



Institut Puig Castellar
Santa Coloma de Gramenet



CFGS Desarrollo de Aplicaciones Web

“Plataforma de Reseñas y Comunidad de Cinéfilos y Seriéfilos”

POSTULANTE : CRISTIAN TAPASCO

CRISTIAN CALDERON

TUTOR : YAGO MORALES

BARCELONA - ESPAÑA 2025

Índice

1. Introducción	5
1.1. Contexto	5
1.2. Justificación	6
1.3. Objetivo	6
1.3.1. Objetivo general	6
1.3.2. Objetivos específicos	6
1.4. Estrategia y planificación del proyecto	7
1.5. Metodología de trabajo	8
1.5.1. Kanban	8
1.5.2. Fases de Kanban	9
1.5.3. Componentes de Kanban	9
1.6. Estudio económico y presupuesto	9
2. Descripción del proyecto	11
2.1. Análisis de requisitos	12
2.1.1. Requisitos funcionales	12
2.1.2. Requisitos no funcionales	13
2.2. Previsión de tareas de investigación	14
2.3. Tecnologías	15
2.3.1. Comparativa de las Tecnologías Valoradas	15
2.3.2. Tecnologías escogidas	16
2.4. Estructura del proyecto	17
2.5. Descripción de los componentes	18
2.5.1. Frontend	19
2.5.1.1 Registro y autenticación de usuarios	19
2.5.1.2. Gestión del perfil usuario	20
2.5.1.3. Búsqueda y consulta de contenido	21
2.5.1.4. Valoración y reseña de contenido	23
2.5.1.5. Sistema de favoritos	24
2.5.1.6. Visualización de historial de contenido (Películas y Series)	25
2.5.1.7. Visualización de logros	26
2.5.1.8 Búsqueda unificada de contenido	27
2.5.1.9. Detalle de la película o serie	28
2.5.1.10. Calificación de contenido	29
2.5.1.11. Visualización de últimas calificaciones	30
2.5.2. Backend	31
2.5.2.1. Gestión de contenido	31
2.5.2.2. Gestión de favoritos	32
2.5.2.3 Gestión de calificaciones	33
2.5.2.4 Gestión de logros	34
2.5.2.5. Gestión social	35
2.5.3. Base de datos	36
2.5.4. API externa	39

2.5.4.1. Propósito de la integración	39
2.5.4.2. Interacción con TMDb	39
2.5.4.3. Gestión segura y eficiente de la API	40
2.5.4.4. Funcionalidades potenciadas por TMDb	40
2.6. Arquitectura del sistema	40
2.6.1. Implementación	40
3. Diseño y pruebas del sistema	43
3.1. Fase de diseño	43
3.1.1. Diseño de interfaz abstracta	43
3.1.2. Diseño navegacional	50
3.2. Fase de pruebas	51
3.2.1. Etapa de Ejecución	51
3.2.2. Pruebas de caja blanca	51
3.2.2. Pruebas de caja negra	57
4. Conclusiones	68
4.1. Conclusiones generales del proyecto	68
4.2. Consecuencias de los objetivos	68
4.3. Valoración de la metodología y planificación	68
4.4. Visión de futuro	69
5. Glosario	71
6. Referencias bibliográficas	74

Índice de Tablas

1. Introducción	5
Tabla 1.1. Costo de desarrollo	9
Tabla 1.2 Costo de mantenimiento (Anuales)	10
2. Descripción del proyecto	12
Tabla 2.1. Tecnologías utilizadas	17
Tabla 2.2. Modelo Relacional	35
3. Diseño y pruebas del sistema	42
Tabla 3.1. Caso de prueba: Login y Registro	50
Tabla 3.2. Caso de prueba: Edición de perfil	51
Tabla 3.3. Caso de prueba: Agregar contenido a favoritos	51
Tabla 3.4. Caso de prueba: Eliminar contenido del historial	52
Tabla 3.5. Caso de prueba: Calificar contenido	52
Tabla 3.6. Caso de prueba: Enviar solicitud de amistad	53
Tabla 3.7. Caso de prueba: Desbloqueo de logro por visualizaciones	53
Tabla 3.8. Caso de prueba: Caso de prueba: Buscar contenido audiovisual	54
Tabla 3.9. Caso de prueba: Cierre de sesión	55
Tabla 3.10. Caso de prueba: Ver el detalle de una película o serie desde su ID	55
Tabla 3.11. Caso de prueba: Login	56
Tabla 3.12. Caso de prueba: Registro de usuario	57
Tabla 3.13. Caso de prueba: Editar perfil	58
Tabla 3.14. Caso de prueba: Búsqueda de contenido	59

Tabla 3.15. Caso de prueba: Visualizar perfil público de otro usuario	60
Tabla 3.16. Caso de prueba: Enviar solicitud de amistad	61
Tabla 3.17. Caso de prueba: Ver el detalle de una película desde su ID	62
Tabla 3.18. Caso de prueba: Calificar contenido (película o serie)	63
Tabla 3.19. Caso de prueba: Agregar contenido a favoritos	64
4. Conclusiones	67
5. Glosario	70
6. Referencias bibliográficas	73
7. Anexos	76

Índice de Figuras

1. Introducción	5
Figura 1.1. Modelo Kanban	8
2. Descripción del proyecto	11
Figura 2.1. Estructura general del proyecto	18
Figura 2.2. Diagrama de casos de uso Registro y Login	19
Figura 2.3. Diagrama de casos de uso Editar Perfil	20
Figura 2.4. Diagrama de casos de uso Búsqueda de película	22
Figura 2.5. Diagrama de casos de uso Valoración de películas o series	23
Figura 2.6. Diagrama de casos de uso Sistema de Favoritos	25
Figura 2.7. Diagrama de casos de uso Historial de Contenido	26
Figura 2.8. Diagrama de casos de uso Visualización de Logros obtenidos	27
Figura 2.9. Diagrama de casos de uso Búsqueda de Contenido agregado	27
Figura 2.10. Diagrama de secuencia Detalle de contenido	29
Figura 2.11. Diagrama de secuencia Calificación de contenido	30
Figura 2.12. Diagrama de secuencia Visualización de últimas calificaciones	30
Figura 2.13. Diagrama de secuencia Gestión de contenido	32
Figura 2.14. Diagrama de secuencia Gestión de favoritos	33
Figura 2.15. Diagrama de secuencia Gestión de calificaciones	34
Figura 2.16. Diagrama de secuencia Gestión de Logros	35
Figura 2.17. Diagrama de secuencia Gestión Social	36
Figura 2.18. Diagrama Entidad Relación	38
Figura 2.19. Diagrama de secuencia Comunicación con la API Externa	39
Figura 2.20. Diagrama de Arquitectura MVC	41
3. Diseño y pruebas del sistema	43
Figura 3.1. Pantalla Principal	44
Figura 3.2. Editar Perfil	45
Figura 3.3. Resultado de búsqueda películas y series	46
Figura 3.4. Pantalla de amigos	47
Figura 3.5. Pantalla información personal de película o serie	49
Figura 3.6. Pantalla Historial	50
Figura 3.7. Diagrama de flujo navegacional	50

74

Capítulo I

Generalidades

1. Introducción

El proyecto **CineChamp** surge como una innovadora aplicación web diseñada exclusivamente para los amantes del cine y las series de nuestras plataformas favoritas. En la actualidad, el acceso masivo a cualquier contenido ha generado una oferta abrumadora, dificultando a los usuarios la elección de qué ver. Frente a esta problemática, busca convertirse en el punto de referencia para la comunidad de cinéfilos y seriéfilos, proporcionando un espacio especializado donde puedan compartir sus experiencias, descubrir nuevas recomendaciones y conectar con personas que comparten sus mismos intereses.

A través de **CineChamp**, los usuarios pueden publicar reseñas detalladas, puntuar películas y series, y llevar un registro personalizado de sus visualizaciones, creando un historial organizado que les ayude a gestionar su contenido favorito. Lo que diferencia de otras redes sociales y sitios de reseñas es su enfoque exclusivo en la experiencia del usuario y la generación de una comunidad interactiva. La plataforma está diseñada para ofrecer un sistema intuitivo y eficiente que permita descubrir contenido relevante basado en las opiniones y valoraciones de otros usuarios con gustos afines.

CineChamp no solo pretende ser un repositorio de reseñas, sino un espacio dinámico de interacción, donde se fomente el debate, el intercambio de opiniones y el descubrimiento de nuevas producciones audiovisuales de una manera más personalizada y confiable. La plataforma contará con un sistema de logros temáticos vinculados a eventos especiales del mundo del cine y las series, como Halloween, estrenos destacados o festivales de cine. Estos logros permitirán a los usuarios desbloquear recompensas exclusivas al completar desafíos relacionados con estos eventos, incentivando la participación y el compromiso dentro de la comunidad.

Con esta propuesta, buscamos crear un espacio pequeño en el mundo de la red donde revolucionar la forma en que los usuarios exploran el mundo del cine y las series, convirtiéndose en una herramienta para los verdaderos amantes del entretenimiento audiovisual.

1.1. Contexto

CineChamp surge de la necesidad del equipo de desarrollo de contar con una plataforma que permita llevar un control y registro de las películas y series vistas a lo largo de su vida. La idea nace del deseo de los miembros del equipo de organizar su historial audiovisual de manera eficiente, al tiempo que se genera una sana competencia entre los usuarios

mediante la acumulación de logros y estadísticas relacionadas con la cantidad de películas y series vistas.

1.2. Justificación

Si bien existen diversas plataformas de streaming y redes sociales, no hay un espacio especializado en habla hispana donde los usuarios puedan llevar un control detallado de sus películas y series en forma de historial. **CineChamp** pretende llenar este vacío al ofrecer una plataforma donde los usuarios puedan agregar, eliminar, marcar como favoritas o pendientes sus películas y series vistas, además de permitir la interacción con otros usuarios agregando amigos y compartiendo sus experiencias.

A diferencia de otras redes sociales, **CineChamp** se centra en la creación de una comunidad activa que permita a los usuarios llevar un registro personal y competitivo de su consumo audiovisual. Los usuarios no solo podrán descubrir nuevas producciones gracias a las opiniones de la comunidad, sino también competir amistosamente con otros en base a logros y estadísticas, fomentando una experiencia más dinámica y participativa dentro del ámbito del entretenimiento audiovisual.

1.3. Objetivo

1.3.1. Objetivo general

Desarrollar una plataforma web en la que los usuarios puedan llevar un control completo de su historial de películas y series, permitiéndoles añadir, eliminar, marcar como favoritas o pendientes, así como interactuar con otros usuarios a través de una comunidad activa de cinéfilos y seriéfilos. Además, se fomentará la competencia amistosa mediante un sistema de logros y estadísticas personales.

1.3.2. Objetivos específicos

- Implementar un sistema de publicación y puntuación de contenido audiovisual.
- Diseñar una interfaz sencilla que facilite la navegación y la interacción entre los usuarios.
- Permitir a los usuarios gestionar su historial de películas y series con opciones para agregar, eliminar y organizar contenido.
- Desarrollar un sistema de logros y recompensas temáticas basado en la actividad del usuario.

- Integrar un sistema de amigos para que los usuarios puedan comparar sus historiales y logros.

1.4. Estrategia y planificación del proyecto

El desarrollo de **CineChamp** surge como respuesta a la necesidad de contar con una plataforma funcional que permita a los usuarios organizar, compartir y valorar su consumo audiovisual. Inspirado en el auge de las plataformas de streaming y las redes sociales centradas en contenido multimedia, este proyecto busca integrar lo mejor de ambas propuestas mediante un enfoque gamificado y una comunidad activa de usuarios.

Aunque existen aplicaciones con funcionalidades similares, **CineChamp** propone una solución adaptada a un público hispanohablante, incorporando características diferenciadoras como logros temáticos, perfiles personalizados y seguimiento detallado del historial audiovisual. Para garantizar la coherencia del sistema con sus objetivos, se ha optado por desarrollar una solución propia que permite un control total sobre el diseño, las funcionalidades y la experiencia de usuario.

El equipo cuenta con los conocimientos técnicos necesarios y el acceso a tecnologías accesibles y actuales, lo que ha permitido abordar de forma autónoma todo el proceso de desarrollo, facilitando la mejora continua y la incorporación de retroalimentación de los usuarios.

La planificación del proyecto se dividió en tres fases clave:

- Fase de diseño: Incluyó el análisis de requisitos funcionales y técnicos, investigación de referencias visuales, elaboración de prototipos mediante herramientas como Figma, y definición de la estructura general de la plataforma con énfasis en la experiencia de usuario.
- Fase de desarrollo: Se implementó el frontend utilizando React y TypeScript, mientras que el backend se construyó con Node.js y Express. La base de datos se diseñó en MySQL y se integró la API externa de TMDb para enriquecer el contenido con datos actualizados de películas y series.
- Fase de prueba: Se verificó el funcionamiento general del sistema a través de pruebas funcionales, navegación por la interfaz y revisión de las funcionalidades clave. Esta etapa aseguró la coherencia e integración de todos los componentes del sistema.

