

---

# **Especificación de requisitos de software**

**Para**

***The Hive***

**Versión 1.3 no aprobado**

**Preparado por Ana Meza, Alejandro Villarreal & Isabella  
Ordosgoitia**

**Universidad Tecnológica de Bolívar**

**17 de marzo de 2025**

## Tabla de contenido

<b>1. Introducción.....</b>	<b>1</b>
<b>2. Descripción general.....</b>	<b>2</b>
2.1 Características del producto.....	2
2.2 Clases de usuarios y características.....	2
2.3 Limitaciones de diseño y aplicación.....	3
2.4 Documentación del Usuario .....	3
2.5 Supuestos y dependencias.....	4
<b>3. Características del sistema.....</b>	<b>5</b>
3.1 Características de Sistema .....	6
3.1.1 Descripción y prioridad .....	6
3.1.2 Secuencias de estímulo/respuesta.....	6
3.1.3 Requisitos funcionales .....	7
<b>4. Requisitos de la interfaz .....</b>	<b>12</b>
4.1 Interfaces de Usuario.....	15
4.2 Interfaces de hardware.....	19
4.3 Interfaces de software .....	20
4.4 Interfaces de Comunicación .....	21
<b>5. Otros requisitos No funcionales .....</b>	<b>22</b>
5.1 Requerimientos de desempeño.....	22
5.2 Requisitos de seguridad.....	22
5.3 Atributos de calidad del software .....	23
<b>6. Otros Requerimientos .....</b>	<b>Error! Bookmark not defined.</b>
<b>7. Referencias .....</b>	<b>Error! Bookmark not defined.</b>

## Historial de revisiones

Nombre	Fecha	Motivo de los cambios	Versión
Alejandro Villareal	17 marzo 2025	Creación del documento. Descripción general del producto.	0.0
Isabella Ordosgoitia	18 marzo 2025	Continuación de Descripción general del producto	0.1
Ana Meza	18 marzo 2025	Establecer y describir características del sistema. Secuencia y requisitos de la autenticación y gestión de usuario.	0.2
Ana Meza	19 marzo 2025	Características del sistema: Secuencia y requisitos de la interacción entre usuarios, interacción usuario-contenido y de búsqueda y filtrado.	0.3
Ana Meza	20 marzo 2025	Características del sistema: Secuencia de la gestión de publicidad y del periódico.	0.4
Isabella Ordosgoitia	20 marzo 2025	Características del sistema: Secuencia y requisitos de gestión de contenido	0.5
Ana Meza	21 marzo 2025	Características del sistema: Requisitos de la gestión de publicidad y del periódico.	0.6
Alejandro Villarreal	14 abril 2025	Cambio de nombre de la aplicación	1.0

Alejandro Villarreal	22 abril 2025	Requisitos de la interfaz: usuario, hardware. Otros requisitos no funcionales. Apéndices A y B	1.1
Alejandro Villarreal	23 abril 2025	Requisitos de la interfaz: software, comunicación.	1.2
Alejandro Villarreal	23 abril 2025	Adicionar apéndice C	1.3

# 1. Introducción

La fragmentación de sistemas y datos en silos es un desafío crítico para esta empresa de medios, que opera múltiples plataformas sin una integración eficiente. Esta desconexión dificulta la analítica integral del cliente, limita la personalización de experiencias y retrasa la innovación al requerir cambios coordinados en diversas infraestructuras.

La siguiente Especificación de Requisitos de Software (SRS) para *The Hive* responde a la necesidad de una empresa de medios de unificar el intercambio de datos entre sus negocios de streaming, publicaciones digitales, publicidad y prensa; y el manejo de su oferta de contenido y suscripciones adaptadas a cada usuario.

El documento está basado bajo las directrices expuestas por el estándar IEEE: *Práctica recomendada del IEEE para Requisitos de software Especificaciones ANSI/IEEE 830, 1998*. En este, se describirá de forma a detallada las interfaces de usuario, de software, del hardware y comunicaciones, así como de los requerimientos del cliente, atributos del sistema entre otros.

## 2. Descripción general

### Perspectiva del producto

*The Hive* es una aplicación web capaz de integrar los datos entre los negocios de streaming, publicidad digital, publicaciones digitales, y prensa.

El software ofrecerá a los usuarios una experiencia integral para descubrir, gestionar y disfrutar del cine, combinando una interfaz segura con funciones avanzadas. Permitirá el registro con autenticación, recuperación de cuenta y un nombre de usuario único. Los usuarios podrán buscar películas, ver detalles, tráilers y soundtracks de Spotify si están disponibles, además de guardar títulos en su watchlist, marcarlos como vistos, calificarlos y llevar un historial de búsqueda. La seguridad estará garantizada con datos cifrados, y recibirán un resumen diario por correo con su actividad. Además, contará con un periódico digital con el top de películas mejor valoradas y nuevos estrenos, noticias sobre la industria, festivales y tendencias, creando así un ecosistema completo para los amantes del cine.

### 2.1 Características del producto

El *StreamHive* tendrá las siguientes características:

- **Gestión de usuario:** Los usuarios registrarse con un correo electrónico y una contraseña segura, iniciar sesión con autenticación, recuperar su cuenta en caso de pérdida y contar con un nombre de usuario único.
- **Gestión de contenido:** Los usuarios podrán buscar películas, acceder a información detallada, ver tráilers oficiales y escuchar el soundtrack disponible en Spotify, si la película lo tiene. También podrán guardar películas en su \*watchlist\*, marcarlas como vistas, calificarlas y llevar un historial de búsqueda para facilitar el acceso a sus intereses previos.
- **Gestión de publicidad:** La plataforma enviará a los usuarios un resumen diario por correo electrónico con su actividad en la plataforma, como nuevas películas guardadas, calificaciones recientes y recomendaciones personalizadas.
- **Gestión de prensa digital:** El periódico digital complementará la experiencia con una sección destacada de las películas mejor valoradas por los usuarios, así como una lista de los nuevos estrenos. Para los amantes del cine que buscan información más profunda, también se publicará todo lo nuevo en la industria: los mejores estrenos, tendencias cinematográficas, festivales, premios y otros eventos relevantes.

### 2.2 Clases de usuarios y características

- **Suscriptores:** Usuarios que tendrán acceso al contenido tanto de la plataforma web de streaming. Podrán registrarse, ser validados, comentar y calificar películas. Su conocimiento técnico será básico.
- **Editores:** Usuarios encargados del contenido del periódico digital. Podrán publicar artículos sobre diversos temas relacionados con el mundo del entretenimiento.

- **Críticos:** Usuarios con un mayor nivel de relevancia en la plataforma de streaming, quienes podrán tener una verificación de forma que sus comentarios y su actividad en la plataforma tengan mayor visibilidad.

## 2.3 Limitaciones de diseño y aplicación

El desarrollo del sistema presenta ciertas limitaciones que se deben tener en cuenta:

- **Políticas corporativas y normativas:** La plataforma debe cumplir con normativas de privacidad, regulaciones locales y políticas internas sobre gestión de contenido y moderación. Además, la publicidad debe garantizar transparencia en la recopilación y uso de datos, evitando prácticas invasivas.
- **Limitaciones de hardware:** El rendimiento del sistema estará influenciado por la infraestructura del servidor y la capacidad de procesamiento de los dispositivos utilizados por los usuarios.
- **Integración con otras aplicaciones:** Será necesario integrar el sistema de bases de datos externas y otros tipos de sistemas para la gestión de contenido, lo que conlleva la necesidad de compatibilidad con APIs específicas.
- **Tecnologías y herramientas:** El desarrollo utilizará una arquitectura basada en microservicios, con tecnologías en la nube para el almacenamiento y procesamiento de datos. Se emplearán frameworks modernos para la realización de la interfaz de usuario.
- **Seguridad:** Se implementará un mecanismo de autenticación para garantizar el acceso seguro a la plataforma, protegiendo la información de los usuarios. Los datos sensibles serán cifrados. Además, se aplicarán políticas de control de acceso según el tipo de usuario, asegurando que cada uno solo pueda visualizar o modificar la información correspondiente a su rol dentro del sistema.
- **Normas de programación:** Se seguirán estándares de desarrollo y buenas prácticas para asegurar la mantenibilidad y escalabilidad del software. Se implementará un proceso de Integración y Despliegue Continuo (CI/CD) para facilitar la implementación de nuevas funciones sin afectar la estabilidad del sistema.

