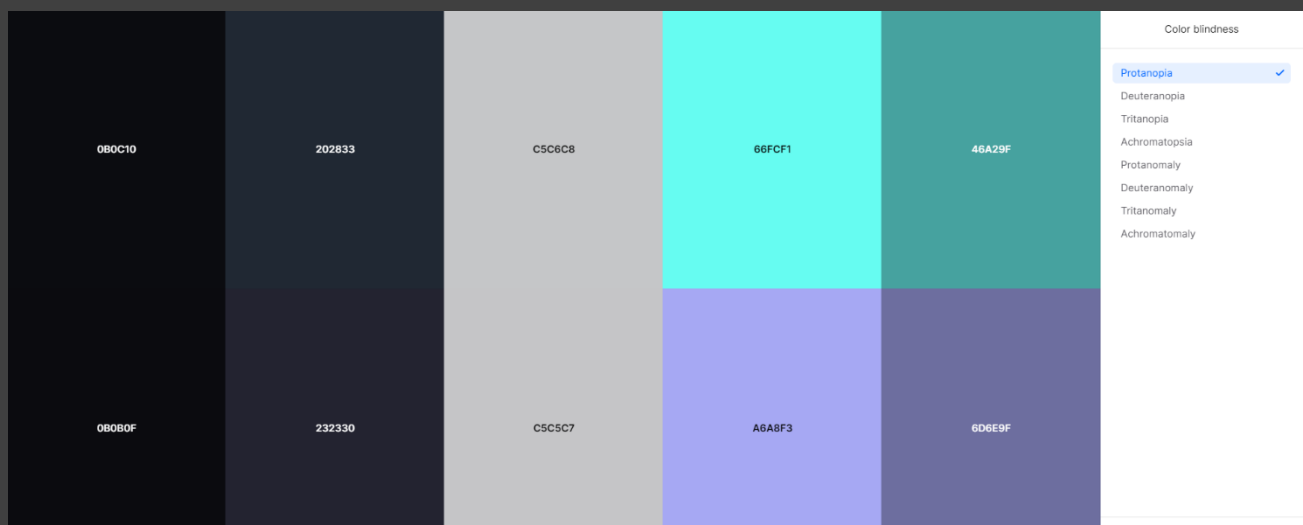
La paleta de colores principal (<https://coolors.co/0b0c10-202833-c5c6c8-66fcf1-46a29f>)



Estos son los colores de la aplicación ToWatchList, los cuales se mantienen presentes a lo largo de todas las vistas. Dichos colores están descritos con su color hexadecimal al igual que sus nombres. La mayor parte del texto principal irá en tono blanco.

Un plus a esta paleta de colores es que debido a la elección de colores oscuros casi no son afectados por el conjunto de daltonismos que existen como se puede ver en los siguientes ejemplos.

Protanopia:



Deuteranopia:



0B0C10	202833	C5C6C8	66FCF1	46A29F	<div>Color blindness</div> <div><div>Protanopia</div><div>Deuteranopia ✓</div><div>Tritanopia</div><div>Achromatopsia</div><div>Protanomaly</div><div>Deuteranomaly</div><div>Tritanomaly</div><div>Achromatomaly</div></div>
0B080E	23222F	C5C5C7	9E93F4	68619F	

Acromatopsia:

0B0C10	202833	C5C6C8	66FCF1	46A29F	<div>Color blindness</div> <div><div>Protanopia</div><div>Deuteranopia</div><div>Tritanopia</div><div>Achromatopsia ✓</div><div>Protanomaly</div><div>Deuteranomaly</div><div>Tritanomaly</div><div>Achromatomaly</div></div>
0C0C0C	262626	C5C5C5	CD0CD0	868686	

Procedimientos de instalación y prueba.

Para instalar el proyecto, primero tendremos que tener XAMPP instalado en nuestro pc y tener activado el servicio de Apache y MySQL.

	Apache	6928 9904	80, 443	Stop
	MySQL	19196	3306	Stop

Ahora abriremos una terminal en la ruta C:\xampp\htdocs.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL





PS C:\xampp\htdocs> 
```

<https://i.gyazo.com/5b0691fd22950cb9d20b35c133327649.mp4>

En esta terminal ejecutaremos el comando **git clone** seguido del link al repositorio de GitHub donde se encuentra alojado el proyecto. ([repositorio](#))