1.5. Metodología de trabajo

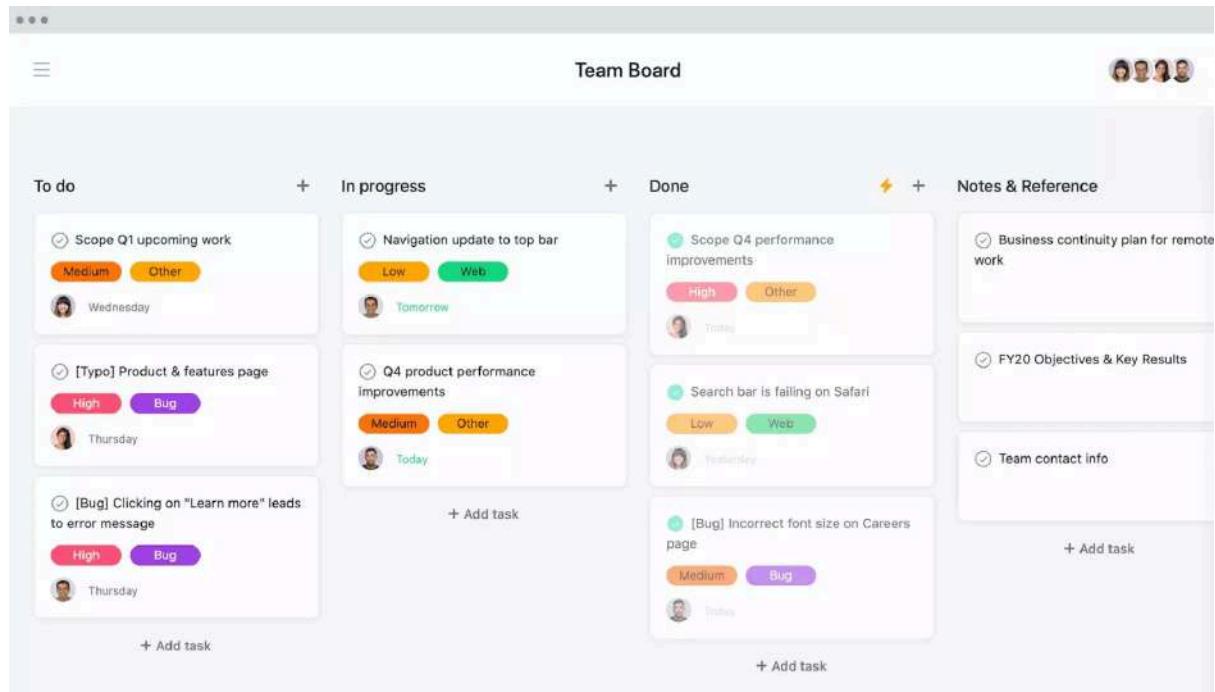
Para el desarrollo de **CineChamp**, se emplea la metodología Ágil Kanban, permitiendo una gestión flexible del proyecto y una rápida adaptación a los cambios.

1.5.1. Kanban

Kanban es una metodología ágil se centra en visualizar todo el proyecto en tableros para aumentar transparencia del proyecto y colaboración entre los miembros del equipo.

Es una metodología que tiene como objetivo la mejora continua, la flexibilidad en gestión de tareas y flujo de trabajo mejorada, con este enfoque ilustrativo, el progreso de todo el proyecto se puede entender fácilmente de un vistazo.

Figura 1.1. Modelo Kanban



Fuente: [\[https://asana.com/es/resources/what-is-kanban\]](https://asana.com/es/resources/what-is-kanban) (2025)

1.5.2. Fases de Kanban

El flujo de trabajo de Kanban se divide en tres etapas principales que ayudan a visualizar y gestionar el avance de las tareas:

- Pendiente (To Do) : Acá se colocan todas las tareas que están por iniciar, es la lista de trabajo por hacer.

- En progreso (In Progress): Son las tareas en las que el equipo está trabajando activamente. Es importante limitar la cantidad de tareas aquí para evitar la sobrecarga.
- Completado (Done): Esta fase representa las tareas finalizadas, permite hacer seguimiento del progreso y analizar la eficiencia del equipo.

1.5.3. Componentes de Kanban

Kanban se apoya en varios componentes clave para funcionar correctamente:

- Tablero Kanban: Herramienta visual donde se reflejan todas las tareas del proyecto, puede ser físico en una pizarra o notas adhesivas en forma de columna o digital en nuestro caso usamos Trello.
- Tarjetas (Cards): Se representan tareas individuales. Contienen información relevante como responsables, fechas límite y descripciones.
- Columnas: Cada columna representa una etapa del proceso, las tareas se mueven de una columna a otra a medida que avanzan.
- Límites de trabajo en curso: La regla la cual restringe el número de tareas en una columna para evitar la saturación y mejorar el flujo.

1.6. Estudio económico y presupuesto

Para garantizar la viabilidad del proyecto **CineChamp**, es fundamental realizar un análisis de costos detallado, abarcando tanto el desarrollo como el mantenimiento de la plataforma. A continuación, se desglosan los principales gastos estimados:

Tabla 1.1. Costo de desarrollo

Concepto	Costo
Hosting y Dominio 1 año	~100 € (plan anual de hosting + dominio)
Desarrollo web (Frontend y Backend)	~ 4000 € (Sitio web personalizada)
Base de datos y Servidores	~ 60 €
Licencia de software	~ 50 €
Seguridad y Certificación SSL	~ 0 € (Certificado SSL gratuito)
Integración de APIs (como IMDb, TMDb)	~ 0 € (API de TMDb gratuita)
Diseño UI/UX	~ 1000 € (diseño de la interfaz y experiencia de usuario)
Total de desarrollo inicial estimada	~ 5210

Fuente: **[Costo Inicial]**

Tabla 1.2 Costo de mantenimiento (Anuales)

Concepto	Costo
Mantenimiento del Servidor	~600 € (administración servidor, ~50 €/mes)
Soporte y Actualizaciones	~600 € (soporte técnico y actualizaciones, ~50 €/mes)
Renovación de Dominio y Hosting	~100 € (renovación anual del dominio + hosting)
Seguridad y Monitoreo de la Plataforma	~200 € (servicio de seguridad adicional, ej. Cloudflare Pro)
Total del mantenimiento anual estimada	~ 1550

Fuente: **[Costo Inicial]**

Capítulo II

Descripción del proyecto

2. Descripción del proyecto

El proyecto **CineChamp** es una aplicación web dirigida a los amantes del cine y las series, cuyo objetivo principal es permitir la gestión de un historial audiovisual, el intercambio de opiniones y la creación de una comunidad activa a través de logros y reseñas. La plataforma permitirá a los usuarios registrar las películas y series vistas, puntuar y reseñar contenido, y mostrar estas valoraciones en su perfil personal. Además, facilitará el descubrimiento de nuevas producciones mediante las recomendaciones y experiencias compartidas por la comunidad.

2.1. Análisis de requisitos

Para considerar finalizado el desarrollo del proyecto, la plataforma debe cumplir una serie de requisitos que garanticen su funcionamiento, calidad y usabilidad. Estos requisitos engloban tanto las funcionalidades principales que debe ofrecer a los usuarios, como las características de rendimiento, seguridad, escalabilidad y experiencia de usuario que aseguren su viabilidad y sostenibilidad a largo plazo.

En términos generales, el sistema deberá cumplir las siguientes acciones:

- Permitir la creación y gestión de perfiles de usuario.
- Ofrecer herramientas para puntuar y reseñar películas o series.
- Mostrar el historial y las valoraciones de cada usuario en su perfil personal.
- Incluir un sistema de logros que motive la participación activa de la comunidad.
- Permitir la interacción social entre usuarios mediante comentarios y la posibilidad de agregar amigos.
- Sistema de niveles que motive al usuario a seguir usando la plataforma, compitiendo con sus amigos u otros usuarios.
- Garantizar la protección de los datos personales y la seguridad de la plataforma.
- Presentar una interfaz intuitiva y accesible, compatible con los principales navegadores web.

2.1.1. Requisitos funcionales

A continuación se detallan los requisitos funcionales que se debe implementar para considerarse completada.

1. Registro y autenticación de usuarios:

- Permitir que los usuarios se registren mediante correo electrónico y contraseña.
- Permitir el inicio de sesión y cierre de sesión de forma segura.

2. Gestión de perfil de usuario:

- Permitir a los usuarios editar su información personal (nombre de usuario y avatar)
- Permitir visualizar el perfil propio y el de otros usuarios

3. Búsqueda y consulta de contenido audiovisual:

- Permitir buscar películas y series mediante títulos o palabras clave.
- Permitir consultar información básica de las películas y series (sinopsis, año, género, valoración).

4. Gestión de historial de contenido:

- Permitir añadir películas y series al historial personal como vistas o pendientes.
- Permitir eliminar contenido del historial.
- Permitir marcar películas y series como favoritas.

5. Sistema de puntuación y reseñas:

- Permitir a los usuarios puntuar las películas y series que han visto.
- Permite escribir, publicar y editar reseñas de películas y series.
- Permitir visualizar las reseñas y puntuaciones de otros usuarios.

6. Sistema de logros y estadísticas:

- Asignar logros automáticos al usuario al cumplir determinadas condiciones (ejemplo: ver 10 películas de terror).
- Mostrar estadísticas personales del usuario (número de películas vistas, géneros más vistos).

7. Interacción social:

- Permitir agregar y eliminar amigos en la plataforma.
- Permitir ver el historial y logros de los amigos.
- Permitir comentar en las reseñas de otros usuarios.

2.1.2. Requisitos no funcionales

Los requisitos no funcionales establecen los criterios de calidad que debe cumplir la plataforma **CineChamp** para garantizar una experiencia de usuario sólida, segura y eficiente. Estos requisitos complementan las funcionalidades principales y aseguran que el sistema sea viable y sostenible a largo plazo.

1. Facilidad de uso:

La plataforma debe disponer de una interfaz intuitiva, accesible y amigable, orientada a usuarios sin conocimientos técnicos, facilitando la navegación y la interacción con las funcionalidades clave.

2. Escalabilidad:

La arquitectura del sistema debe estar preparada para permitir la incorporación de nuevas funcionalidades y soportar un aumento progresivo del número de usuarios sin comprometer el rendimiento.

3. Compatibilidad:

La aplicación web debe ser completamente funcional en los navegadores más utilizados, como Google Chrome, Mozilla Firefox, Safari y Microsoft Edge, asegurando una experiencia consistente en distintos entornos.

2.2. Estudios previos del desarrollo

Durante el desarrollo del proyecto CineChamp, se llevarán a cabo distintas tareas de investigación orientadas a profundizar en el uso óptimo de las tecnologías seleccionadas, garantizar la integración eficaz de los distintos componentes y asegurar el cumplimiento de los objetivos funcionales y no funcionales de la plataforma. A continuación, se enumeran las principales tareas de investigación previstas:

1. Investigación de mejores prácticas en el uso de React con TypeScript y Vite, con el fin de desarrollar una interfaz de usuario eficiente, escalable y mantenible.
2. Análisis de patrones y arquitecturas recomendadas para la implementación de APIs REST con Node.js y Express, garantizando una estructura limpia y modular.
3. Estudio de técnicas de integración entre el backend (Node.js + Express) y la base de datos MySQL, incluyendo el uso de ORM.
4. Investigación de la integración de la API pública The Movie Database (TMDb) en el backend, considerando el manejo de claves API, límites de uso y almacenamiento en caché de resultados.
5. Análisis de estrategias de seguridad en aplicaciones web full stack, incluyendo el cifrado de contraseñas (por ejemplo, con bcrypt), protección contra ataques de inyección SQL.
6. Investigación sobre la implementación de un sistema de logros basado en la actividad de los usuarios, explorando posibles estructuras de datos y lógica de negocio en el backend.
7. Estudio de las opciones para enviar notificaciones dentro de la plataforma (notificaciones internas o vía correo electrónico) y su integración con Node.js.
8. Investigación de herramientas de testing para frontend (por ejemplo, React Testing Library) y backend (por ejemplo, Jest o Supertest) para garantizar la calidad y fiabilidad del sistema.
9. Análisis de las obligaciones legales relacionadas con la protección de datos personales (Reglamento General de Protección de Datos, RGPD) y su aplicación en la plataforma.
10. Investigación de técnicas y herramientas de control de versiones y colaboración en el desarrollo, evaluando opciones como GitHub, GitLab o Bitbucket.
11. Estudio de servicios de despliegue y alojamiento adecuados para una aplicación full stack, evaluando opciones como Render, Heroku, Vercel o servidores propios en la nube.

2.3. Tecnologías

Durante el desarrollo del proyecto CineChamp, se valoraron diferentes tecnologías tanto para el frontend como para el backend y la gestión de la base de datos. A continuación, se realiza una comparativa de las principales tecnologías consideradas y se justifican las decisiones tomadas respecto a las tecnologías finalmente escogidas.

2.3.1. Comparativa de las Tecnologías Valoradas

A lo largo del proceso de análisis y diseño del proyecto se estudiaron varias alternativas tecnológicas para cada uno de los componentes principales de la plataforma. La siguiente comparativa resume las opciones más relevantes evaluadas:

1. Frontend se valoraron las siguientes tecnologías:

React es una biblioteca JavaScript muy popular, basada en componentes, con una gran comunidad de soporte y ecosistema de herramientas. Facilita el desarrollo de interfaces dinámicas y escalables. Entre sus ventajas se encuentran su alta popularidad, la facilidad de integración con TypeScript y el rendimiento optimizado mediante virtual DOM. Como desventaja, presenta una curva de aprendizaje inicial si se combinan varias librerías externas.

Angular es un framework completo que incluye una estructura más rígida y herramientas integradas por defecto. Sus ventajas son la estructura predefinida y el soporte completo desde el propio framework. Su principal desventaja es la mayor complejidad inicial y el mayor peso de las aplicaciones.