## 2.4 Documentación del Usuario

Para asegurar una experiencia de usuario excepcional y facilitar la adaptación al sistema, se ofrecerá la siguiente documentación:

- **Manual de usuario:** Contará con instrucciones detalladas sobre la configuración, funciones principales y una resolución de problemas.
- **Guía rápida:** Tendrá un resumen breve y conciso de las funciones esenciales. Esta guía será accesible desde las plataformas.
- **Ayuda en línea:** Con respuestas a preguntas frecuentes y explicaciones paso a paso.
- **Tutoriales:** En formato video sobre el uso del sistema.

- **Documentación técnica:** Contendrá información sobre la arquitectura, APIs e integración con otros sistemas.

## 2.5 Supuestos y dependencias

Al desarrollar el sistema, se han considerado ciertos supuestos que podrían afectar los requisitos establecidos:

- **Compatibilidad con infraestructuras existentes:** Se espera que el entorno operativo, incluyendo servidores, bases de datos y redes, soporte la arquitectura propuesta sin requerir cambios sustanciales.
- **Disponibilidad de servicios de terceros:** Se asume que las APIs externas y servicios de terceros estarán operativas y disponibles para su integración sin modificaciones significativas.
- **Conectividad a Internet:** Se presume que los usuarios contarán con una conexión estable que les permita la sincronización de datos en tiempo real.
- **Disponibilidad de hardware:** Se espera que los dispositivos de los usuarios cumplan con los requisitos mínimos necesarios para ejecutar la aplicación sin inconvenientes.

### 3. Características del sistema

El sistema de *StreamHive* ha sido diseñado para ofrecer a los usuarios una experiencia completa y personalizada en la gestión y disfrute del cine. Esta plataforma integra diversas funcionalidades para garantizar una navegación intuitiva y segura, proporcionando herramientas como creación, autenticación, modificación y eliminación de cuentas, vista de la información de películas como tráileres, reseñas y calificación; búsqueda de películas por medio de filtros, almacenamiento de películas en “watchlists”, creación de listas de películas personalizadas, y la conexión con plataformas como Spotify para mostrar el “soundtrack”. Además, ofrece un periódico digital que hace uso de las estadísticas obtenidas por el usuario para hallar el ranking de películas del país en el que se encuentre la persona. Cada funcionalidad del sistema es desarrollada para optimizar la operatividad y minimizar los errores humanos al momento de la gestión, con el objetivo de crear el mejor ecosistema interactivo ideal para los amantes del cine.

#### 3.1 Autenticar Usuario

##### 3.1.1 Descripción y prioridad

El sistema debe permitir el ingreso y salida de usuarios a través de su correo electrónico o nombre de usuario y una contraseña para acceder a sus funciones.

Prioridad: Alta.

##### 3.1.2 Secuencias de estímulo/respuesta

El sistema permitirá iniciar de sesión al usuario por medio de su correo electrónico o nombre de usuario y su contraseña.

Si el identificador se encuentra dentro de la base de datos, pero la contraseña es incorrecta, el sistema notificará al usuario sobre esto por medio de un mensaje de error. El usuario podrá equivocarse un máximo de cinco veces al ingresar la contraseña, en caso de que este número de veces se sobrepase, el sistema bloqueará temporalmente su cuenta.

Si usuario olvidó su contraseña, el sistema permitirá su recuperación por medio de un link que será enviado a su correo electrónico.

En caso de que el campo de correo electrónico o nombre de usuario esté lleno y no se encuentren dentro de la base de datos, el sistema mostrará un mensaje de error al usuario notificándole sobre este. Asimismo, si no han sido rellenado alguno de los campos y el usuario desea ingresar, el sistema desplegará un mensaje de error.

El sistema permitirá al usuario cerrar sesión en cualquier momento en la que esta esté abierta.

##### 3.1.3 Requisitos funcionales

AUTH-1: El sistema debe permitir a un usuario registrado, iniciar sesión con su correo electrónico o nombre de usuario y contraseña.

AUTH-2: El sistema debe comprobar que el formato de cada campo de la información ingresada por el usuario al iniciar sesión sea correcto.

AUTH-3: El sistema debe validar que los campos a ingresar hayan sido llenados.



- AUTH-4: El sistema debe validar que la información de los campos coincida con la almacenada en la base de datos.
- AUTH-5: El sistema debe registrar la fecha y hora del último inicio de sesión realizado por el usuario.
- AUTH-6: El sistema debe permitir al usuario recuperar su contraseña por medio de un enlace de uso único y temporal enviado al correo electrónico registrado.
- AUTH-7: El sistema bloqueará temporalmente la cuenta del usuario, si este digita más de cinco veces una contraseña inválida.
- AUTH-8: El sistema debe ir mostrando la cantidad de intentos disponibles del usuario al equivocarse ingresando su contraseña.
- AUTH-8: El sistema debe permitir al usuario cerrar su sesión de manera segura.
- AUTH-9: El sistema debe desplegar un mensaje de error cuando el usuario intente ingresar, pero los campos se encuentren vacíos.
- AUTH-10: El sistema debe mostrar un mensaje de error al usuario en caso de que alguno de los campos ingresados no coincida con los datos de la base de datos.

## 3.2 Gestionar Usuario

### 3.2.1 Descripción y prioridad

El sistema debe permitir la modificación y eliminación de usuarios. Todo usuario que se registre automáticamente adquirirá la clase de suscriptor. Luego, su rol podrá ser actualizado a editor o crítico según sea necesario.

Prioridad: Alta.

### 3.2.2 Secuencias de estímulo/respuesta

El sistema permitirá la creación de usuario solamente si cumple con el llenado y formato de todos los campos requeridos (correo electrónico, nombre de usuario único y contraseña para los suscriptores y críticos) y acepta los términos y condiciones de la aplicación.

En caso de no cumplir con alguna de las restricciones, el sistema mostrará un mensaje de error o falta de llenado sobre los campos que incumplan la creación de usuario. Si el usuario cumple con todo lo necesario, se enviará un mensaje a su correo electrónico para verificarlo. En caso de que el correo no sea del usuario, este podrá modificarlo.

Para ser un usuario de clase crítico, el usuario debe aplicar para obtener una verificación.

Al crear un usuario *suscriptor* o *crítico*, el sistema deberá colocar una foto de perfil predeterminada, crear una “watchlist” y una lista de amigos vacía al usuario. Si es un *editor*, el sistema debe permitirle publicar artículos en el periódico.

El sistema permitirá la modificación o actualización de los datos de un usuario existente, siempre y cuando el campo cumpla con su formato. En caso de la actualización de la contraseña, se necesitará de la contraseña anterior y se enviará al correo electrónico un aviso de que la contraseña ha sido modificada. El sistema permitirá agregar y modificar información adicional de usuario como un “about me” y fecha de cumpleaños.

El sistema permitirá la eliminación de un usuario de clase *suscriptor* o *crítico*, sí y solo sí la contraseña registrada bajo su usuario es idéntica a la digitada en el campo antes de efectuar la eliminación, y si acepta la verificación de que en realidad quiere eliminar su cuenta.

El sistema enviará un correo electrónico al usuario para confirmar que su cuenta ha sido eliminada, que contenga un link para reestablecer su cuenta. El usuario tendrá 30 días para reestablecer su cuenta, luego de estos días, el sistema eliminará su información de la base de datos.

