# Diapo 1

Buenas tardes, soy Diego Viñals y mi proyecto fin de grado va sobre la creación de una aplicación móvil que permita gestionar las series de televisión para grupos familiares o grupos de amigos.

# Diapo 2: Índice

Haré una breve introducción al problema que se pretende resolver, así como la solución la motivación que me ha llevado a realizar este proyecto.

Explicare la metodología que se quiso seguir y sus pasos.

Comentare los objetivos propuestos al principio del proyecto

El Backend y frontend de la aplicación

Como funciona el proceso de despliegue de la aplicación

Una pequeña demo y las conclusiones

# Diapo 3: Introducción

El problema que se presenta es el problema de la gestión de series, en un grupo familiar se puede dar el caso de que los hijos han visto un capítulo más que los padres, y en este caso los padres no sabrían porque capitulo van los hijos, y cuando se quiere ver la serie con la familia entera nos damos cuenta de que no se puede ver porque se va desfasado en la familia.

La solución que se propone a este problema es la creación de una aplicación móvil en la que los distintos miembros de la familia vayan poniendo por donde van en las distintas series, para así poder ver de una manera sencilla como va cada uno.

La motivación surge porque este problema descrito nos pasa en mi familia. Nosotros desde pequeños siempre hemos visto las series todos juntos, y es algo que es tradición en nuestra familia, pues hace unos años se fueron a vivir a estados unidos y cada cierto tiempo venían a España y seguíamos viendo las series juntos pero nos pasaba que nosotros no sabíamos que capítulos habían visto y cuales no, por lo que había cierto desorden y era casi imposible coordinarse.

Con esta aplicación este problema quería resuelto, ya que cuando mis padres ven un capítulo de una serie, lo marcan, y nosotros ya sabemos que tenemos uno pendiente por ver.

# Diapo 4: Metodología

La metodología escogida en este proyecto es la metodología en cascada, ya que solo yo estoy haciendo el proyecto.

Esta metodología coincide con los paquetes de trabajo realizados:

* Análisis de Requisitos: Se envió un formulario y con esa información se detallaron los requisitos funcionales y no funcionales.
* Diseño del sistema: Con estos requisitos se creó en Figma un diseño preliminar que se debía de seguir en la implementación del frontend
* Implementación: Realización del frontend y backend en React native y expo (para poder ver el código en tiempo real en simulador y luego hacer los builds de Android e iOS)
* Pruebas y calidad: Se envió un enlace a ciertas personas junto con un formulario en el que valoraban la aplicación, enviaban comentarios, etc.
* Despliegue: Se subió la aplicación a Google y Apple.

# Diapo 5: Objetivos

El objetivo principal es facilitar el seguimiento conjunto de series de televisión favoritas para grupos de usuarios como familias o amigos. Con este objetivo cumplido se resolvería el problema propuesto.

Los objetivos específicos serian:

1. Facilitar la coordinación de visualización en grupos.
2. Mejorar la toma de decisiones colectivas sobre qué ver
3. Enriquecer la experiencia compartida de visualización mediante la creación de un espacio para comentarios
4. Proporcionar una interfaz intuitiva y accesible
5. Estadísticas de visualización

# Diapo 6: Backend

# Lo primero que hay que mencionar es que toda la información relacionada con las series de televisión se ha obtenido de una API (The Movie Database), que incluye datos como la descripción de la serie, pósters, temporadas, etc. El backend se ha implementado en un servidor local, al cual la aplicación accede de manera global. Para ello, hemos utilizado Docker, lo que nos permite crear contenedores que necesitamos y facilita su comunicación entre ellos. Se creó un contenedor de MariaDB que almacena la base de datos, la cual se modifica mediante la API que desarrollé (que es otro contenedor). También hemos utilizado Cloudflare para acceder al servidor mediante un DNS y Traefik, que gestiona el enrutamiento hacia el contenedor deseado.

# Diapo 7: API

# La API que hemos creado contiene todos los endpoints necesarios para acceder a la información de la base de datos mediante los métodos GET, POST, DELETE y PUT. Esta API es la única forma de conexión con la base de datos, lo que significa que solo se puede modificar la base de datos a través de esta API o directamente mediante el contenedor de MariaDB. Ningún usuario externo tiene acceso directo a la base de datos.

# Diapo 8: MariaDB

# Este contenedor funciona como la base de datos propiamente dicha. Alberga todas las tablas necesarias para almacenar la información, tales como las tablas de usuarios, series, grupos y usuario\_grupo (que indica a qué grupos pertenecen los usuarios), entre otras. Como se mencionó anteriormente, toda la información de las series se obtiene de una API externa. En nuestra tabla de Series, solo almacenamos ciertos datos como el ID para facilitar una búsqueda más rápida de la información.

# Diapo 9: MariaDB

Aquí podemos ver cómo queda las relaciones entre las tablas, como el id de usuario se copia en las demás tablas etc.