Vue.js es un framework progresivo que permite ir escalando en complejidad. Entre sus ventajas destaca la facilidad de aprendizaje y una documentación clara. Como desventaja, presenta menor popularidad en grandes proyectos internacionales.

2. Herramientas de construcción (bundler): Se compararon Vite y Webpack.

Vite es una herramienta moderna de construcción y desarrollo optimizada para frameworks actuales. Sus ventajas son los tiempos de arranque rápidos y un hot module replacement eficiente. Su desventaja es el menor soporte de plugins en comparación con Webpack.

Webpack es una herramienta ampliamente adoptada y personalizable. Ofrece como ventaja su gran cantidad de plugins y configuración detallada. Su desventaja es la configuración más compleja y los tiempos de compilación más largos.

3. Backend: Se evaluaron Node.js con Express, Django y Spring Boot.

Node.js con Express es un entorno de ejecución JavaScript con un framework ligero y flexible para construir APIs REST. Sus ventajas incluyen la compatibilidad con JavaScript en

todo el stack, la alta velocidad de desarrollo y un gran ecosistema de librerías. Su principal desventaja es la necesidad de controlar la asíncronía y la escalabilidad manualmente.

Django (Python) es un framework completo que incluye ORM y herramientas integradas. Sus ventajas son la autenticación y el ORM integrado, además de rapidez en el desarrollo backend. Su desventaja es estar menos alineado con JavaScript y tener un mayor consumo de recursos.

Spring Boot (Java) es un framework robusto preparado para grandes proyectos empresariales. Su ventaja principal es la escalabilidad y la robustez empresarial. Su desventaja es la mayor complejidad y una curva de aprendizaje elevada.

4. Base de datos: Se analizaron MySQL y MongoDB.

MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado. Sus ventajas son la fiabilidad, el soporte de transacciones y las consultas estructuradas. Su desventaja es ser menos flexible para estructuras no relacionales.

MongoDB es una base de datos no relacional basada en documentos. Sus ventajas incluyen la alta flexibilidad de esquemas y la escalabilidad horizontal. Su principal desventaja es la menor consistencia de datos en operaciones transaccionales complejas.

2.3.2. Tecnologías escogidas

Tras el análisis comparativo de las diferentes tecnologías valoradas, se seleccionaron aquellas que mejor se ajustaban a las necesidades funcionales y no funcionales del proyecto **CineChamp**, considerando criterios como rendimiento, escalabilidad, compatibilidad, experiencia previa del equipo de desarrollo y recursos disponibles.

Para el desarrollo del frontend se escogió React junto con TypeScript y Vite. React fue elegido por su flexibilidad y el enfoque basado en componentes, lo que facilita la construcción de interfaces dinámicas y reutilizables. TypeScript aporta ventajas como el tipado estático, la detección temprana de errores y una mayor mantenibilidad del código. Por su parte, Vite se seleccionó como herramienta de construcción y desarrollo por su rapidez en la compilación y la recarga en caliente, lo que mejora la productividad y la experiencia de desarrollo.

En el backend se optó por Node.js con el framework Express. Esta combinación permite trabajar con JavaScript en todo el stack de la aplicación, lo que simplifica la integración entre frontend y backend. Express facilita la creación de una API REST de manera ligera y modular, ofreciendo una curva de aprendizaje asequible y una amplia comunidad de soporte.

Para la gestión de datos se eligió MySQL como sistema de gestión de bases de datos relacional. La elección se fundamenta en su fiabilidad, su soporte de transacciones y su compatibilidad con aplicaciones web basadas en Node.js. MySQL ofrece una estructura de datos adecuada para modelar las relaciones entre usuarios, películas, series, reseñas y

logros, manteniendo la integridad referencial y permitiendo consultas estructuradas y eficientes.

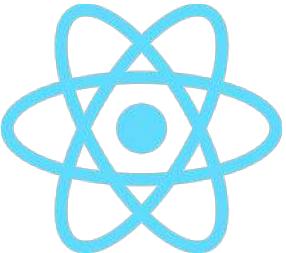
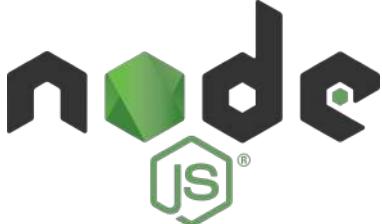
La selección de estas tecnologías responde al equilibrio entre rendimiento, facilidad de desarrollo, escalabilidad futura y disponibilidad de recursos de aprendizaje y soporte. Además, permite garantizar la compatibilidad entre los distintos componentes del proyecto, facilitando el mantenimiento y las posibles ampliaciones de la plataforma en el futuro.

2.4. Estructura del proyecto

La plataforma **CineChamp** se basa en una arquitectura de tres capas: frontend, backend y base de datos, complementada con la integración de una API externa para enriquecer los datos audiovisuales. Esta estructura permite una separación clara de responsabilidades y facilita la escalabilidad del sistema.

A continuación, se presenta un esquema general de la estructura tecnológica del proyecto:

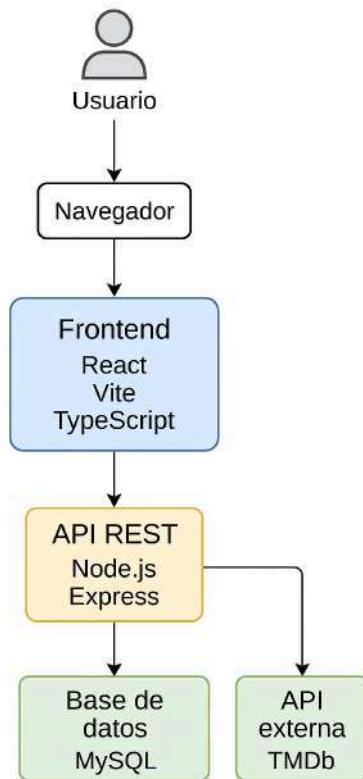
Tabla 2.1. Tecnologías utilizadas

Componente	Tecnología	Descripción breve
	React + TypeScript	Desarrollo del frontend, gestión de la interfaz de usuario.
	Node.js	Entorno de ejecución para el backend
	Express	Framework para definir la lógica del backend y las rutas de la API.

	MySQL	Base de datos relacional donde se almacena la información estructurada
	TMDB API	Fuente externa de información audiovisual.

Fuente: **Elaboración propia**

Figura 2.1. Estructura general del proyecto



Fuente: **Elaboración propia**

2.5. Descripción de los componentes

El sistema CineChamp se compone de cuatro grandes bloques funcionales: la interfaz de usuario (frontend), la API de servidor (backend), la base de datos relacional y la integración

con una API externa. A continuación, se describen de forma individual y detallada cada uno de estos componentes, sus funcionalidades internas, y las tecnologías utilizadas.

2.5.1. Frontend

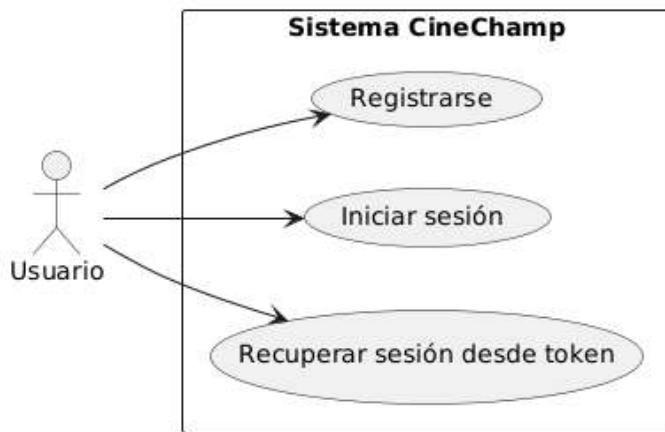
El frontend del proyecto está desarrollado utilizando React con TypeScript, gestionado mediante Vite como herramienta de desarrollo. Se encarga de la interfaz visual del usuario, la navegación entre páginas y la comunicación con la API REST del backend.

La estructura del código se organiza en carpetas como components, que contiene módulos reutilizables (por ejemplo, Avatar, ModalPuntuacion, Header, Carrusel), y pages, donde se definen las vistas principales como Perfil.tsx, PaginaPelicula.tsx, Buscador.tsx y ListaContenido.tsx. También se incluyen carpetas específicas para la autenticación (auth) y secciones privadas (private).

2.5.1.1 Registro y autenticación de usuarios

La funcionalidad de registro y autenticación permite a los usuarios crear una cuenta en CineChamp y acceder a su perfil privado. Es el punto de entrada principal para interactuar con la plataforma. Esta funcionalidad garantiza que cada usuario tenga un espacio único y seguro donde gestionar su actividad, contenido y relaciones sociales.

Figura 2.2. Diagrama de casos de uso Registro y Login



Fuente: **Elaboración propia**

Proceso conceptual

Registro:

1. El usuario accede al formulario de registro (Register.tsx).
2. Introduce su nickname, correo electrónico y contraseña.
3. El frontend valida los datos y los envía al backend vía API.
4. El backend:
 - a. Verifica que el correo y el nick no estén registrados.
 - b. Encripta la contraseña con bcrypt.

- c. Crea un nuevo usuario en la base de datos.
- d. Devuelve un mensaje de éxito o error.

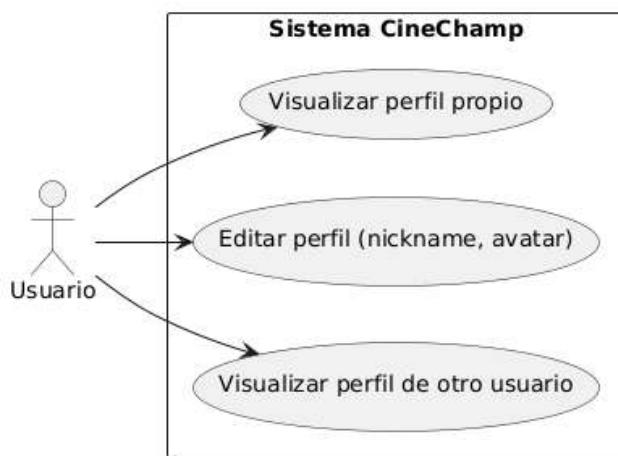
Autenticación (login):

1. El usuario accede al formulario de inicio de sesión (Login.tsx).
2. Introduce su correo y contraseña.
3. El backend
 - a. Verifica si el correo existe.
 - b. Compara la contraseña con la almacenada (encriptada).
 - c. Si es correcta, genera un JWT.
 - d. Devuelve el token al frontend, que lo almacena en localStorage.

2.5.1.2. Gestión del perfil usuario

La funcionalidad de gestión del perfil de usuario permite a cada persona registrada modificar su información personal, visualizar su historial de actividad, logros, favoritos y gestionar su avatar. Este componente refuerza la personalización y el sentido de pertenencia dentro de la plataforma.

Figura 2.3. Diagrama de casos de uso Editar Perfil



Fuente: **Elaboración propia**

Proceso conceptual

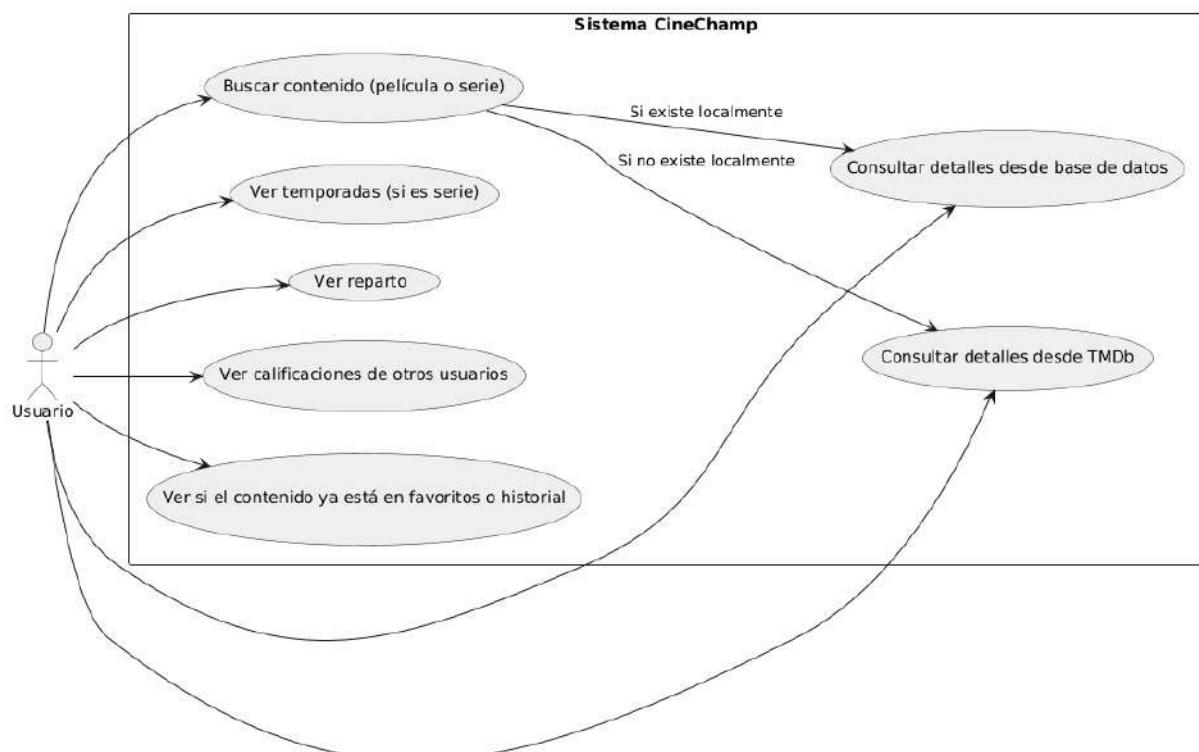
1. Visualización de perfil personal:
 - a. El usuario accede a /perfil.
 - b. Se muestra su información (nick, email, avatar), su actividad (historial de películas y series), logros obtenidos, favoritos y últimas calificaciones.
2. Edición de perfil:
 - a. Desde la sección del perfil, el usuario accede a EditarPerfil.tsx.