### 3.2.3 Requisitos funcionales

- UMGT-1: El sistema debe permitir al usuario registrarse con un correo electrónico, nombre de usuario y contraseña.
- UMGT-2: El sistema debe permitir que el usuario acepte los términos y condiciones al registrarse.
- UMGT-3: El sistema debe validar el formato de cada campo de la información ingresada por el usuario al registrarse.
- UMGT-4: El sistema debe mostrar un mensaje de error al usuario en caso de que el formato de algún campo sea incorrecto.
- UMGT-5: El sistema debe enviar un correo electrónico para confirmar su registro a la aplicación.
- UMGT-6: El sistema debe permitir al usuario modificar su correo electrónico en caso de no confirmar su registro.
- UMGT-7: El sistema debe crear un usuario que contenga una los datos ingresados, una foto de perfil predeterminada, una “watchlist”, un historial de búsqueda y una lista de amigos vacía.
- UMGT-8: El sistema debe crear una lista vacía de artículos al registrar un usuario *editor*.
- UMGT-9 El sistema debe permitir la modificación o actualización de los datos de un usuario existente.
- UMGT-10: El sistema debe verificar que el formato del campo a actualizar este correcto.
- UMGT-11: El sistema debe mostrar un mensaje de error en caso de que el campo a modificar no cumpla con su formato.
- UMGT-12: El sistema debe permitir al usuario cambiar su contraseña.
- UMGT-13: El sistema debe pedir la contraseña anterior del usuario para poder hacer el cambio de este campo.
- UMGT-14: El sistema debe validar que la contraseña anterior coincida con la almacenada en la base de datos, para poder hacer el cambio de contraseña.
- UMGT-15: El sistema debe enviar un correo electrónico de validación al usuario que modifique su contraseña.
- UMGT-16: El sistema debe permitir ingresar a los usuarios registrados, su fecha de nacimiento, género y número de celular.
- UMGT-17: El sistema debe permitir la eliminación de un usuario de clase suscriptor o crítico si y solo si la contraseña digitada es idéntica a la almacenada en la base de datos.
- UMGT-18: El sistema debe mostrar un mensaje de verificación, antes de eliminar una cuenta, solicitando confirmación.
- UMGT-19: El sistema debe enviar un correo electrónico confirmando que la cuenta del usuario ha sido eliminada, en caso de que esto pase. Además, este correo electrónico debe contener un link de recuperación.
- UMGT-20: El sistema debe permitir a un usuario reestablecer o recuperar su cuenta, para esto tendrá 30 días.
- UMGT-21: El sistema debe eliminar la información de un usuario eliminado luego de 30 días.
- UMGT-22: El sistema debe guardar la fecha y hora de registro de un usuario.
- UMGT-23: El sistema debe colocarles un símbolo de verificación que puedan ver los demás usuarios, a los usuarios registrados de clase *crítico*.
- UMGT-24: El sistema debe permitir a los usuarios de clase *suscriptor* aplicar para obtener una verificación de crítico.

UMGT-25: El sistema debe analizar la aplicación de verificación de *crítico* y notificarle al usuario que la solicita si ha sido aceptada o negada.

UMGT-26: El sistema debe crear automáticamente un ID único de usuario cuando un usuario se registre.

### 3.3 Gestionar Contenido

#### 3.3.1 Descripción y prioridad

El sistema debe mostrar la calificación, visualización de disponibilidad en el país e información (tráiler(es) oficial(es), actores, director(es), productor(es), género, ...) del contenido.

Prioridad: Alta.

#### 3.3.2 Secuencias de estímulo/respuesta

El sistema permitirá al usuario visualizar la información detallada del contenido seleccionado, mostrando el tráiler oficial, el póster promocional, el título en su idioma original, la plataforma de streaming donde el contenido está disponible, el/los género/s, la fecha de estreno, la lista de actores, directores, productores y escritores que participaron en la producción del contenido y las noticias relacionadas con el contenido.

El usuario podrá visualizar la calificación promedio del contenido, representada mediante un sistema de estrellas. Asimismo, el usuario tendrá la opción de acceder al soundtrack oficial a través de una integración con Spotify. En caso de que la banda sonora del contenido no esté disponible en dicha plataforma, el sistema mostrará un mensaje informando al usuario que no hay un soundtrack disponible para ese contenido.

#### 3.3.3 Requisitos funcionales

CMGT-1: El sistema debe realizar un promedio de calificaciones de los usuarios de cada contenido.

CMGT-2: El sistema debe mostrar la calificación promedio de un contenido por estrellas.

CMGT-3: El sistema debe mostrar la información detallada del contenido seleccionado por el usuario.

CMGT-4: El sistema debe permitir al usuario acceder al “soundtrack” del contenido a través del consumo de una API específica.

CMGT-5: El sistema debe permitir que los usuarios compartan el contenido de la plataforma en redes.

CMGT-6: El sistema debe mostrar un mensaje de error si un usuario intenta acceder a una banda sonora que no se encuentra disponible en Spotify.

CMGT-7: El sistema debe mostrar un mensaje de error si se produce un fallo en la carga del tráiler oficial del contenido seleccionado por el usuario.

CMGT-8: El sistema debe mostrar el tráiler oficial del contenido seleccionado.

CMGT-9: El sistema debe mostrar el póster promocional del contenido seleccionado.

CMGT-10: El sistema debe mostrar la lista de las personas que formaron parte de la producción de un contenido (actores, directores, guionistas, productores, etc.)

CMGT-11: El sistema debe mostrar las plataformas de streaming donde el contenido está disponible.

CMGT-12: El sistema debe mostrar noticias relacionadas con el contenido seleccionado.

CMGT-13: El sistema debe permitir a los usuarios reportar información incorrecta sobre un contenido.

CMGT-14: El sistema debe mostrar la fecha de estreno del contenido seleccionado.

## 3.4 Gestionar Prensa Digital

### 3.4.1 Descripción y prioridad

El sistema debe permitir la creación de un periódico digital que se actualizará semanalmente (todos los jueves) que contará con una sección destacada de las películas mejor valoradas por los usuarios, así como una lista de los nuevos estrenos. Además, contará con artículos sobre los mejores estrenos, tendencias cinematográficas, festivales, premios y otros eventos relevantes.

Prioridad: Alta.

### 3.4.2 Secuencias de estímulo/respuesta

El sistema permitirá a los usuarios de clase editor escribir, subir y programar la publicación de artículos en el periódico digital para que sean visualizados los jueves. El usuario editor debe agregar etiquetas del tema al momento de crear un artículo, por ejemplo, si el artículo es top 10 películas de terror del 2025, el editor debe agregar las etiquetas ranking y terror (género).

El periódico estará dividido en secciones (ranking; premios, eventos y festivales; estrenos) y cada artículo contendrá como mínimo una etiqueta que represente su sección. Además, cada artículo debe contener la fecha de publicación y el/los escritor/es.

El sistema permitirá a los usuarios ver los artículos publicados cada semana en el periódico digital, ya sea de manera general o por secciones, también podrá ver los artículos de las semanas anteriores.

El usuario editor podrá adicionar y editar artículos sobre un contenido multimedia en específico en el periódico digital. Sin embargo, solo podrá modificar artículos de su autoría. Para ello, el sistema deberá autenticar las credenciales del editor y encontrar en su historial el registro del artículo que intenta editar. Si las credenciales son erróneas o no se encuentra asociado el artículo con el editor, el sistema debe mandar un mensaje indicando que el editor no tiene permisos para modificar dicho contenido.

Un usuario de clase *editor* podrá eliminar un artículo, siempre y cuando sea de su autoría, y de una razón justa. El sistema debe autenticar su autoría y encontrar en la base de datos el archivo que intenta eliminar. En caso de que no sea el autor, debe mostrar un mensaje indicando que no es el autor y no tiene permisos para modificar el contenido.

El sistema debe notificar al autor de un artículo (usuario *editor*) que su artículo ha sido eliminado. Esto lo hará por medio del correo electrónico.

Un usuario *editor* podrá recuperar su artículo máximo 30 días luego de ser eliminado, dependiendo de la razón por la que lo haya bajado. Esto solo lo puede hacer en caso de que haya sido el editor quién lo elimino.

