

# Requerimientos del Proyecto:

## API, Frontend y Base de Datos

### Servidor(API) 30hr

#### Módulo de Autenticación

- Implementar un sistema de autenticación que valide las credenciales de cada usuario de manera segura y eficiente. Tras la verificación exitosa, se generará un JWT (JSON Web Token) que permitirá el acceso a las rutas y funcionalidades protegidas de la aplicación.
- Considerar la implementación de refresh tokens para extender la sesión del usuario de manera segura sin requerir un nuevo inicio de sesión, mejorando tanto la experiencia del usuario como la seguridad del sistema.

#### Controladores y Servicios

1. **Autenticación:**
  - Desarrollar controladores dedicados a manejar el registro, inicio de sesión, y la validación continua de los usuarios mediante JWT.
  - Implementar servicios para gestionar la generación, almacenamiento y verificación de tokens de manera centralizada, garantizando la coherencia y seguridad del sistema.
2. **Gestión de Información:**
  - Crear controladores para manejar la creación, lectura, actualización, y eliminación (CRUD) de la información relacionada con actores, personajes, y otros datos clave.
  - Desarrollar servicios para procesar la lógica de negocio asociada a la gestión de la información, asegurando que las operaciones sean eficientes y cumplan con las reglas de negocio definidas.
3. **Gestión de Roles:**
  - Implementar un sistema de roles que permita definir permisos específicos para diferentes tipos de usuarios (por ejemplo, administradores, usuarios regulares).
  - Crear controladores y servicios para asignar, modificar, y verificar los roles de los usuarios, asegurando que los permisos sean actualizables y escalables según las necesidades futuras de la aplicación.
4. **Autorización de Permisos para Rutas:**
  - Desarrollar middleware que asegure que cada ruta de la API esté protegida según los roles y permisos asignados. Solo los usuarios con los roles adecuados podrán acceder a determinadas rutas o realizar acciones específicas.

- Implementar un sistema flexible que permita la actualización y modificación de permisos de manera dinámica, facilitando la adaptación a nuevos requerimientos sin necesidad de cambios significativos en el código base.

### **Auditoría y Monitoreo:**

- Integrar un sistema de auditoría que registre las acciones críticas (como cambios de roles o modificaciones de información) realizadas por los usuarios, proporcionando un historial detallado para mejorar la seguridad y trazabilidad.
- Implementar un sistema de logs para monitorizar el rendimiento de la API y capturar errores, facilitando la detección de problemas y su resolución oportuna.

## **Cliente(FRONT) 40hr**

### **Registro de Nuevos Actores**

- Tanto los usuarios como los administradores podrán agregar nuevos actores junto con los personajes correspondientes, facilitando la expansión continua de la base de datos de la aplicación.

### **Visualización de Información de Personaje**

- Los usuarios tendrán acceso a una vista detallada de la información de los personajes, permitiéndoles explorar de manera más profunda y organizada los detalles de cada personaje.

### **Registro de Actores Leídos o Investigados**

- Los usuarios podrán marcar actores como "leídos" o "investigados", manteniendo un registro personal para evitar duplicidades y mejorar la organización de su exploración.

### **Autenticación de Usuario**

- La aplicación requerirá un inicio de sesión y registro para que la información personal de cada usuario esté protegida, asegurando que cada cuenta sea única y segura.

### **Panel de Administración de Usuarios**

- Los administradores tendrán la capacidad de supervisar a los usuarios registrados, con la opción de eliminar cuentas o modificar permisos en caso de problemas o situaciones específicas.

### **Panel de Administración de Información de Personajes**

- Los administradores podrán actualizar o modificar la información de personajes y actores, garantizando la veracidad y el cumplimiento de las normas de seguridad de la aplicación.

### **Actualización de Información**

- Los usuarios tendrán la posibilidad de actualizar o mejorar la información existente sobre un actor o personaje, así como eliminarla si es necesario, contribuyendo activamente al mantenimiento y la precisión de la base de datos.

### **Búsqueda de Personajes**

- Se proporcionará una función de búsqueda que permitirá a los usuarios localizar personajes específicos o buscar personajes principales asociados a una película o serie, facilitando la navegación y el acceso a la información.

### **Panel de Actores Creados**

- Los usuarios dispondrán de un panel donde podrán visualizar los actores que han creado, con la opción de actualizar o editar la información de estos según sea necesario.

### **Panel de Actores Leídos**

- Los usuarios tendrán acceso a un panel que mostrará una lista de los actores que han marcado como leídos, permitiendo un fácil seguimiento de su progreso y exploración dentro de la aplicación.

## **Base de datos(MONGODB) 5hr**

Es fundamental diseñar e implementar una base de datos que almacene eficientemente toda la información relevante de la aplicación. MongoDB ha sido seleccionada como la tecnología ideal para este proyecto debido a su capacidad de escalar de manera eficiente y manejar grandes volúmenes de datos. La estructura de la base de datos debe ser flexible y permitir la rápida incorporación de nuevos campos y colecciones, asegurando que pueda crecer junto con las necesidades futuras de la aplicación.

**Modelar Colecciones y Documentos:** Definir las colecciones clave y estructurar los documentos dentro de estas colecciones para reflejar las relaciones y la naturaleza jerárquica de los datos.

## **Mínimo entregable. Base de datos 25hr**

Base de datos en mongoDB

## **servidor**

- rutas para autenticación de usuario
- rutas para manejo de la información de personaje (get, post, delete)

## **Cliente**

- visualización de información
- creación de nueva información
- login y registro.
- marcar actor como leído