- b. Puede actualizar su nickname y subir una nueva imagen de perfil.
 - c. La imagen se carga en el servidor mediante el middleware multerConfig.js.
3. Visualización del perfil de otro usuario:
- a. Desde la lista de amigos o resultados de búsqueda, el usuario accede a /perfil-publico/:id.
 - b. Se muestran los datos públicos del usuario: avatar, logros visibles, actividad y amistades comunes.
4. Restricciones:
- a. Cada usuario solo puede modificar su propio perfil.
 - b. Si se accede a otro perfil, se deshabilita la opción de edición y se ocultan datos privados.

2.5.1.3. Búsqueda y consulta de contenido

Esta funcionalidad permite a los usuarios buscar películas y series dentro de la plataforma, accediendo a contenido tanto almacenado localmente como desde la API externa de The Movie Database (TMDb). También permite consultar en detalle cada título, visualizando información enriquecida como sinopsis, año de estreno, reparto, temporadas (en el caso de series) y reseñas de otros usuarios.

Figura 2.4. Diagrama de casos de uso Búsqueda de película



Fuente: **Elaboración propia**

Proceso conceptual

1. Búsqueda:
 - a. El usuario introduce una palabra clave en el BuscadorUnificado.
 - b. Se envía una petición al backend para consultar tanto la base de datos local como resultados en TMDb.
 - c. Se presentan los resultados agrupados por tipo (películas o series), con opciones para visualizar los detalles.
2. Visualización de detalles de contenido:
 - a. Al hacer clic en un resultado, se accede a PaginaPelicula.tsx.
 - b. El sistema muestra:
 - i. Título, sinopsis, fecha de estreno.
 - ii. Botón para agregar al historial.
 - iii. Información del reparto (RepartoContenido.tsx) si está disponible.
 - iv. Temporadas (TemporadasContenido.tsx) en caso de que se trate de una serie.
 - v. Calificaciones de usuarios.
3. Integración con TMDb:
 - a. Se utiliza el identificador TMDb para obtener datos actualizados sin almacenar información redundante.
 - i. El backend se encarga de comunicar con la API de TMDb, aplicar caché si es necesario, y servir los datos al frontend.

2.5.1.4. Valoración y reseña de contenido

Los usuarios autenticados pueden valorar películas o series y escribir una reseña personal para compartir su opinión con la comunidad. Esta funcionalidad tiene dos objetivos principales:

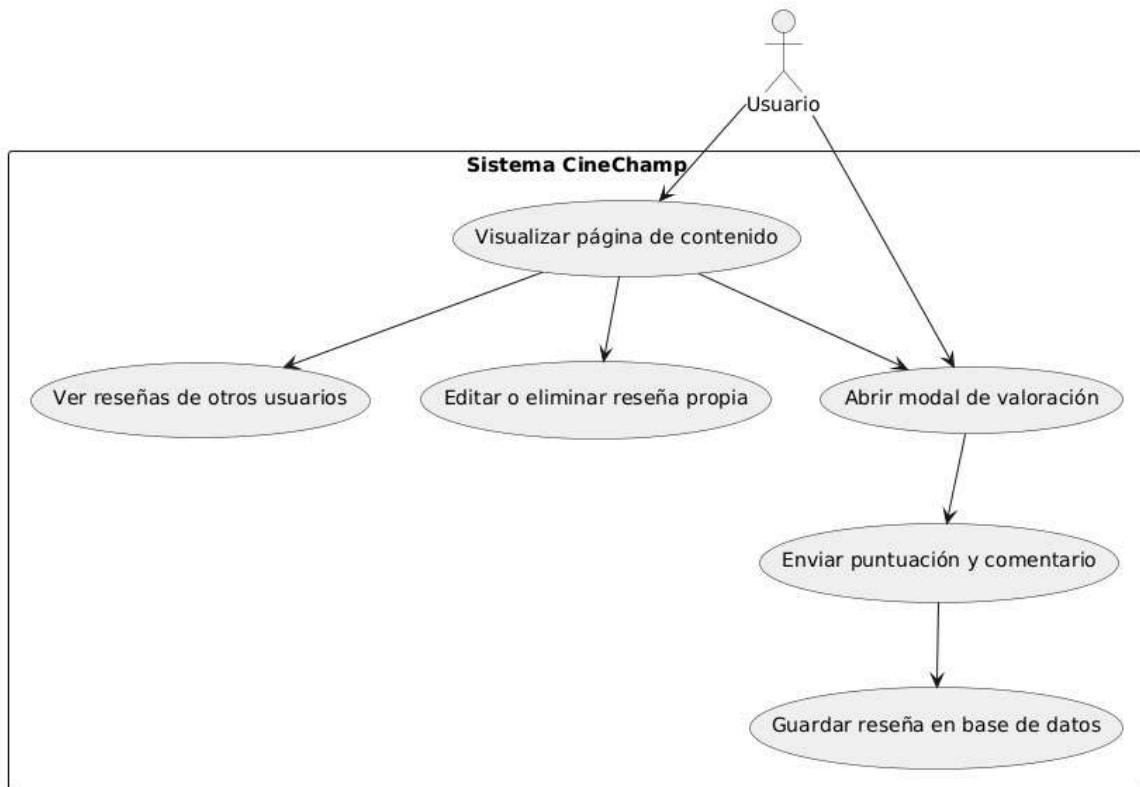
- Guardar la puntuación y comentario del usuario en la base de datos (tabla calificación).
- Mostrar la opinión agregada del resto de usuarios en la ficha de cada contenido, mediante una media aritmética.

Desde la vista PaginaPelicula.tsx, se habilita un botón de "Valorar" que abre un modal (ModalPuntuacion.tsx) donde el usuario puede seleccionar una puntuación del 1 al 10 y escribir un comentario. Este dato se envía al backend, que valida y almacena la información en la tabla calificación.

Una vez registrado, el contenido se actualiza dinámicamente y se muestra la reseña publicada por el usuario, junto con un listado de otras valoraciones (vista en ReseñaPorUsuarios.tsx y UltimasCalificaciones.tsx).

También se comprueba si el usuario ya ha valorado ese contenido, en cuyo caso se le ofrece editar o eliminar su reseña.

Figura 2.5. Diagrama de casos de uso Valoración de películas o series



Fuente: **Elaboración propia**

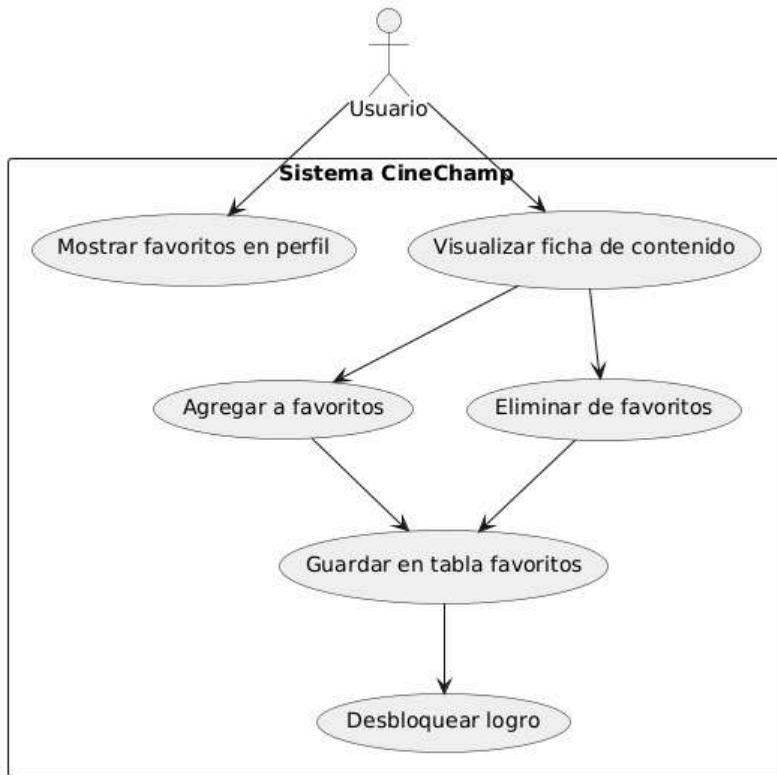
2.5.1.5. Sistema de favoritos

La plataforma permite a los usuarios marcar películas o series como favoritas, facilitando un acceso rápido a sus contenidos más apreciados desde su perfil. Esta funcionalidad se encuentra disponible tanto en las fichas de contenido como en los resultados de búsqueda.

Cuando el usuario hace clic en el botón de "Aregar a favoritos", se realiza una petición al backend para registrar el id_tmdb del contenido, junto con el id_usuario. Esta información se almacena en la tabla favoritos. Si ya se ha marcado como favorito, se ofrece la opción de eliminarlo.

En el perfil personal (Perfil.tsx), se muestra una sección exclusiva donde se listan los favoritos del usuario separados por tipo: películas y series. Esta vista utiliza componentes reutilizables de Carrusel.tsx.

Figura 2.6. Diagrama de casos de uso Sistema de Favoritos



Fuente: **Elaboración propia**

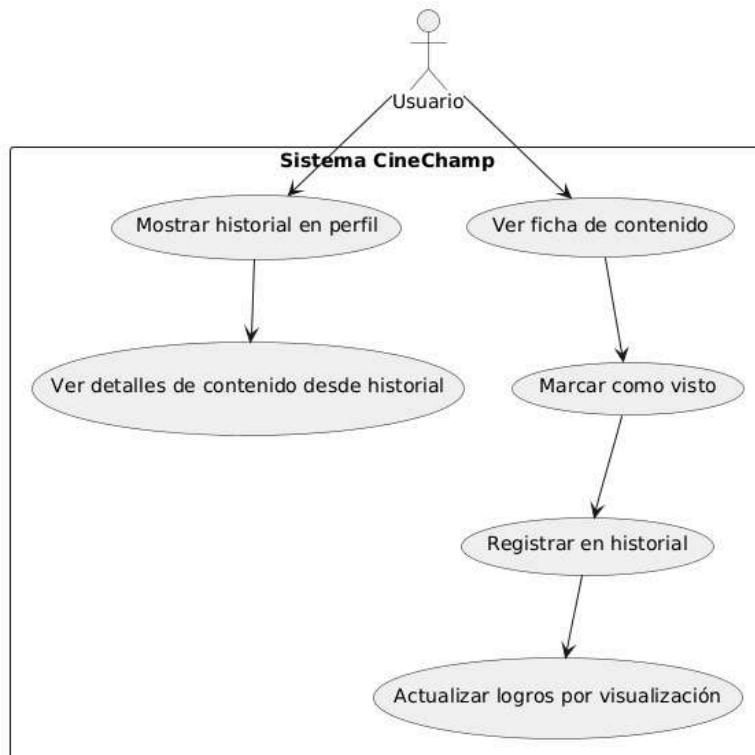
2.5.1.6. Visualización de historial de contenido (Películas y Series)

CineChamp permite a los usuarios llevar un registro personal de los contenidos que han visto. Desde cada ficha de contenido (película o serie), pueden marcar como "vista", lo cual añade ese ítem al historial del usuario. Este historial se presenta en su perfil personal, dividido por tipo de contenido:

- Historial de Películas
- Historial de Series

Cada entrada incluye título, imagen, fecha de visualización, y puntuación si la ha realizado.

Figura 2.7. Diagrama de casos de uso Historial de Contenido



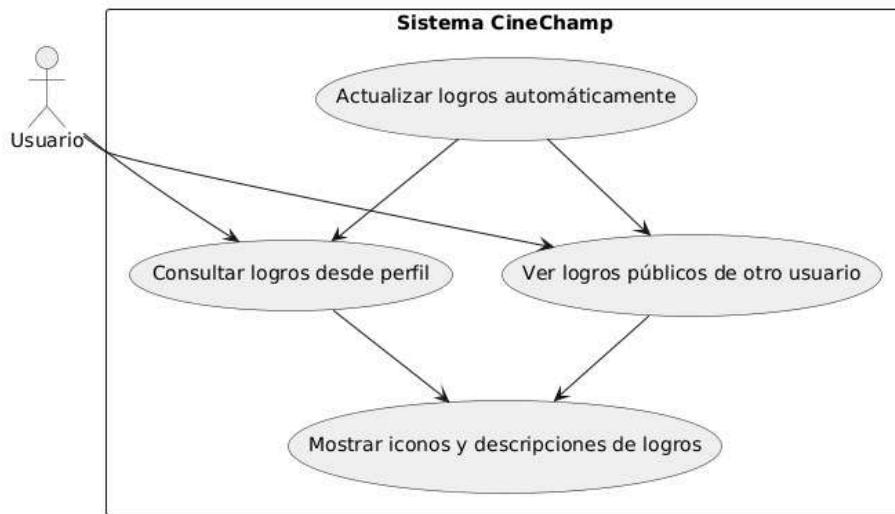
Fuente: **Elaboración propia**

2.5.1.7. Visualización de logros

El sistema de logros en CineChamp motiva a los usuarios premiando su actividad dentro de la plataforma. Existen diferentes tipos de logros: por número de películas vistas, por ser de los primeros en registrarse, por uso de favoritos, entre otros.