### 3.4.3 Requisitos funcionales

- NEWS-1: El sistema debe permitir a los usuarios de clase *editor* escribir y publicar artículos en el periódico digital.
- NEWS-2: El sistema debe permitir a los *editores* programar la publicación de artículos.
- NEWS-3: El sistema debe publicar artículos todos los jueves en el periódico semanal.
- NEWS-4: El sistema debe permitir al *editor* añadir etiquetas a los artículos por género.
- NEWS-5: El sistema debe permitir que el periódico digital se encuentre dividido en secciones: ranking; premios, eventos y festivales; estrenos.
- NEWS-6: El sistema debe añadir una etiqueta a los artículos por la sección en la que se encuentre.
- NEWS-7: El sistema debe contener como mínimo una etiqueta (la de su sección).
- NEWS-8: El sistema debe añadir colocar la fecha de publicación y escritor (usuario clase *editor*) a cada artículo.
- NEWS-9: El sistema debe añadir cada artículo publicado por un escritor al historial de artículos de un usuario clase *editor*.
- NEWS-10: El sistema debe permitir a los usuarios ver los artículos publicados cada jueves en el periódico digital.
- NEWS-11: El sistema debe permitir a los usuarios ver los artículos publicados de todas las semanas en el periódico digital.
- NEWS-12: El sistema debe permitir a los usuarios ver los artículos en cada sección del periódico digital.
- NEWS-13: El sistema debe permitir al usuario clase *editor* adicionar artículos sobre un contenido multimedia en específico en el periódico digital.
- NEWS-14: El sistema debe permitir al *editor* modificar artículos de su autoría.
- NEWS-15: El sistema debe autenticar las credenciales de un *editor* antes de dejarlo modificar un artículo.
- NEWS-16: El sistema debe buscar en el historial de artículos de un *editor* por el artículo que quiere modificar.
- NEWS-17: El sistema no debe permitir a un *editor* modificar un artículo si su autenticación falla.
- NEWS-18: El sistema debe mostrarle un mensaje al *editor* indicando que no tiene permisos para modificar un artículo en caso de que su autenticación falle.
- NEWS-19: El sistema debe permitir a los usuarios de clase *editor* eliminar un artículo si y solo si es de su autoría y coloque una razón justa.
- NEWS-20: El sistema debe verificar que el artículo que el *editor* trata de eliminar es de su autoría (que se encuentre en su historial de artículos).
- NEWS-21: El sistema debe mostrar un mensaje indicando que no es el autor y no tiene permiso para eliminar el contenido en caso de que la autenticación sea inválida.
- NEWS-22: El sistema debe notificar por correo electrónico al autor de un artículo, que este ha sido eliminado.
- NEWS-23: El sistema debe permitir a un usuario *editor* recuperar su artículo máximo 30 días luego de ser eliminado, en caso de que él lo haya eliminado.
- NEWS-24: El sistema debe eliminar un artículo de la base de datos luego de 31 días de su eliminación, en caso de que no haya sido recuperado.

## 3.5 Gestionar Publicidad

### 3.5.1 Descripción y prioridad

El sistema permite enviar correos electrónicos a los usuarios de clase *suscriptor o crítico* sobre su actividad en la plataforma, así como notificación de la publicación del periódico o recomendaciones de contenido multimedia basadas en el contenido multimedia visto anteriormente.

Prioridad: Alta.

### 3.5.2 Secuencias de estímulo/respuesta

El sistema debe recopilar estadísticas de la actividad en la plataforma de un usuario clase *suscriptor* o *crítico* cada día. Luego, el sistema creará una estadística semanal para cada usuario.

El sistema enviará correos electrónicos a los usuarios de clase *suscriptor* o *crítico* cada semana mostrando su actividad en la plataforma.

El sistema notificará sobre la publicación del periódico semanal a los usuarios por medio de su correo electrónico. Además, enviará recomendaciones de contenido multimedia, basadas en su actividad semanal, o cuando ingrese una nueva calificación.

El sistema colocará un enlace debajo de cada publicidad que les permitirá a los usuarios cancelar su suscripción a la publicidad enviada al correo electrónico.

### 3.5.3 Requisitos funcionales

ADS-1: El sistema debe recopilar estadísticas de la actividad diaria en la plataforma de un usuario suscriptor o crítico.

ADS-2: El sistema debe crear un reporte estadístico semanal de cada usuario.

ADS-3: El sistema debe enviar un correo electrónico semanalmente a un usuario clase suscriptor o crítico que contenga sus estadísticas semanales sobre la actividad en la plataforma.

ADS-4: El sistema debe notificar por medio del correo electrónico sobre la publicación del periódico semanal a todos los usuarios suscritos de clase suscriptor o crítico.

ADS-5: El sistema debe enviar al correo electrónico de los usuarios, recomendaciones de contenido multimedia cada vez que califiquen algo, basadas en esta calificación.

ADS-6: El sistema debe enviar al correo electrónico recomendaciones de contenido multimedia luego de su estadística semanal, basadas en su actividad esa semana.

ADS-7: El sistema debe colocar un link en cada correo electrónico de publicidad que permita a los usuarios cancelar su suscripción (que no le envíen más publicidad).

ADS-8: El sistema debe permitir a los usuarios cancelar su suscripción (a la publicidad enviada al correo electrónico).

## 3.6 Interacción entre Usuarios

### Descripción y prioridad

El sistema debe permitir a los usuarios tener amigos, darle “like” (me gusta) a la reseña (“review”) de otra persona, así como comentarla.

Prioridad: Media.

### Secuencias de estímulo/respuesta

Un usuario *crítico*, *suscriptor* o *editor* realiza comentarios o reseñas sobre algún contenido (como películas). El sistema permitirá que estos comentarios sean visibles y dará una mayor prioridad a las reseñas de un usuario clase *crítico*. Además, el usuario debe establecer si el comentario

contiene “spoiler” o no al momento de montarlo, debido a que el sistema señalará esto en cada comentario.

Al crear un comentario, el sistema deberá guardar la fecha, hora y su creador.

El sistema permitirá a los usuarios clase *crítico*, *suscriptor* o *editor* darle me gusta a los comentarios, así como responderlos con otro comentario.

El sistema permitirá a los creadores de un comentario eliminarlos o modificarlos. El sistema debe permitir a los usuarios ser amigos.

### Requisitos funcionales

SOCIAL-1: El sistema debe permitir a un usuario *crítico*, *suscriptor* o *editor* realizar comentarios reseñas sobre un contenido.

SOCIAL-2: El sistema debe guardar la fecha, hora y creador de un comentario.

SOCIAL-3: El sistema debe permitir que los comentarios realizados sen visibles.

SOCIAL-4: El sistema debe darles una mayor prioridad a las reseñas de un usuario clase *crítico*.

SOCIAL-5: El sistema debe permitir al usuario establecer si un comentario contiene o no “spoiler”.

SOCIAL-6: El sistema debe clasificar los comentarios entre aquellos que contienen “spoiler” y los que no.

SOCIAL-7: El sistema debe señalar si un comentario contiene o no “spoiler”.

SOCIAL-8: El sistema debe permitir a los usuarios *crítico*, *suscriptor* o *editor* darle me gusta a los comentarios de otro usuario.

SOCIAL-9: El sistema debe permitir a los usuarios *crítico*, *suscriptor* o *editor* responder a un comentario de otro usuario con un comentario.

SOCIAL-10: El sistema debe permitir a los creadores de un comentario modificarlo.

SOCIAL-11: El sistema debe mostrar cuando un comentario ha sido modificado.

SOCIAL-12: El sistema debe guardar la fecha y hora de modificación de un comentario.

SOCIAL-13: El sistema debe permitir a los creadores de un comentario eliminarlos.

SOCIAL-14: El sistema debe notificar al correo electrónico de un creador de comentario cuando su comentario haya sido eliminado.

SOCIAL-15: El sistema debe permitir a los usuarios clase *crítico*, *suscriptor* o *editor* ser amigos.

## 3.7 Interacción Usuario – Contenido

### Descripción y prioridad

El sistema permite a los usuarios crear listas en su perfil a las que le pueda agregar contenido multimedia. Las listas de contenido podrán ser públicas (solo para amigos o para todos) o privadas (para ti).

Prioridad: Alta.