Explicar cada tabla

# Diapo 10: Cloudfalre y Traefik

# Aquí tenemos un diagrama que nos ayuda a entender cómo funciona el tráfico de la petición desde la app móvil hasta la base de datos y cómo vuelve. Veamos cómo sería este proceso al crear un usuario. Primero, la app recoge los datos y los envía al DNS. Cloudflare, actuando como DNS, crea un túnel entre la aplicación y el servidor, y redirige los datos al contenedor de Traefik. Este, al detectar que se trata de una petición a la API, pasa los datos y el endpoint al contenedor de la API, en este caso al endpoint de creación, donde se ejecuta la sentencia SQL 'CREATE usuario' con los datos proporcionados. Si la API necesitara enviar información de vuelta a la aplicación, el proceso sería el mismo pero en dirección inversa, como se muestra en el diagrama.

# Diapo 11: Frontend

# El frontend se ha desarrollado en React Native, lo que permite compilar la aplicación tanto para Android como para iPhone, convirtiéndola en una herramienta accesible para todo tipo de usuarios. Se ha utilizado junto con Expo, que me permite ver en el simulador cómo queda la app y cómo funciona en tiempo real. Luego, Expo se encarga de tomar el código de React y compilarlo para la plataforma deseada. El código se compone de funcionalidades, como por ejemplo, mostrar series, marcar como visto, enviar comentarios, etc. Los componentes se encargan de visualizar la aplicación; son similares a HTML, pero no exactamente iguales. Por último, los estilos, que determinan cómo se verá la aplicación, son similares a CSS pero con sintaxis propia. A continuación, veremos algunos ejemplos de pantallas de la aplicación.

# Diapo 12: Frontend ejemplos

# Aquí tenemos unos ejemplos de cómo se muestra una serie que se está viendo, incluyendo su descripción y quién va por qué capítulo. Podemos ver que Diego va por el capítulo XX. También podemos ver el chat de una serie, la pantalla de inicio con las series en el grupo de Diego, con los botones, etc. Aquí vemos cómo se detalla una temporada, con botones de 'visto' en verde, lo que indica que ya se ha visto el capítulo, y de 'marcar como visto' en azul, para los que no se han visto. Se pueden marcar y desmarcar capítulos como vistos y no vistos, por si hubiera algún error al marcar. Más adelante en la demo, veremos el resto de pantallas y cómo se comportan.

# Diapo 13: Despliegue

# Para el despliegue, se ha optado por lanzar tanto en Google Play como en Apple Store. Al principio, pensaba que Google Play sería más sencillo, pero resultó ser lo contrario. Para esto, ha sido necesario crear cuentas en ambas tiendas y rellenar todos los datos que solicitan para poder realizar el despliegue, tales como el nombre, capturas de pantalla, descripción, versión, etc.

Ahora vamos a explicar el proceso de despliegue en amabas tiendas

# Diapo 14: Despliegue Android

# El método de despliegue de Android es el siguiente: se compila con Expo la aplicación en Android, y esto crea un archivo ABB. Este archivo contiene la versión y la aplicación. Luego, lo subimos a Google Play Console, donde añadimos el nombre y la descripción. Creamos una prueba interna a la que solo pueden acceder usuarios específicos, y con esto obtenemos el acceso a la prueba cerrada. Esta prueba es similar a la interna, pero la aplicación es revisada por los miembros de Google para comprobar que es estable y funciona correctamente, por lo que tardan unos días en otorgar acceso a la prueba cerrada. La prueba abierta es una etapa en la que, con el link de descarga, ya se puede descargar la aplicación; en la prueba cerrada, yo tengo que añadir el email del probador para que pueda descargar la aplicación. Una vez que se tenga a 20 probadores durante 15 días en prueba cerrada, Google otorga acceso a la prueba abierta y a la producción. Actualmente, la aplicación se encuentra en el estado de prueba cerrada.

# Diapo 15: Despliegue iOS

# El proceso es similar al despliegue en Android. Se compila el archivo en Apple, que devuelve un archivo .ipa.. Este se sube a Apple Store Connect, donde comienza la prueba interna. A esta prueba solo pueden acceder miembros del equipo de probadores, seleccionados manualmente por mí. Una vez terminada esta prueba, se pasa a la prueba cerrada, en la que Apple revisa la aplicación para asegurar su fiabilidad. A esta prueba solo se puede acceder mediante un enlace de invitación, pero cualquier persona con este enlace puede entrar, a diferencia de Google, que requiere un email. Una vez aceptada en la prueba cerrada, se completa toda la información necesaria y se sube al Apple Store. Este proceso tarda unos días ya que la aplicación es revisada de nuevo por el equipo de Apple. Actualmente, la aplicación está disponible en el Apple Store, de hecho, si la buscas, la puedes descargar como cualquier otra aplicación

# Diapo 16: Demo

Ahora veremos una demo de en iOS cómo funciona la aplicación. (dura 3 mins)

# Diapo 17: Conclusiones

Se llega a la conclusión con esta app de que el objetivo principal ha sido alcanzado, al igual que todos los objetivos específicos, excepto uno, ya que no se ha implementado la visualización de estadísticas. Como conclusión, se ha desarrollado una app que está desplegada actualmente en iOS y en prueba cerrada en Google, pero es cuestión de tiempo que esté subida en Google Play

Esto es todo muchas gracias