Desde el perfil personal, los usuarios pueden visualizar los logros obtenidos junto con sus respectivos nombres, descripciones e íconos ilustrativos. Además, los logros también se muestran en los perfiles públicos, fomentando la gamificación y el reconocimiento entre la comunidad.

Figura 2.8. Diagrama de casos de uso Visualización de Logros obtenidos



Fuente: **Elaboración propia**

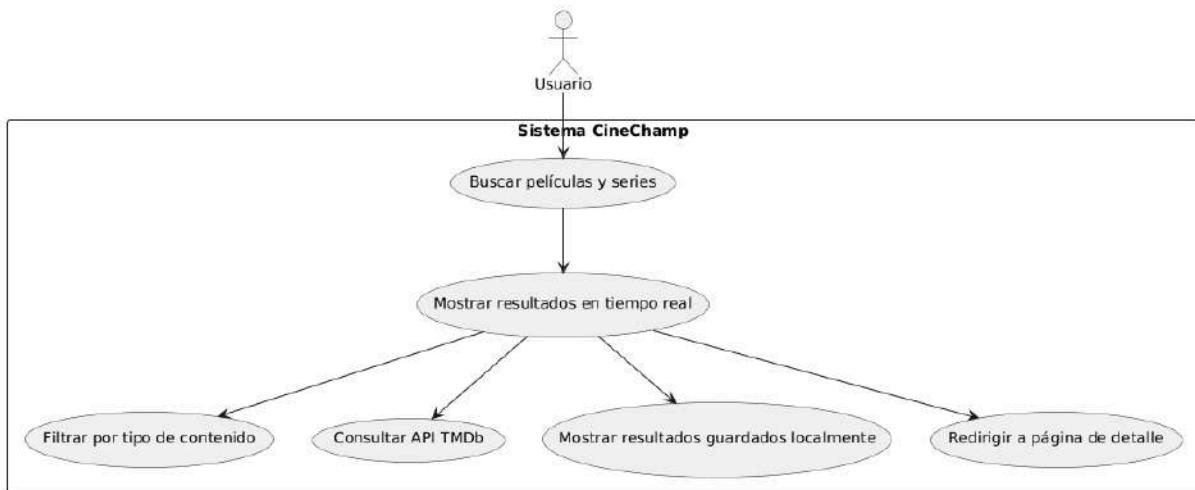
2.5.1.8 Búsqueda unificada de contenido

La plataforma **CineChamp** permite a los usuarios buscar películas y series mediante un buscador unificado. Esta funcionalidad es clave para mejorar la experiencia de usuario, ya que ofrece resultados inmediatos que combinan:

- Contenido proveniente de la base de datos local (contenido guardado previamente).
- Resultados en tiempo real desde la API de TMDb, utilizando el texto ingresado por el usuario.

Esto permite una búsqueda rápida, eficiente y enriquecida con información visual, títulos y sinopsis actualizados.

Figura 2.9. Diagrama de casos de uso Búsqueda de Contenido agregado



Fuente: **Elaboración propia**

2.5.1.9. Detalle de la película o serie

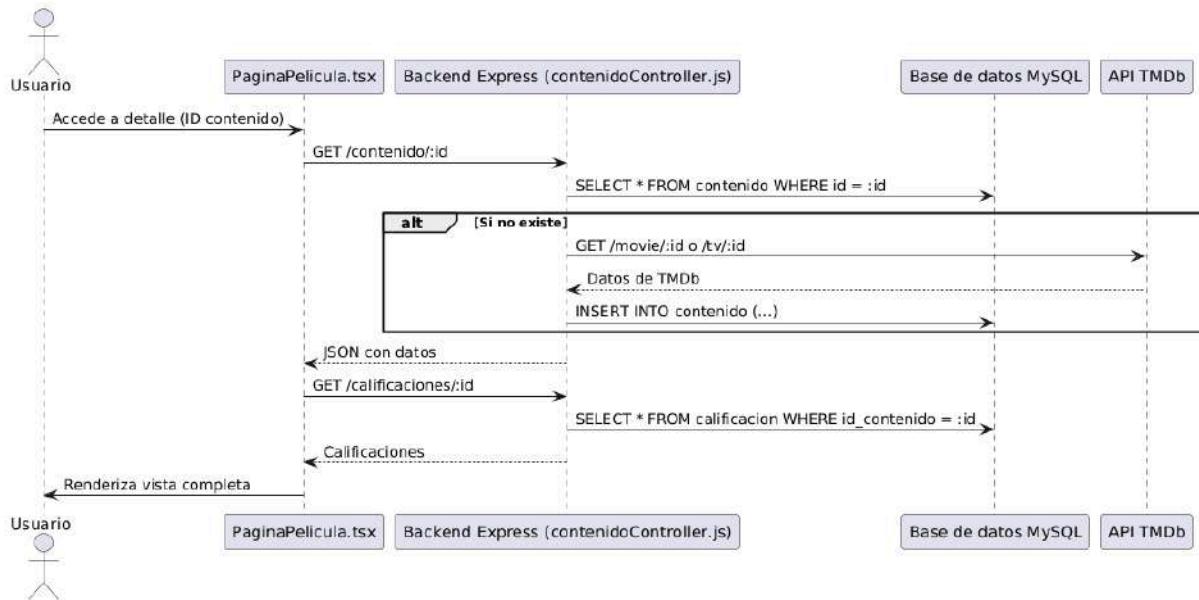
Esta funcionalidad permite al usuario consultar la información detallada de una película o serie concreta. Esta página se muestra al hacer clic en un resultado de búsqueda o en cualquier componente que enlaza a un contenido específico (como favoritos, historial).

Cuando el usuario accede al detalle, se muestra:

- El título, año de estreno, sinopsis y poster.
- La puntuación media obtenida por otros usuarios.
- Botones para:
 - Añadir al historial.
 - Agregar a favoritos.
 - Calificar con comentario.
- Secciones adicionales como:
 - Temporadas si es una serie.
 - Reparto (extraído de la API de TMDb).

La lógica en el backend comprueba si ese contenido ya fue guardado en la base de datos. Si no, lo registra al vuelo con el ID de TMDb y se recupera la información mediante la API externa.

Figura 2.10. Diagrama de secuencia Detalle de contenido



Fuente: **Elaboración propia**

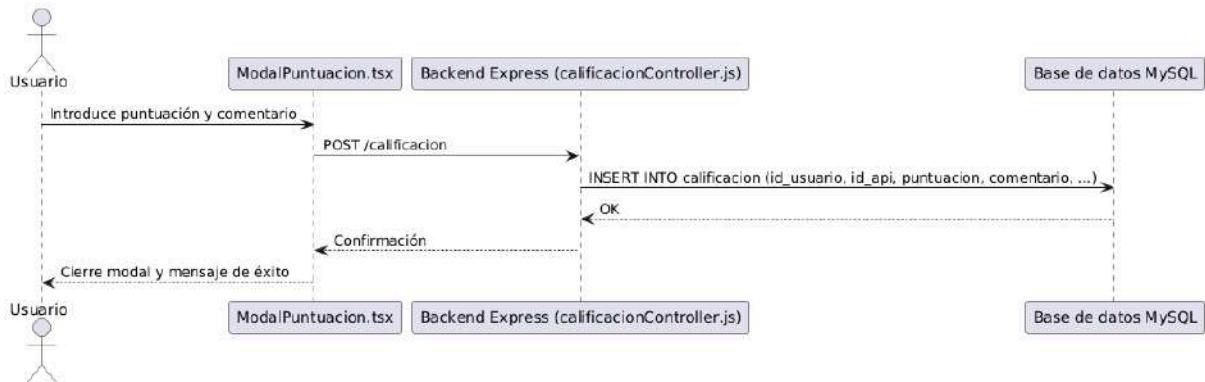
2.5.1.10. Calificación de contenido

Los usuarios pueden valorar películas o series a través de un sistema de puntuación del 1 al 10, acompañando (opcionalmente) su voto con un comentario.

El sistema contempla lo siguiente:

- Cada usuario solo puede calificar una vez cada contenido.
- La calificación se guarda junto con:
 - ID del contenido (de TMDB).
 - Tipo (película o serie).
 - Fecha.
 - Comentario.
- Los datos se muestran en:
 - El detalle del contenido.
 - El historial de calificaciones del usuario.
 - La lista de calificaciones recientes en los perfiles públicos.

Figura 2.11. Diagrama de secuencia Calificación de contenido



Fuente: Elaboración propia

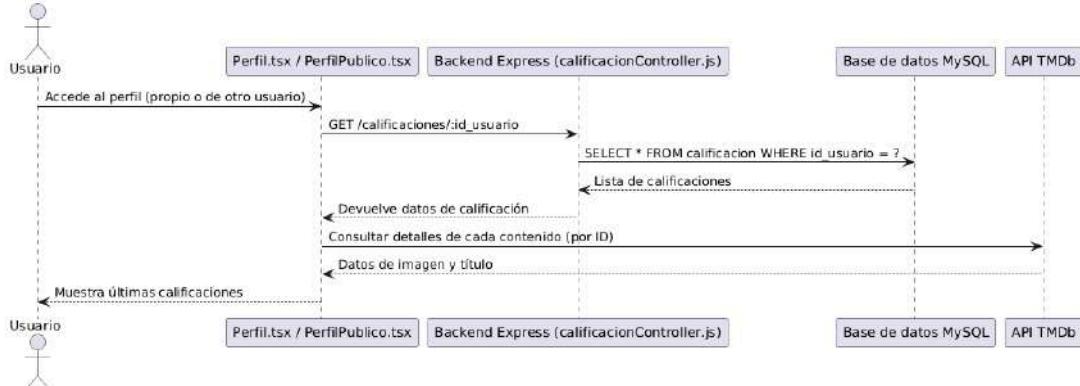
2.5.1.11. Visualización de últimas calificaciones

Esta funcionalidad permite mostrar, tanto en el perfil público como en el perfil propio, las últimas calificaciones realizadas por un usuario. Cada entrada incluye:

- Miniatura del contenido (consultado desde TMDb).
- Título del contenido.
- Puntuación asignada (1–10).
- Fecha de la calificación.
- Comentario (si se proporcionó).
- Tipo de contenido (película o serie).

Esta característica permite a los demás usuarios conocer los gustos recientes de otros y fomenta la interacción social dentro de la plataforma.

Figura 2.12. Diagrama de secuencia Visualización de últimas calificaciones



Fuente: Elaboración propia

2.5.2. Backend

El backend de **CineChamp** está desarrollado en Node.js con el framework Express, lo que permite una arquitectura modular basada en controladores, modelos y rutas. Esta capa de la aplicación se encarga de procesar la lógica del negocio, servir como intermediario entre el frontend y la base de datos, gestionar las peticiones HTTP, y comunicarse con servicios externos como TMDb (The Movie Database).

A lo largo del desarrollo, se ha aplicado el principio de separación de responsabilidades, dividiendo la lógica en carpetas como: controllers, models, routes y middlewares, lo que facilita la escalabilidad y el mantenimiento del proyecto. El servidor está configurado para escuchar en el puerto 3001 y permite peticiones desde el frontend.

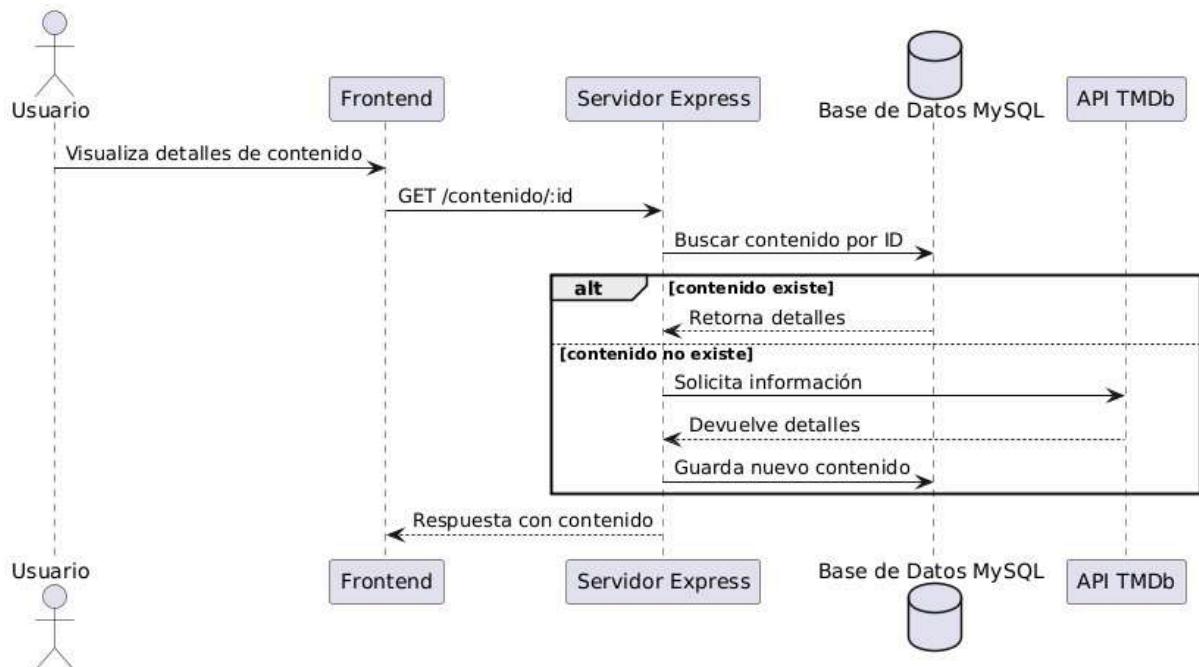
A continuación, se detallan los principales componentes y responsabilidades del backend:

- Controladores (controllers/): contienen la lógica principal para gestionar usuarios, contenido, calificaciones, relaciones sociales (amistades), y logros. Cada uno responde a distintas rutas expuestas al cliente, ejecutando validaciones, consultas y retornando respuestas con formato JSON.
- Modelos (models/): encapsulan las interacciones con la base de datos MySQL. Estos archivos utilizan consultas SQL para insertar, recuperar o modificar datos relacionados con usuarios, contenido audiovisual, calificaciones y más.
- Rutas (routes/) : definen los endpoints RESTful que el frontend puede consumir. Cada ruta está asociada a un controlador específico, y se agrupan por temática (usuarios, logros, contenido).
- Configuración y utilidades: el backend incluye un archivo db.js para establecer la conexión con MySQL, y configuraciones adicionales como multerConfig.js para la gestión de subida de imágenes. También se aplica lógica para el sistema de logros, como en logros/index.js.