### Secuencias de estímulo/respuesta

El sistema permitirá a los usuarios crear listas con diferente contenido, para esto se debe colocar un nombre diferente a los de las otras listas creadas por el usuario. Además, los usuarios deben determinar si la lista será abierta a todo público, solo para sus amigos o privada. De manera que los usuarios compartir sus listas de contenido con otros usuarios. Además, el sistema permitirá

compartir las listas creadas por el usuario por medio de un enlace para que cualquiera con el enlace pueda verla.

El sistema permitirá a los usuarios agregar contenido a las listas. El usuario también podrá tener el contenido de las listas organizado por orden alfabético (título o director) y fecha en la que se agregó.

El sistema permitirá al usuario eliminar una lista creada. El sistema permitirá a los usuarios de clase *suscriptor o crítico* calificar el contenido por medio de estrellas. Además, dependiendo del género del contenido podrá realizar una calificación especial, por ejemplo, para el contenido de romance podrán calificar por medio de corazones y nivel de lloradita.

### Requisitos funcionales

UCI-1: El sistema debe permitir crear una lista de contenido multimedia vacía a los usuarios de clase *suscriptor, crítico y editor*.

UCI-2: El sistema debe permitir colocar un nombre a la lista que admita letras, números, caracteres especiales y emojis.

UCI-3: El sistema debe verificar que el nombre de la lista debe tener al menos un carácter.

UCI-4: El sistema debe verificar que el nombre la lista tenga un máximo de 120 caracteres.

UCI-5: El sistema debe mostrar un mensaje de error en caso de que el nombre de la lista sea igual al de alguna lista del usuario.

UCI-6: El sistema debe mostrar un mensaje de error al usuario en caso de que el nombre de la lista no cuente con el rango de caracteres (de uno a 120).

UCI-7: El sistema debe administrar la privacidad de las listas, es decir, que sean abiertas a todo el público, a los amigos del usuario o privada.

UCI-8: El sistema debe colocar por defecto la lista debe ser pública.

UCI-9: El sistema debe permitir que el usuario cambie la privacidad de las listas.

UCI-10: El sistema debe permitir al usuario compartir sus listas de contenido con otros usuarios.

UCI-11: El sistema debe permitir al usuario crear un enlace para compartir una lista de contenido creada.

UCI-12: El sistema debe crear un enlace para poder compartir una lista de contenido con las personas que lo usen.

UCI-13: El sistema debe organizar el contenido de las listas por defecto por medio de la fecha en la que se agregó el contenido a este.

UCI-14: El sistema debe permitir al usuario organizar el contenido de las listas por un orden alfabético guiado por el título o director.

UCI-15: El sistema debe permitir al usuario eliminar una lista creada.

UCI-16: El sistema debe permitir al usuario eliminar contenido de una lista.

UCI-17: El sistema debe notificar por correo electrónico al creador de una lista cuando esta sea eliminada.

UCI-18: El sistema debe mostrar una notificación al usuario al momento de eliminar una lista.

UCI-19: El sistema debe permitir al usuario aceptar o cancelar la notificación de eliminar una lista.

UCI-20: El sistema debe permitir a los usuarios de clase *suscriptor o crítico* calificar el contenido por medio de estrellas.

UCI-21: El sistema debe permitir a los usuarios de clase *suscriptor o crítico* realizar calificaciones de contenido clasificados por género.

UCI-22: El sistema debe guardar la calificación sobre un contenido de un usuario.



## 3.8 Búsqueda y Filtrado

### Descripción y prioridad

El sistema permite realizar opciones avanzadas de búsqueda y filtrado del contenido multimedia por medio de características como género, actor, director, productor, escritor, plataformas disponibles, año de estreno, rango de duración y rating. Asimismo, el sistema permite realizar opciones avanzadas de búsqueda y filtrado de artículos del periódico por medio de las características fecha, escritor y sus etiquetas como sección.

Prioridad: Media.

### Secuencias de estímulo/respuesta

El sistema permitirá a los usuarios realizar una búsqueda avanzada y filtrada de contenido. El usuario debe elegir una o más opciones/filtros. El sistema mostrará el contenido multimedia que contenga la(s) característica(s) elegida(s) por el usuario.

El sistema permitirá a los usuarios realizar una búsqueda avanzada y filtrada de artículos al entrar al periódico. El usuario podrá elegir una o más opciones. El sistema mostrará los artículos que contengan las características ingresadas por el usuario.

En caso de que no se encuentre contenido multimedia o artículos que contengan las características elegidas por el usuario, el sistema debe mostrar un mensaje que diga que no se haya. El sistema mostrará contenido multimedia debajo del mensaje, que contenga alguno de los filtros ingresados por el usuario o recomendaciones basadas en los gustos del usuario. Si es un artículo, el sistema mostrará los últimos artículos publicados en el periódico.

El sistema guardará las búsquedas, creando un historial que el usuario podrá visualizar en cualquier momento.

### Requisitos funcionales

SANDF-1: El sistema debe permitir a los usuarios buscar contenido multimedia.

SANDF-2: El sistema debe permitir a los usuarios realizar una búsqueda avanzada haciendo uso del filtrado por una o más características del contenido multimedia.

SANDF-3: El sistema debe permitir usuario debe elegir una o más características por las cuales desea filtrar el contenido multimedia.

SANDF-4: El sistema debe mostrar el contenido multimedia que contenga las características elegidas por el usuario en una búsqueda avanzada.

SANDF-5: El sistema debe mostrar el contenido multimedia encontrado según la búsqueda de un usuario.

SANDF-6: El sistema debe mostrar un mensaje en caso de que no se encuentre contenido multimedia con las características buscadas.

SANDF-7: El sistema debe mostrar contenido multimedia recomendado en caso de que no se encuentre lo que el usuario está buscando.

SANDF-8: El sistema debe permitir a los usuarios buscar artículos en el periódico digital.

SANDF-9: El sistema debe mostrar al usuario los resultados de su búsqueda en el periódico.

SANDF-10: El sistema debe permitir a los usuarios hacer búsquedas avanzadas de los artículos en el periódico digital.

SANDF-11: El sistema debe permitir a los usuarios seleccionar y especificar características sobre los artículos para hacer búsquedas avanzadas.

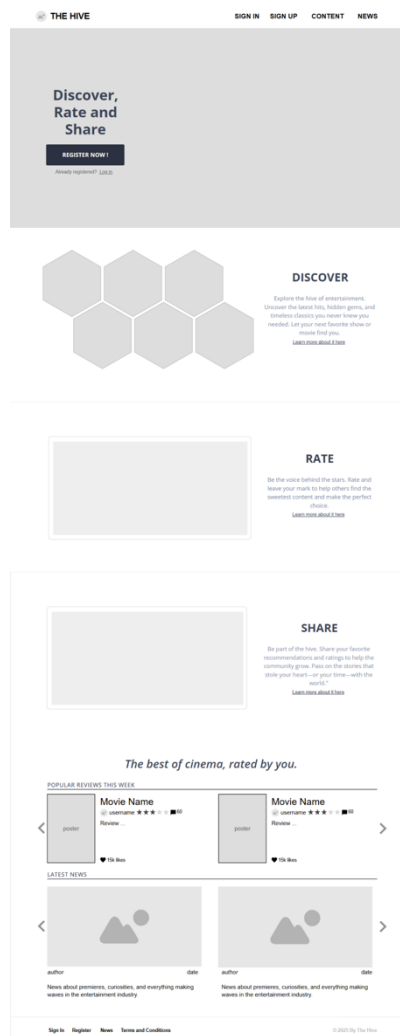
- SANDF-12: El sistema debe mostrar al usuario los resultados de artículos de su búsqueda avanzada.
- SANDF-13: El sistema debe mostrar un mensaje en caso de que no se encuentren artículos relacionados a la búsqueda.
- SANDF-14: El sistema debe mostrar los artículos publicados más recientes si no hay resultados de búsqueda de artículos en el periódico.
- SANDF-15: El sistema guardará un historial de búsquedas del usuario.
- SANDF-16: EL sistema permitirá al usuario visualizar su historial de búsquedas.