```
PS C:\xampp\htdocs> https://github.com/jaimesvlucas/repositorioSanchez.git
```

Una vez se haya ejecutado correctamente se nos creará la siguiente estructura de carpetas.

Este equipo > Disco local (C:) > xampp > htdocs > repositorioSanchez >		
Nombre	Fecha de modificación	
 backendSanchez	04/06/2021 22:21	C
 frontendSanchez	06/06/2021 19:50	C
 README.md	05/06/2021 19:09	A
 scriptSanchez.sql	04/06/2021 23:41	A

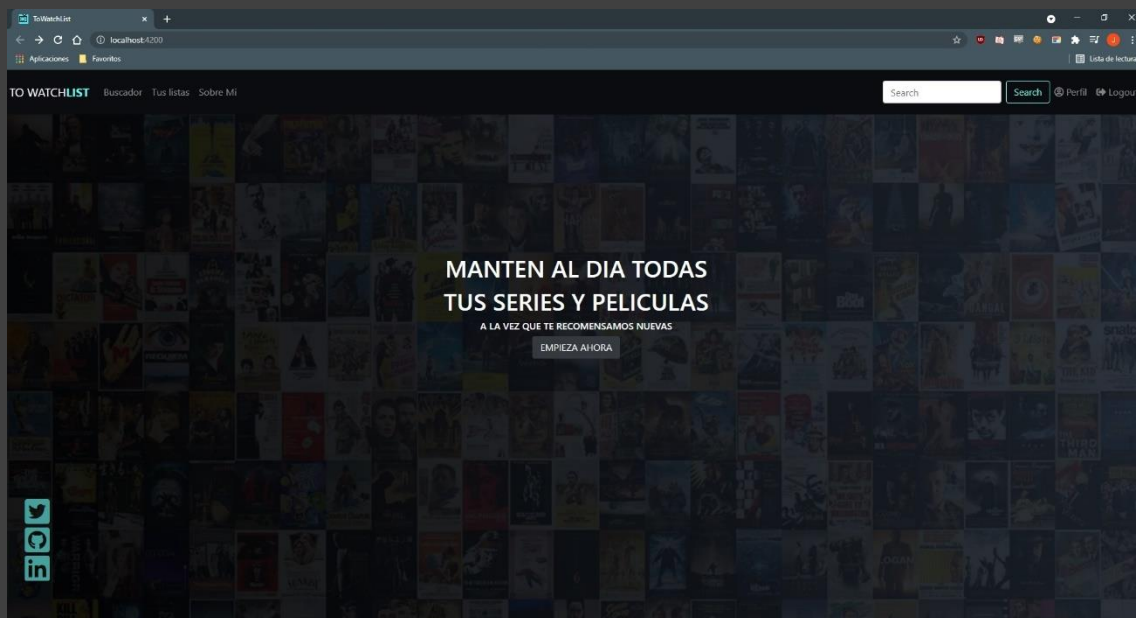
Una vez creada dicha estructura, abriremos phpMyAdmin e importaremos el scriptSanchez para crear la base de datos que utilizará nuestra aplicación. El script generará la base de datos con toda la información, tablas y relaciones.

	Table	Action	Rows	Type	Collation	Size	Overhead
	actores	★ Browse Structure Search Insert Empty Drop	150	InnoDB	utf8_spanish2_ci	32.0 KiB	-
	creadores	★ Browse Structure Search Insert Empty Drop	33	InnoDB	utf8_spanish2_ci	48.0 KiB	-
	directores	★ Browse Structure Search Insert Empty Drop	26	InnoDB	utf8_spanish2_ci	32.0 KiB	-
	escritores	★ Browse Structure Search Insert Empty Drop	50	InnoDB	utf8_spanish2_ci	32.0 KiB	-
	generos	★ Browse Structure Search Insert Empty Drop	127	InnoDB	utf8_spanish2_ci	32.0 KiB	-
	generos_des	★ Browse Structure Search Insert Empty Drop	18	InnoDB	utf8_spanish2_ci	16.0 KiB	-
	listas_usuarios	★ Browse Structure Search Insert Empty Drop	28	InnoDB	utf8_spanish2_ci	48.0 KiB	-
	nombres	★ Browse Structure Search Insert Empty Drop	214	InnoDB	utf8_spanish2_ci	16.0 KiB	-
	nombres_imagenes	★ Browse Structure Search Insert Empty Drop	214	InnoDB	utf8_spanish2_ci	80.0 KiB	-
	notas	★ Browse Structure Search Insert Empty Drop	50	InnoDB	utf8_spanish2_ci	16.0 KiB	-
	temporadas	★ Browse Structure Search Insert Empty Drop	25	InnoDB	utf8_spanish2_ci	48.0 KiB	-
	titulos	★ Browse Structure Search Insert Empty Drop	50	InnoDB	utf8_spanish2_ci	48.0 KiB	-
	titulos_imagenes	★ Browse Structure Search Insert Empty Drop	50	InnoDB	utf8_spanish2_ci	32.0 KiB	-
	users	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_spanish2_ci	16.0 KiB	-
	14 tables	Sum	1,036	InnoDB	utf8_spanish2_ci	496.0 KiB	0 B

Volviendo a la terminal que abrimos previamente usaremos el comando **cd frontend** + tab para que autocomplete el nombre de la carpeta he entraremos en la carpeta de frontendSanchez.