2.5.2.1. Gestión de contenido

La gestión del contenido en **CineChamp** es una de las funcionalidades centrales del backend. Se encarga de administrar y exponer los datos relacionados con películas y series, ya sea para mostrarlos en la plataforma o para ser utilizados por otras funcionalidades (como calificaciones o favoritos). La lógica del backend se estructura siguiendo el patrón Modelo-Vista-Controlador (MVC), lo que facilita la modularidad y el mantenimiento del código.

Figura 2.13. Diagrama de secuencia Gestión de contenido



Fuente: **Elaboración propia**

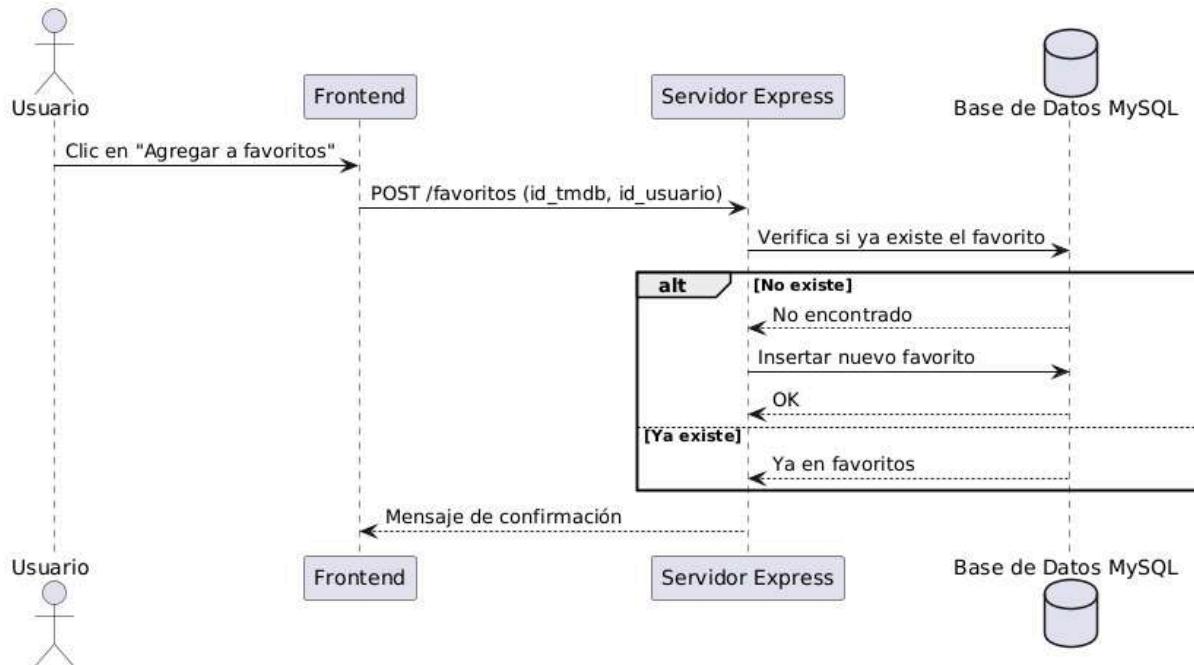
2.5.2.2. Gestión de favoritos

La funcionalidad de favoritos permite a los usuarios marcar películas o series como contenido preferido dentro de CineChamp. Esta opción no solo mejora la experiencia del usuario, sino que también posibilita una mayor personalización del contenido visible en su perfil.

Descripción funcional:

- Los usuarios pueden agregar películas o series a su lista personal de favoritos.
- Permite consultar todos los favoritos de un usuario.
- Implementa una lógica de control para evitar duplicados mediante restricciones únicas en la base de datos.
- La información de los favoritos se utiliza también para mostrar estadísticas o recomendaciones personalizadas en el perfil del usuario.

Figura 2.14. Diagrama de secuencia Gestión de favoritos



Fuente: **Elaboración propia**

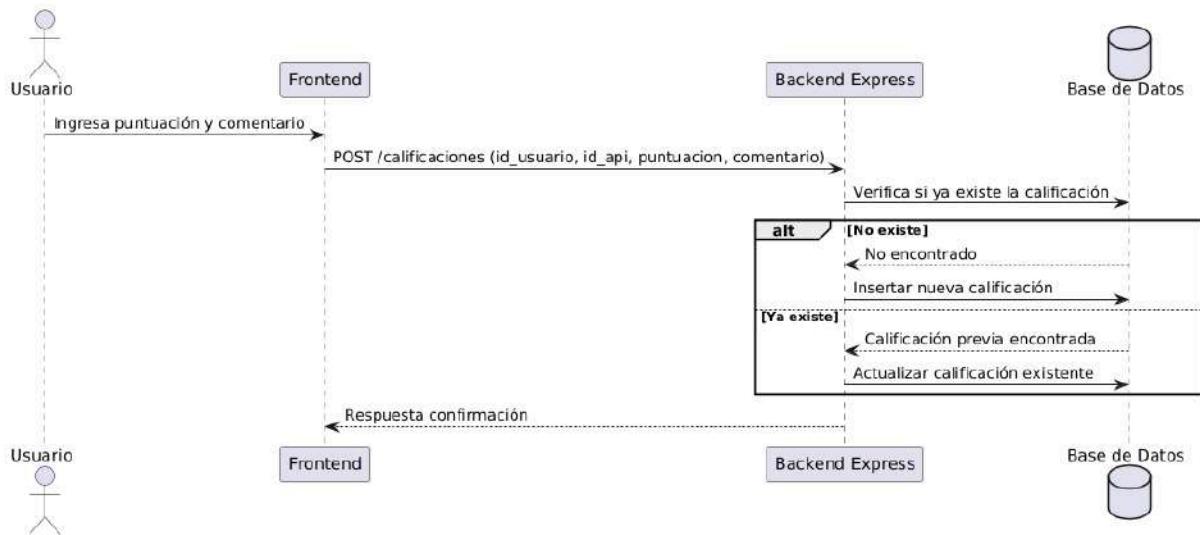
2.5.2.3 Gestión de calificaciones

El sistema de calificaciones permite a los usuarios valorar el contenido (películas y series) según su experiencia personal. Esta funcionalidad es clave para generar un promedio de puntuación que se refleja en las fichas del contenido, y que además contribuye a desbloquear logros según la actividad del usuario.

Descripción funcional:

- Los usuarios pueden valorar con una puntuación del 1 al 10.
- Pueden dejar un comentario opcional junto a su valoración.
- Cada usuario solo puede calificar una vez cada película o serie.
- Las calificaciones se asocian directamente al id_tmdb para evitar duplicación de contenido externo.
- Se utiliza esta información para:
 - Mostrar la nota media.
 - Mostrar comentarios recientes en el perfil del contenido.
 - Generar logros automáticos al alcanzar ciertos umbrales de actividad.

Figura 2.15. Diagrama de secuencia Gestión de calificaciones



Fuente: **Elaboración propia**

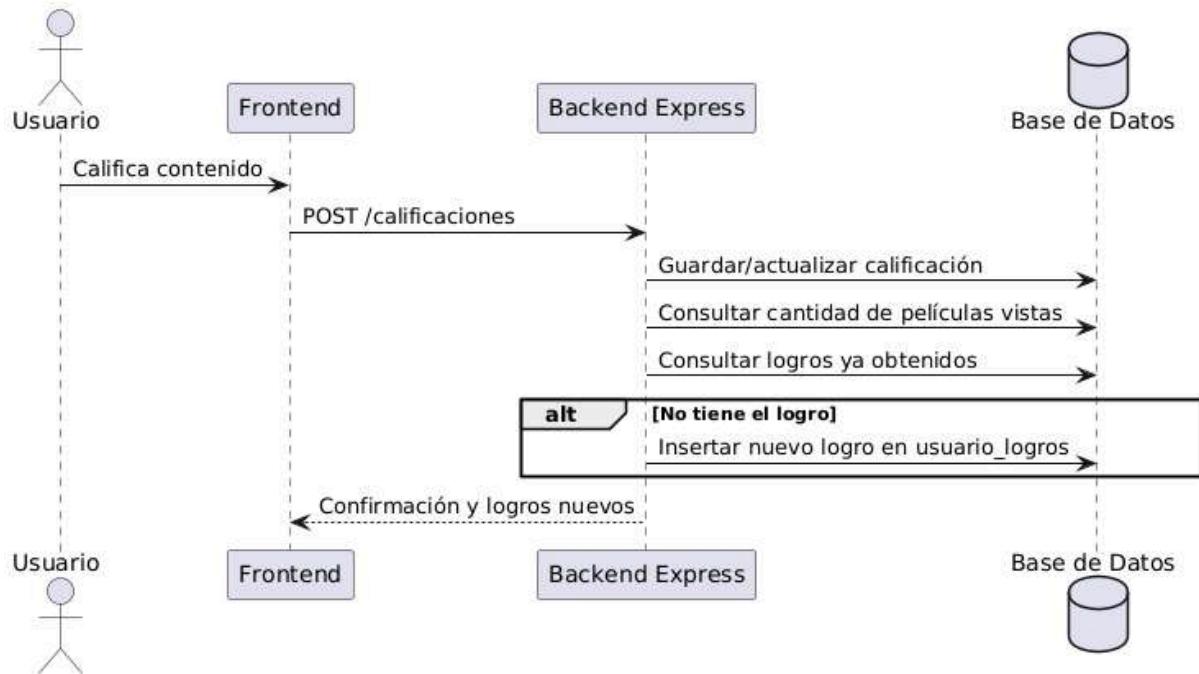
2.5.2.4 Gestión de logros

La gestión de logros es una funcionalidad que busca gamificar la experiencia del usuario en CineChamp. A medida que los usuarios realizan acciones relevantes dentro de la plataforma (como registrarse, calificar contenido o agregar favoritos), se les otorgan logros visuales que se muestran en su perfil público.

Descripción funcional:

- Los logros están predefinidos en una tabla específica del sistema.
- Cada logro contiene:
 - Código único
 - Título
 - Descripción
 - URL de la imagen asociada
- Se evalúan ciertas condiciones al realizar acciones para asignar logros:
 - Registro entre los primeros usuarios
 - Número de películas vistas
 - Agregar contenido a favoritos
- Los logros se guardan en una tabla intermedia `usuario_logros` que relaciona usuarios y logros desbloqueados.

Figura 2.16. Diagrama de secuencia Gestión de Logros



Fuente: **Elaboración propia**

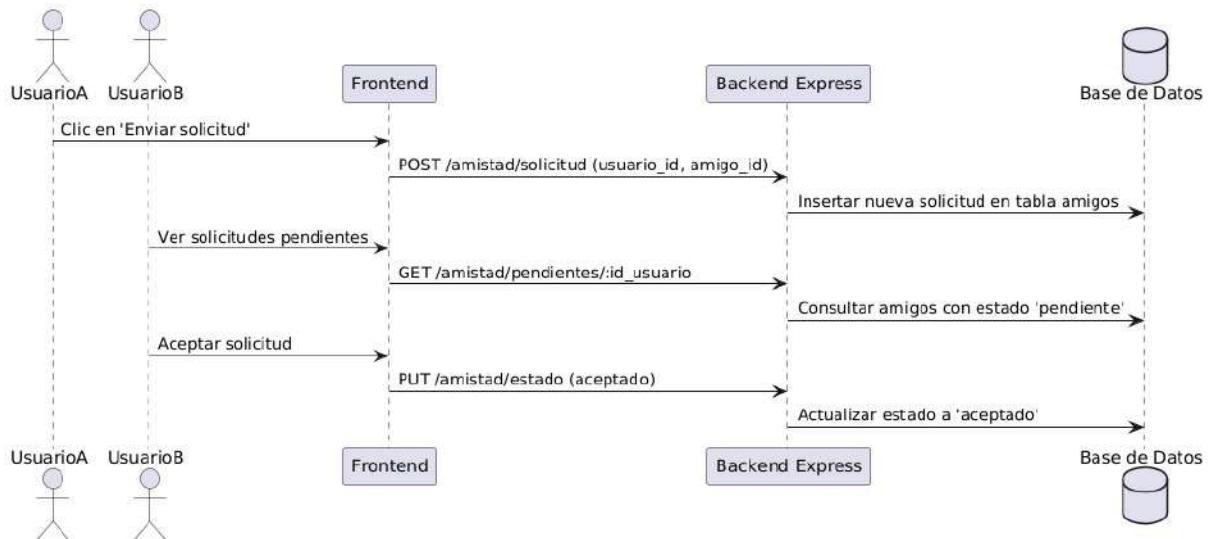
2.5.2.5. Gestión social

La funcionalidad social en **CineChamp** permite a los usuarios establecer relaciones de amistad con otros miembros de la comunidad. Esta característica mejora la interacción dentro de la plataforma, permitiendo ver el contenido favorito, calificaciones recientes y logros de los amigos.