## Requisitos de la interfaz

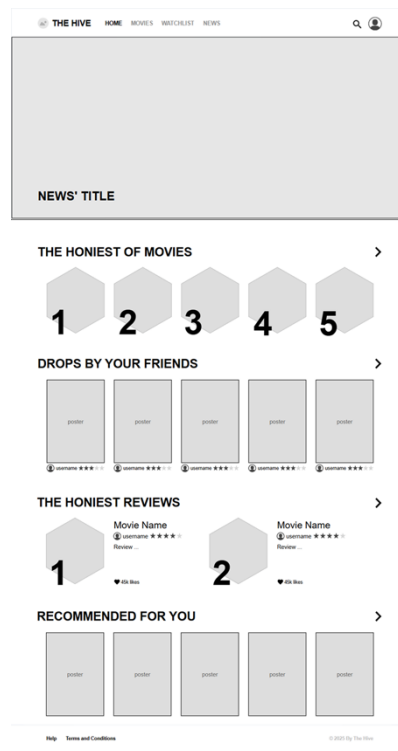
### 3.9 Interfaces de Usuario

A continuación, se mostrarán los distintos Mock Ups (vistas en construcción) de la aplicación *The Hive*.

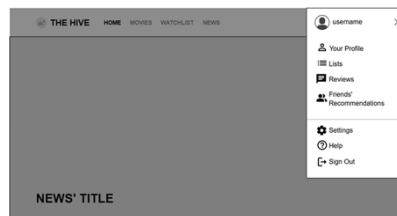
#### Inicio sin autenticación



## Inicio con autenticación



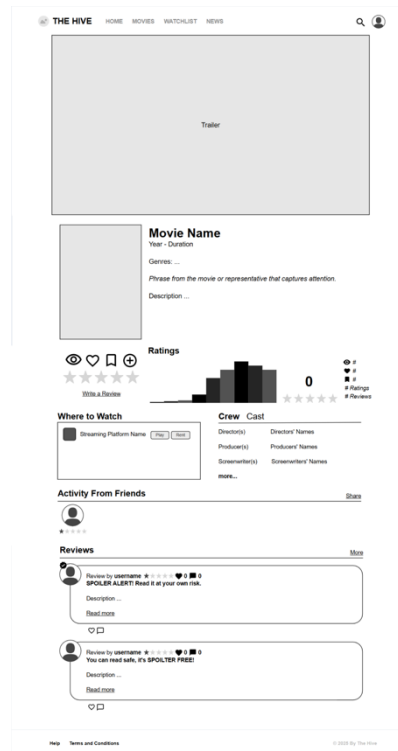
## Sección del usuario en el inicio



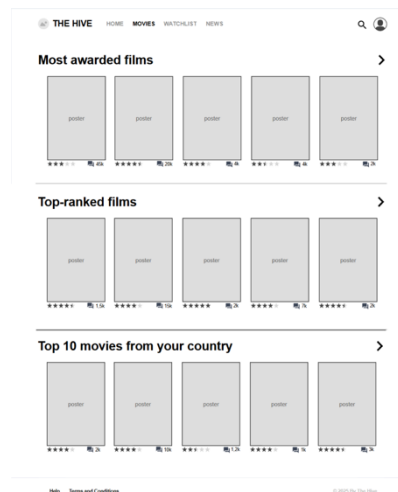
## Perfil del usuario



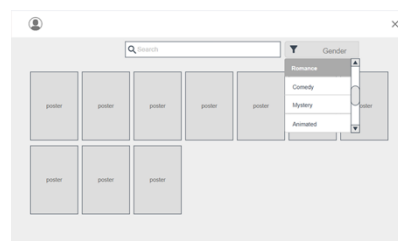
## Información sobre la película



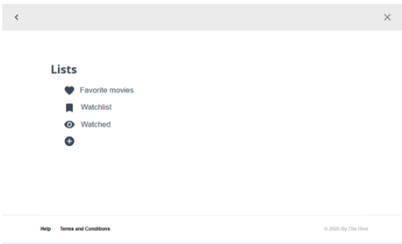
## Catálogo de películas



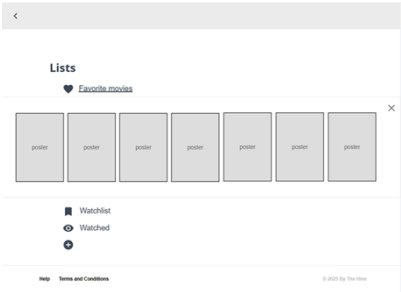
## Búsqueda de películas



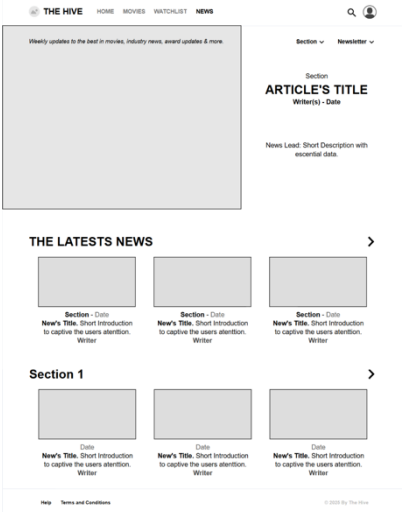
Vista de listas de películas



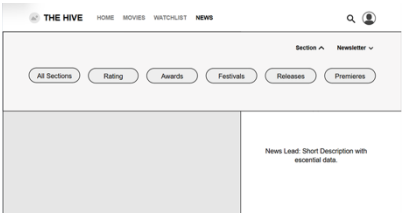
Ver información de la lista



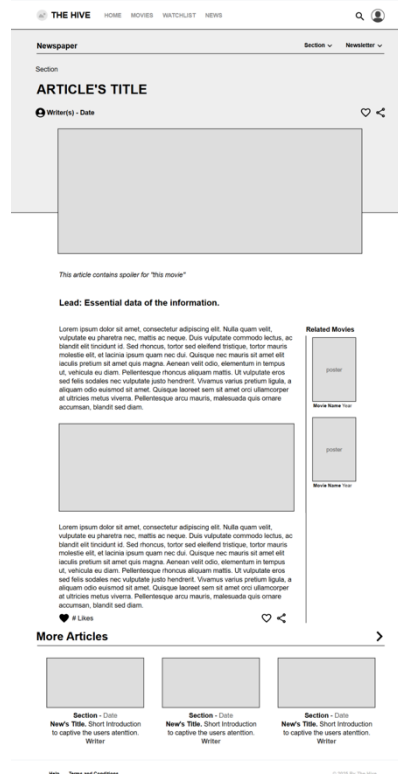
Catálogo de Noticias



Busqueda de Noticias



## Vista de la noticia



## 3.10 Interfaces de hardware

### Interfaz de usuario

- **Dispositivos admitidos:** Navegadores web modernos (Flutter Web).
- **Características lógicas:**
  - UI interactiva en Flutter (Dart) que consume servicios REST.
  - Entrada del usuario: teclado/táctil para comentarios y calificaciones.
  - Datos en formato JSON para renderizar la vista master-detail
  - Comunicación asíncrona con el backend usando HTTP.
- **Protocolos:**
  - HTTP/HTTPS para llamadas REST.
  - JSON como formato de intercambio de datos entre frontend y backend.

### Interfaz del backend

- **Tipo de dispositivo admitido:** Contenedor Docker
- **Características lógicas:**
  - API REST para manejar solicitudes de comentarios, calificaciones y noticias.
  - Procesamiento de lógica de negocio, control de acceso, validación de datos.
  - Comunicación con base de datos para almacenar y recuperar información.
- **Protocolos:**
  - HTTP/HTTPS para el frontend.
  - JSON como formato de intercambio de datos.
  - REST para los microservicios involucrados.

### Interfaz con la base de datos

- **Tipo de dispositivo admitido:** SQLite.
- **Características lógicas:**
  - Consultas SQL o llamadas a través de ORM para leer/escribir datos.
  - Almacenamiento de usuarios, películas, reacciones, calificaciones y noticias.