```
PS C:\xampp\htdocs\repositorioSanchez> cd .\frontendSanchez\
PS C:\xampp\htdocs\repositorioSanchez\frontendSanchez>
```

Ahora con el comando **npm install** instalaremos todas las dependencias que nuestro proyecto necesita y una vez termine dicho comando usaremos **npm start**. Una vez se acabe de ejecutar el comando anterior, podremos acceder a la aplicación introduciendo la siguiente dirección en nuestro navegador: <http://localhost:4200/>. Una vez dentro nos aparecerá la pantalla principal de la aplicación.



En este punto puedes crear un nuevo usuario o utilizar el usuario con las siguientes credenciales:

Correo: jaime@gmail.com

Contraseña: 1234

Arquitectura del sistema y modelo de datos

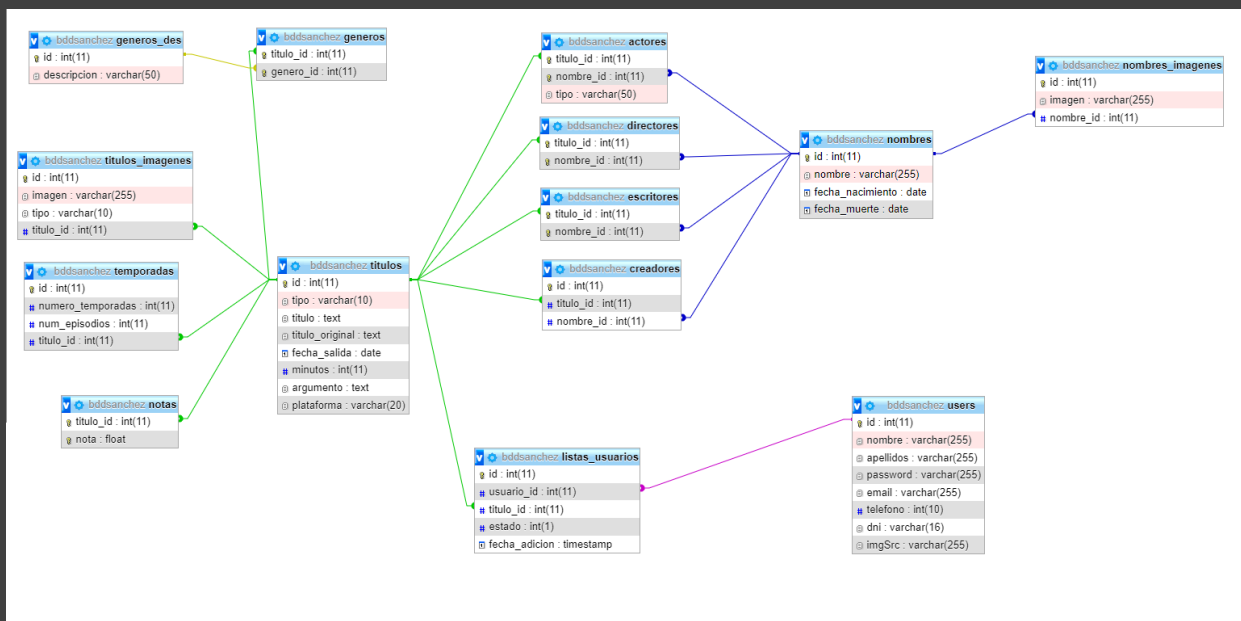
La arquitectura que sigue el frontend se compone de una serie de módulos que componen la aplicación. Existe un grupo de módulos agrupados en la carpeta componentes. Cada uno de estos módulos desempeña un servicio específico en la página.

El componente auth a su vez reúne todos los componentes relacionados con los usuarios. También componentes como buscador o listas usuarios agrupan los archivos que hacen posibles dichos servicios.

También el frontend dispone de módulos externos como pueden ser las clases, pipes o el componente servicios que guarda los archivos que gestionan las llamadas al backend. Por último, módulos como appModules o appRouting controlan las dependencias que utiliza la aplicación y el sistema de enrutamiento.

En cuanto al backend la arquitectura que existe tiene dos principales grupos de archivos: los controladores y los modelos. Los controladores que responden a las llamadas hechas por el cliente e invocan al modelo cuando se hace alguna petición de información. Y los modelos son los que gestionan toda la información y hacen las llamadas a la base de datos.

En cuanto a la base de datos utilizo un modelo relacional que consiste en el siguiente esquema:



Mejoras futuras

La principal mejora y la que creo que es la más importante es conseguir una base de datos con información real y que se actualice día. Es una de las principales cosas a mejorar ya que en este momento solo consta de una basa de datos con 50 series y películas en su conjunto.

Otra modificaciones o mejoras futuras podrían ser:

- Rediseño de las páginas de información de los títulos
- Rediseño del perfil de usuario
- La introducción de etiquetas personalizadas para las películas
- Introducción de un sistema de recomendaciones personalizadas