Descripción funcional:

- Los usuarios pueden:
 - Enviar solicitudes de amistad.
 - Aceptar o rechazar solicitudes recibidas.
 - Ver una lista de sus amigos actuales.
 - Ver perfiles de amigos y sus actividades públicas.
- El estado de cada relación se gestiona mediante la tabla amigos.

Figura 2.17. Diagrama de secuencia Gestión Social



Fuente: **Elaboración propia**

2.5.3. Base de datos

La base de datos de **CineChamp** está diseñada siguiendo un modelo relacional con el sistema gestor MySQL. Se ha utilizado MySQL como sistema de gestión de bases de datos relacional por su rendimiento, soporte y compatibilidad con Node.js mediante el paquete mysql2.

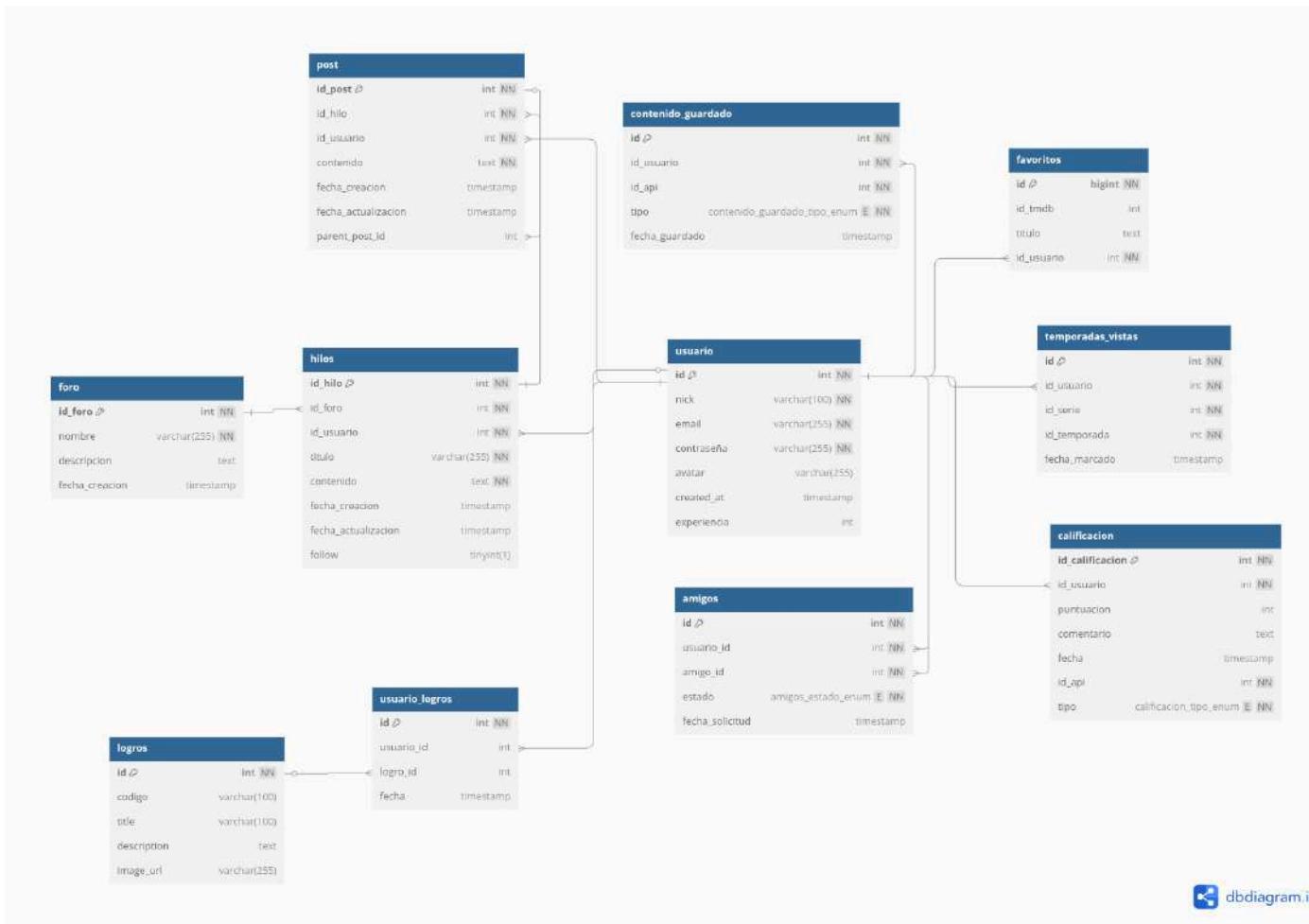
Tabla 2.2. Modelo Relacional

Tabla	Función principal
usuario	Almacena los datos personales de los usuarios registrados (nick, email, contraseña encriptada, avatar y experiencia).
amigos	Representa relaciones sociales entre usuarios. Permite solicitudes de amistad, aceptaciones, rechazos o bloqueos.
contenido_guardado	Permite que un usuario guarde contenido proveniente de la API externa (películas o series) para visualizarlo posteriormente.
favoritos	contiene contenido marcado como favorito por un usuario. Se

	garantiza la unicidad por usuario y contenido.
calificacion	forman el módulo del foro. Cada foro puede tener varios hilos, y cada hilo puede tener múltiples respuestas (post), incluso en forma de subrespuestas (parent_post_id).
foro, hilos, post	Define los logros disponibles en la plataforma.
logros	define los posibles logros que los usuarios pueden obtener (por número de películas vistas, primeras acciones, etc.).
usuario_logros	tabla intermedia que almacena qué logros ha alcanzado cada usuario.
temporadas_vistas	almacena qué temporadas de una serie han sido marcadas como vistas por un usuario, permitiendo llevar un seguimiento del progreso.

Fuente: **Elaboración propia**

Figura 2.18. Diagrama Entidad Relación



Fuente: Elaboración propia

2.5.4. API externa

El proyecto CineChamp se apoya en una fuente externa de datos audiovisuales mediante la integración de la API pública de The Movie Database (TMDb). Esta API es una de las más completas y utilizadas internacionalmente para acceder a información actualizada sobre películas, series, actores, sinopsis, imágenes promocionales, puntuaciones.

2.5.4.1. Propósito de la integración

El uso de esta API permite evitar el almacenamiento redundante de grandes volúmenes de contenido estático como títulos, pósteres, sinopsis y fechas de estreno. En su lugar, la aplicación guarda únicamente el ID de TMDb para cada película o serie, lo que permite:

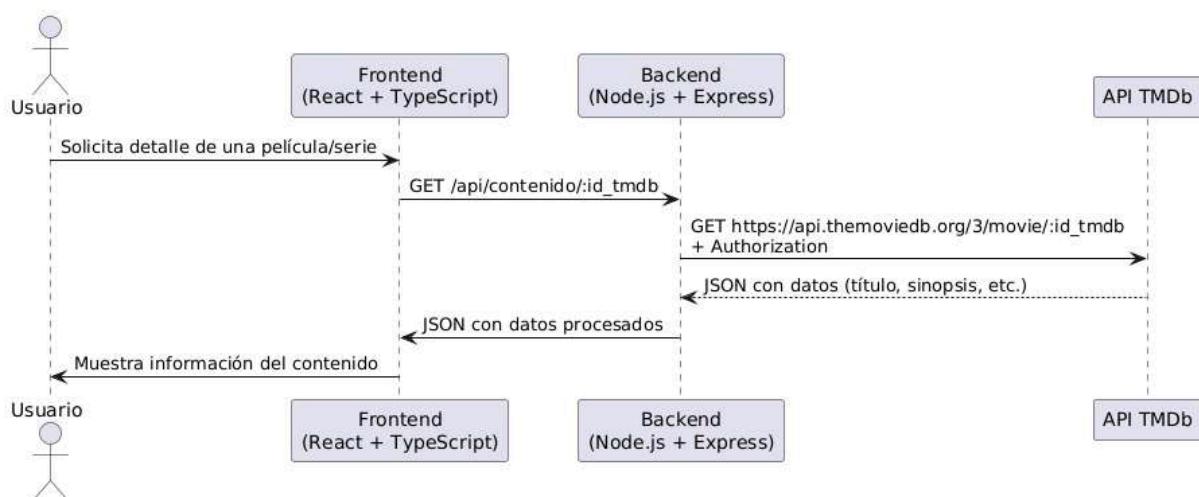
- Consultar información actualizada en tiempo real.
- Reducir la carga de datos en la base de datos local.
- Agilizar el desarrollo evitando la construcción de un sistema propio de catalogación de contenido.
- Facilitar búsquedas y detalles enriquecidos para los usuarios.

2.5.4.2. Interacción con TMDb

La comunicación entre CineChamp y la API de TMDb se realiza desde el backend, que actúa como intermediario entre el frontend y el servicio externo. Esta arquitectura permite ocultar claves API, aplicar lógica adicional si es necesario, y evitar llamadas directas desde el cliente.

A continuación, se muestra un diagrama de secuencia que representa esta interacción:

Figura 2.19. Diagrama de secuencia Comunicación con la API Externa



Fuente: **Elaboración propia**

2.5.4.3. Gestión segura y eficiente de la API

Para garantizar la seguridad y el correcto uso de la API de TMDb, se implementan mecanismos como:

- Almacenamiento seguro de la clave API en variables de entorno en el backend (.env), evitando así su exposición en el cliente.
- Uso de almacenamiento en caché o persistencia temporal cuando es necesario mostrar contenido frecuente, como en la página principal o detalles de películas populares, lo que reduce la latencia y mejora la experiencia de usuario.

2.5.4.4. Funcionalidades potenciadas por TMDb

Gracias a esta integración, **CineChamp** permite al usuario disfrutar de funcionalidades mejoradas como:

- Autocompletado en la búsqueda de contenido utilizando los endpoints /search/movie y /search/tv.
- Visualización del reparto de actores en una película o serie mediante /movie/{id}/credits o /tv/{id}/credits.
- Navegación por temporadas y episodios de una serie con el endpoint /tv/{id}/season/{season_number}.
- Personalización del historial y favoritos, ya que el sistema reconoce el contenido gracias al id_tmdb.

Estas funcionalidades añaden valor al usuario sin cargar la base de datos local con información redundante o voluminosa.

2.6. Arquitectura del sistema

El proyecto **CineChamp** se estructura siguiendo el patrón arquitectónico Modelo Vista Controlador (MVC), ampliamente utilizado en el desarrollo de aplicaciones web por su capacidad para separar responsabilidades, mejorar la mantenibilidad y facilitar la escalabilidad del sistema.

2.6.1. Implementación

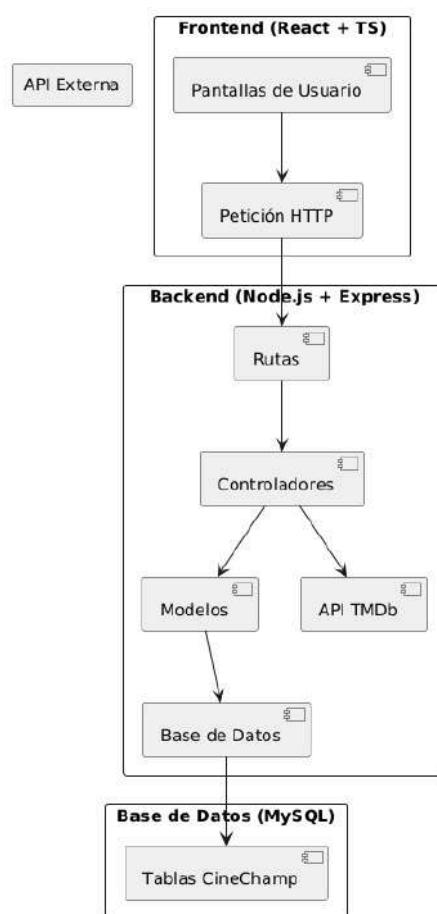
En **CineChamp**, el patrón MVC se aplica con claridad en la separación entre frontend, backend y la base de datos:

- **Modelo (Model):** Gestiona la lógica y el acceso a los datos. Archivos como usuarioModel.js, contenidoModel.js y logrosModel.js contiene la definición de las entidades y sus operaciones sobre la base de datos MySQL.

- Vista (View): Está compuesta por componentes React escritos en TypeScript. Representan la interfaz gráfica con la que interactúa el usuario, mostrando información y permitiendo la interacción con el sistema.
- Controlador (Controller): Los controladores Express como usuarioController.js, contenidoController.js, entre otros, contienen la lógica de negocio y orquestan la interacción entre rutas, modelos y respuestas al cliente.