## 3.11 Interfaces de software

Este producto de software interactúa con diversos componentes para cumplir con sus funcionalidades. A continuación se describen las conexiones, herramientas, bibliotecas y mensajes implicados en su funcionamiento:

### Componentes del Backend (Python)

- **Framework principal:** FastAPI (versión reciente): permite la creación de servicios RESTful eficientes y asíncronos.
- **Modelado de datos y validación:** Pydantic y SQLAlchemy, usados para definir esquemas de datos, validarlos automáticamente y facilitar la interacción con SQLite.
- **Seguridad y autenticación:**
  - **bcrypt:** utilizado para el cifrado de contraseñas de los usuarios.
  - **uuid:** utilizado para generar los distintos identificadores únicos.

### Base de Datos

- SQLite (motor de base de datos relacional):
  - Los datos persistentes del sistema se almacenan en un archivo *.sqlite*.
  - El acceso se realiza mediante SQLAlchemy, con operaciones CRUD para usuarios, películas, comentarios, calificaciones y noticias.
  - No se requiere un servidor de base de datos externo.

### Mensajes Entrantes y Salientes

El backend expone una serie de endpoints HTTP que reciben y envían datos en formato JSON. Todos los mensajes son validados y procesados con FastAPI y Pydantic, asegurando integridad y estructura consistente.

### Frontend Flutter

- Utiliza librerías nativas de Flutter y plugins comunes para interfaces web.
- Se comunica con el backend a través de solicitudes HTTP usando el paquete http.
- Los datos se consumen en formato JSON, renderizados en componentes como listas, tarjetas o vistas de detalle (master-detail).

## 3.12 Interfaces de Comunicación

El sistema requiere funciones de comunicación para garantizar la interacción entre el cliente (frontend en Flutter) y el servidor (backend en Python con FastAPI). A continuación se detallan los elementos clave:

### Navegador Web y Protocolo

- El sistema se ejecutará en navegadores web modernos como Chrome, Firefox, y Edge.
- El protocolo de comunicación entre el frontend y el backend será HTTP/HTTPS, siendo HTTPS obligatorio en ambientes de producción para garantizar la seguridad y cifrado de los datos transmitidos.

### Formato de Mensajes

Todos los mensajes entre cliente y servidor se enviarán en formato JSON, tanto en las solicitudes (request) como en las respuestas (response).

### Seguridad y Cifrado

- Las comunicaciones estarán protegidas mediante el uso de HTTPS, con certificados SSL válidos.



- Las contraseñas de los usuarios se enviarán cifradas y serán almacenadas con hashing seguro (bcrypt) en el servidor.
- Los tokens de sesión serán transmitidos de forma segura por cabeceras HTTP (Authorization) y tendrán un tiempo de expiración definido.

## 4. Otros requisitos No funcionales

### 4.1 Requerimientos de desempeño

- **Tiempo de respuesta del servidor:** Cuando un usuario hace una acción (como ver una película o dejar un comentario), el servidor debe responder en menos de 2 segundos, para que la experiencia se sienta rápida y fluida.
- **Carga rápida de la aplicación web:** La página debe estar lista para usarse en máximo 5 segundos después de que el usuario la abra en su navegador. Esto ayuda a que no se desesperen esperando.
- **Actualización de la vista master-detail:** La carga del detalle de una película seleccionada desde la lista debe completarse en menos de 1 segundo, para mantener una experiencia fluida al usuario.
- **Contraseñas seguras:** Las contraseñas que los usuarios creen deben tener al menos 8 caracteres, con letras, números y símbolos especiales (como @, # o !), para que no sean fáciles de adivinar.
- **Conexión segura (HTTPS):** Toda la comunicación entre el usuario y el servidor debe estar cifrada usando HTTPS, así los datos (como contraseñas o comentarios) no se pueden ver ni robar fácilmente en el camino.
- **Respaldo de datos:** Cada día se hará una copia de seguridad automática de la base de datos (archivo SQLite), para que si algo falla, puedas recuperar la información sin perder nada importante.

### 4.2 Requisitos de seguridad

#### Autenticar al usuario.

El sistema implementará un mecanismo de autenticación que permitirá verificar la identidad de cada usuario mediante correo electrónico y contraseña, garantizando que solo usuarios registrados puedan acceder a las funcionalidades personalizadas de la plataforma.

#### Confirmar términos y condiciones para protección de los datos personales.

Antes de completar el registro, el usuario deberá aceptar los términos y condiciones, los cuales incluyen una política clara sobre el tratamiento y protección de los datos personales, en cumplimiento con las normativas de privacidad vigentes.

**Cifrado de datos.**

Toda la información transmitida entre el cliente y el servidor estará protegida mediante cifrado HTTPS, y las contraseñas se almacenarán de forma segura utilizando algoritmos de hashing como bcrypt para evitar el acceso no autorizado.

**Respaldo de recuperación de datos**

El sistema contará con mecanismos de respaldo periódico de la base de datos, permitiendo la recuperación de la información en caso de pérdida, fallo técnico o ataques externos.

**Contraseña con mínimo 8 con un número, carácter especial y letras.**

Durante el registro, se requerirá que las contraseñas tengan al menos ocho caracteres, incluyendo una letra mayúscula, una minúscula, un número y un carácter especial, con el fin de fortalecer la seguridad de las cuentas de usuario.

## **4.3 Atributos de calidad del software**

**Usabilidad:**

La interfaz estará diseñada con principios de diseño intuitivo y accesible, permitiendo que un nuevo usuario pueda registrarse, calificar y comentar películas en menos de 3 minutos sin necesidad de ayuda externa. La usabilidad será priorizada por encima de la facilidad de aprendizaje, favoreciendo una experiencia de navegación fluida y atractiva desde el primer uso.

**Disponibilidad:**

El sistema deberá tener una disponibilidad mínima del 99% mensual, lo que equivale a un tiempo de inactividad máximo de aproximadamente 7 horas al mes. Para ello se utilizarán servicios estables y monitoreo básico en el servidor backend.

**Fiabilidad:**

La aplicación debe ser capaz de manejar correctamente solicitudes concurrentes de al menos 100 usuarios simultáneos sin perder datos ni generar errores, asegurando una experiencia consistente.

**Mantenibilidad:**

El código fuente estará estructurado en módulos desacoplados (frontend, y backend), con comentarios claros y documentación técnica básica, facilitando que un desarrollador pueda realizar una corrección o mejora menor en menos de 2 horas.

**Portabilidad:**

La solución podrá ejecutarse en navegadores modernos mediante Flutter Web, con compatibilidad verificada en Chrome, Firefox y Edge.

**Interoperabilidad:**

El sistema utilizará servicios REST con JSON como formato de intercambio, permitiendo su integración futura con otros sistemas como APIs de recomendaciones o redes sociales, sin necesidad de grandes modificaciones.

**Corrección:**

El sistema debe cumplir con todos los requisitos funcionales establecidos, permitiendo acciones como ver noticias, comentar y calificar películas sin errores funcionales en al menos el 95% de los casos de prueba.

**Reutilización:**

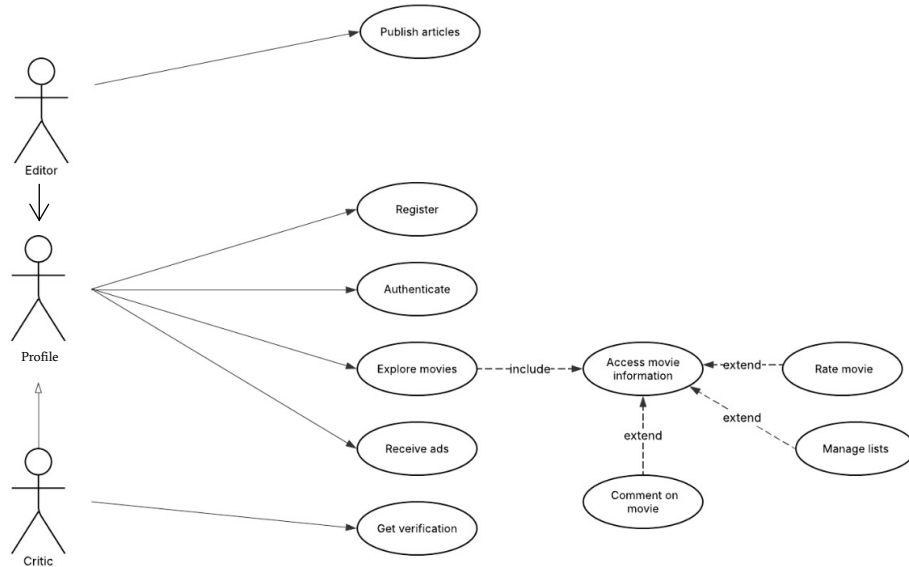
Los componentes del frontend estarán diseñados en Flutter como widgets reutilizables (ej. tarjetas de películas, vistas de detalle), facilitando su uso en otras partes del sistema o en futuros proyectos.

## Apéndice A: Glosario

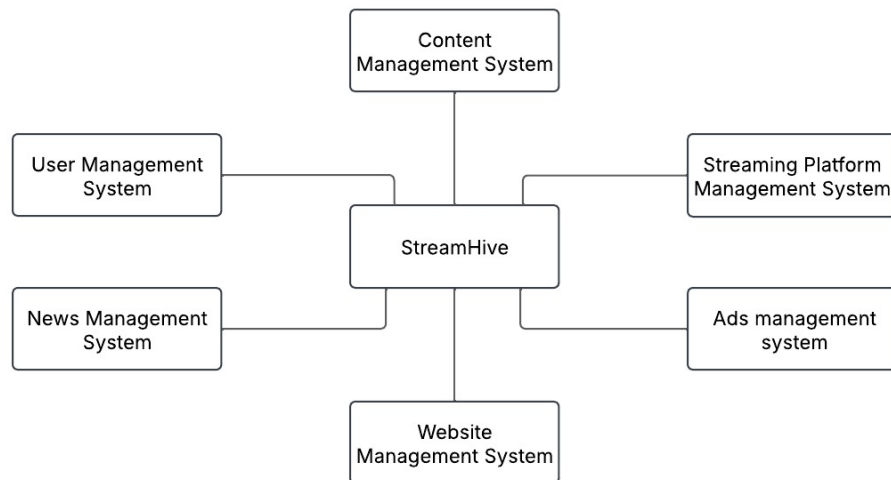
- **API (Application Programming Interface):** Conjunto de reglas y definiciones que permiten que el frontend y backend se comuniquen entre sí mediante solicitudes y respuestas, generalmente usando JSON.
- **JSON (JavaScript Object Notation):** Formato de texto ligero para el intercambio de datos estructurados, usado entre el frontend Flutter y el backend en Python.
- **Flutter:** Framework de desarrollo de interfaces de usuario desarrollado por Google, utilizado para construir el frontend del sistema.
- **SQLite:** Motor de base de datos relacional ligero y autocontenido, utilizado para almacenar los datos del sistema.
- **Frontend:** Parte del software con la que interactúa el usuario directamente, en este caso desarrollado con Flutter.
- **Backend:** Parte del software que procesa la lógica de negocio, la autenticación, y el acceso a la base de datos. En este proyecto está desarrollado en Python.
- **SRS (Software Requirements Specification):** Documento que define de forma precisa y completa los requisitos funcionales y no funcionales del sistema.
- **CRUD:** Acrónimo de Create, Read, Update y Delete. Son las operaciones básicas que se realizan sobre los datos.
- **HTTPS:** Protocolo seguro de transferencia de hipertexto, usado para la comunicación cifrada entre cliente y servidor.
- **UI (User Interface):** Interfaz de usuario. Es la parte visual y de interacción del sistema.
- **UX (User Experience):** Experiencia del usuario. Evalúa qué tan fácil, agradable y útil es interactuar con el sistema.
- **REST (Representational State Transfer):** Estilo de arquitectura de software utilizado para diseñar servicios web, comúnmente con métodos HTTP.
- **Master-Detail:** Patrón de diseño donde se muestra una lista (master) y, al seleccionar un ítem, se despliega el detalle del mismo.
- **Hashing:** Técnica criptográfica usada para almacenar contraseñas de manera segura (por ejemplo, usando bcrypt).
- **ORM (Object-Relational Mapping):** Herramienta que permite trabajar con bases de datos relacionales usando objetos en código, como SQLAlchemy en Python.

## Apéndice B: Modelos de análisis

### Diagrama de caso de uso

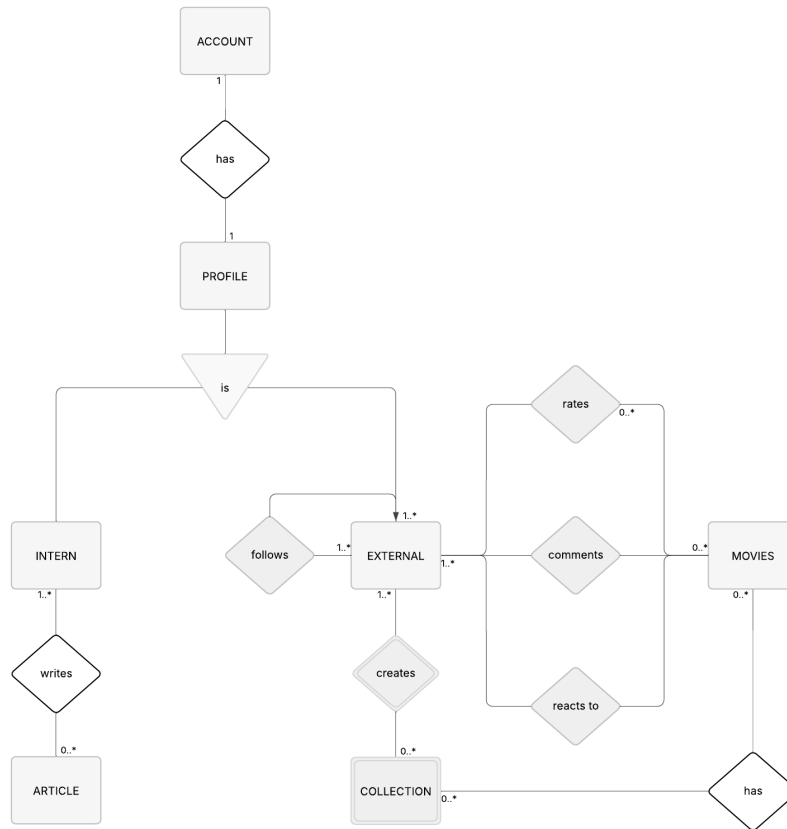


### Diagrama de contexto

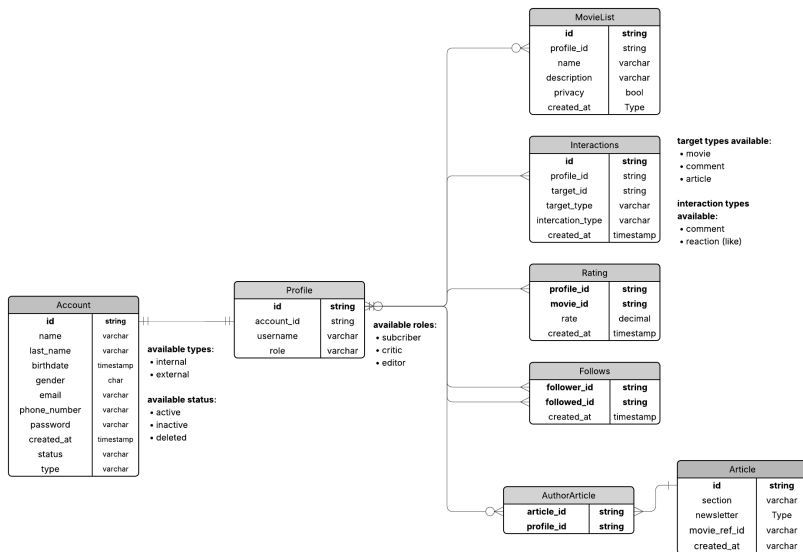


## Apéndice C: Diagramas estructurales

### Modelo Entidad-Relación



## Tablas de relación



## Modelo máster-detail