Este patrón permite que cada parte del sistema evolucione de manera independiente, y asegura que los datos y la interfaz estén claramente desacoplados.

Figura 2.20. Diagrama de Arquitectura MVC



Fuente: **Elaboración propia**

Capítulo III

Funcionalidad y

pruebas del sistema

3. Funcionalidad y pruebas del sistema

3.1. Funcionalidad general de la plataforma

En esta sección se describen las principales funcionalidades del sistema **CineChamp** desde el punto de vista del usuario.

La interfaz se centra especialmente en el perfil del usuario, como eje central del sistema. Desde allí, se accede a funcionalidades como:

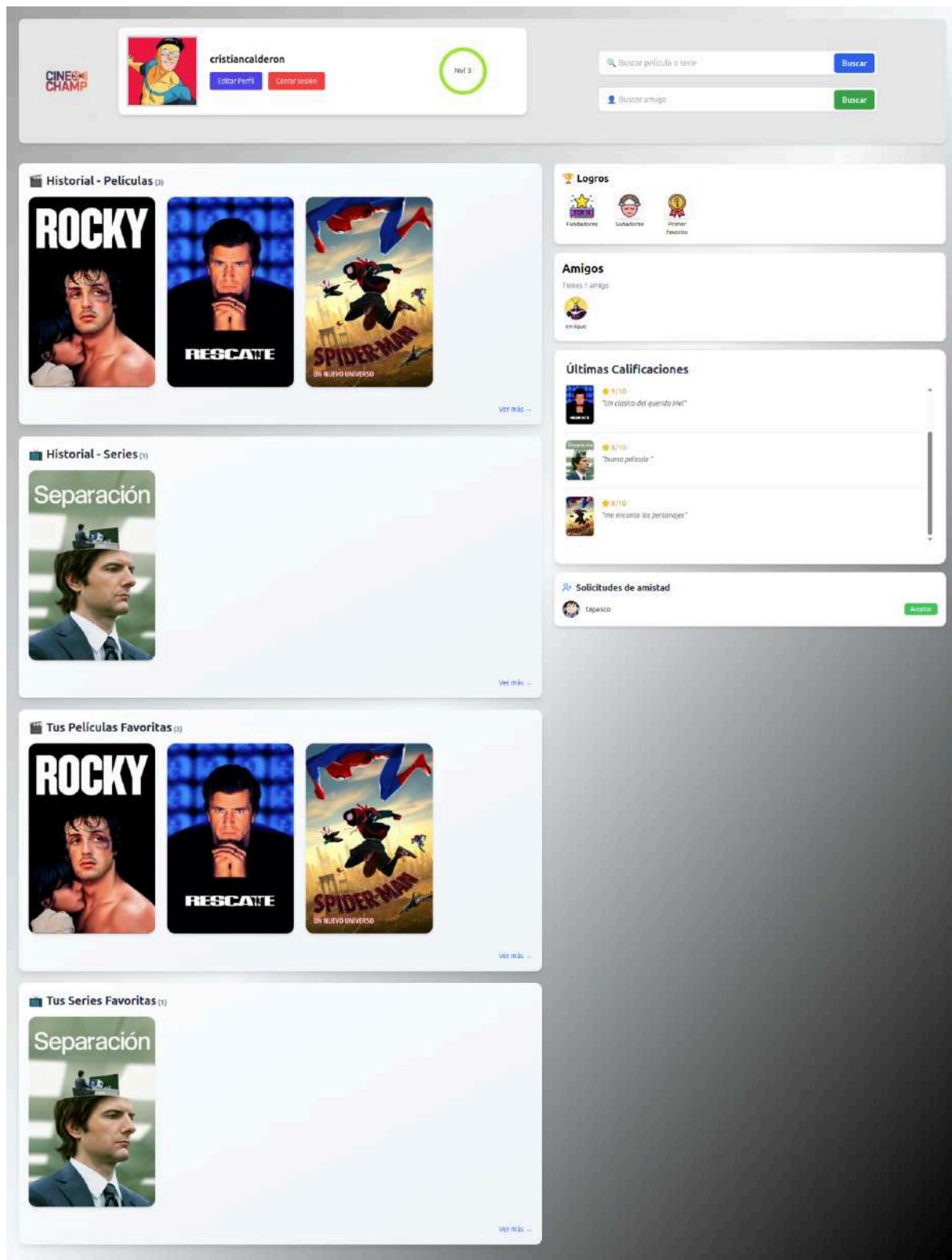
- Visualización del historial de películas y series.
- Gestión de contenido favorito.
- Acceso a logros desbloqueados.
- Interacción con otros usuarios mediante el sistema de amigos.
- Navegación por reseñas, calificaciones y estadísticas.

3.1.1. Diseño de interfaz abstracta

En esta fase se definieron las estructuras base de las interfaces gráficas de la aplicación, enfocándose en la organización lógica de los elementos y en la interacción del usuario con cada componente.

A continuación se presentan algunas de las interfaces clave diseñadas para CineChamp:

Figura 3.1. Pantalla Principal



Fuente: Elaboración propia

Pantalla principal (Perfil de usuario)

Contiene accesos directos a las funcionalidades más importantes:

- Editar perfil
- Cerrar sesión
- Nivel de usuario
- Buscador de películas o series
- Apartado para los logros obtenidos
- Historial de Películas
- Historial de Series
- Películas favoritas
- Series favoritas
- Amigos
- Últimas calificaciones
- Solicitudes pendientes de amistad

Botón “Editar perfil”

Permite al usuario modificar su nombre de usuario (nick) y su imagen de avatar. El formulario presenta una validación simple y una interfaz amigable para cargar una nueva imagen.

Figura 3.2. Editar Perfil

Editar Perfil

Nick
cristiancalderon

Avatar (URL)
/uploads/1748442940072-perfil invencible-rojo.jpeg

o Subir nueva imagen
Choose File No file chosen

Guardar cambios **Cancelar**

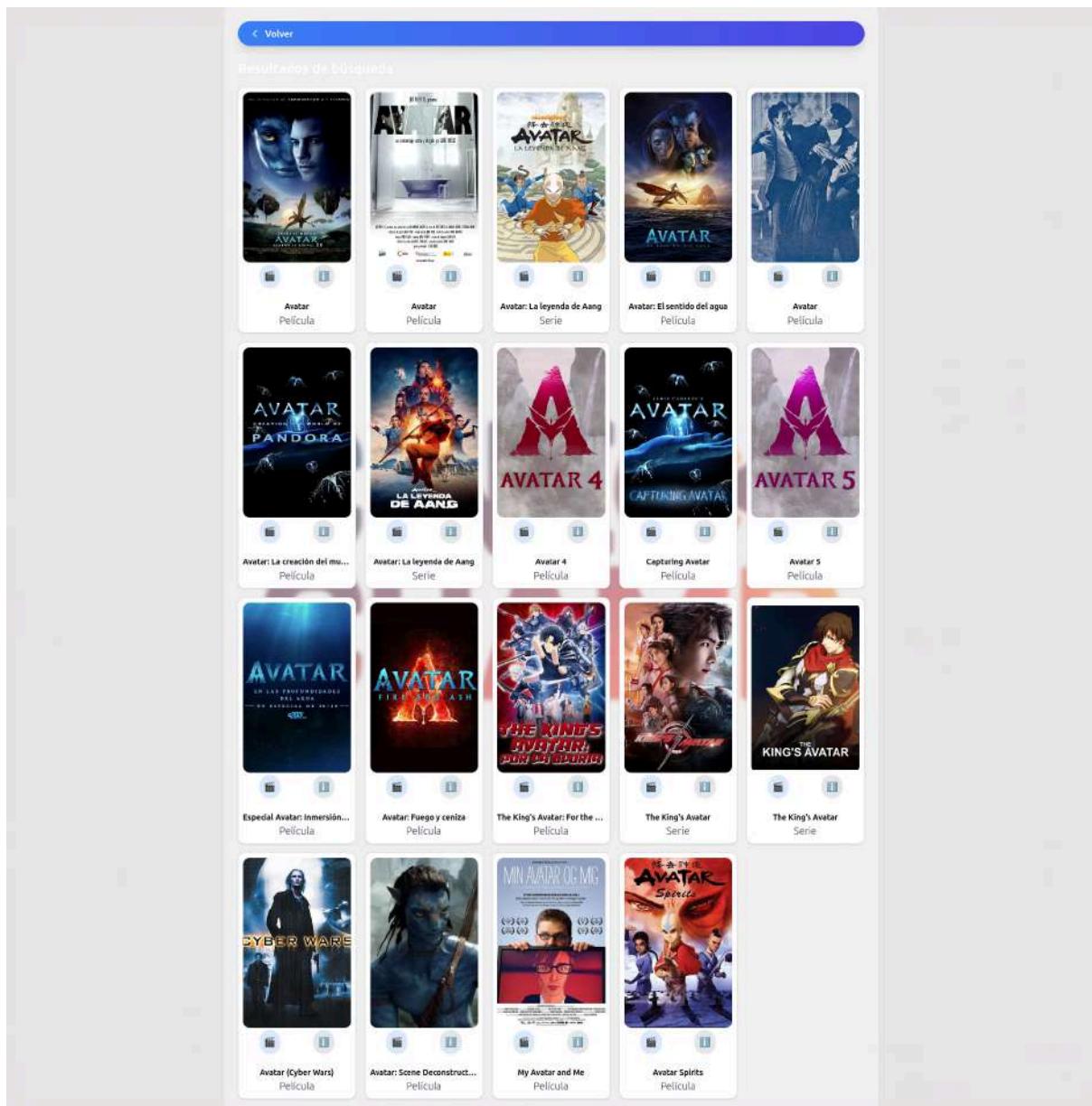
Fuente: **Elaboración propia**

Pantalla de buscador resultado película o serie

Muestra información obtenida desde la API externa TMDb (sinopsis, póster, puntuación, reparto) junto con botones para:

- Marcar como visto.
- Calificar.
- Acceder a la ficha de esta película.

Figura 3.3. Resultado de búsqueda películas y series

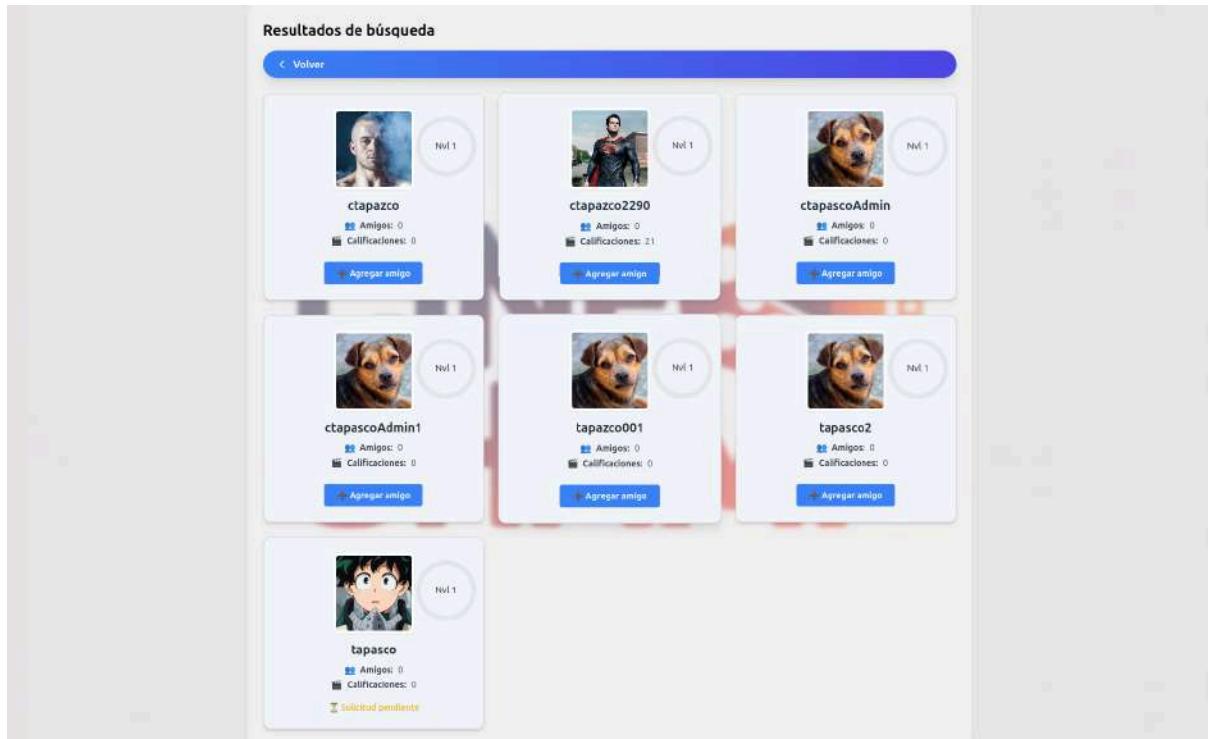


Fuente: Elaboración propia

Pantalla de amigos

Muestra la cantidad de resultados de usuarios con el mismo nombre de búsqueda junto con un botón para agregar amigo en el caso de que ya fuera agregado se mostrará, solicitud pendiente.

Figura 3.4. Pantalla de amigos



Fuente: Elaboración propia

Pantalla de información de película o serie

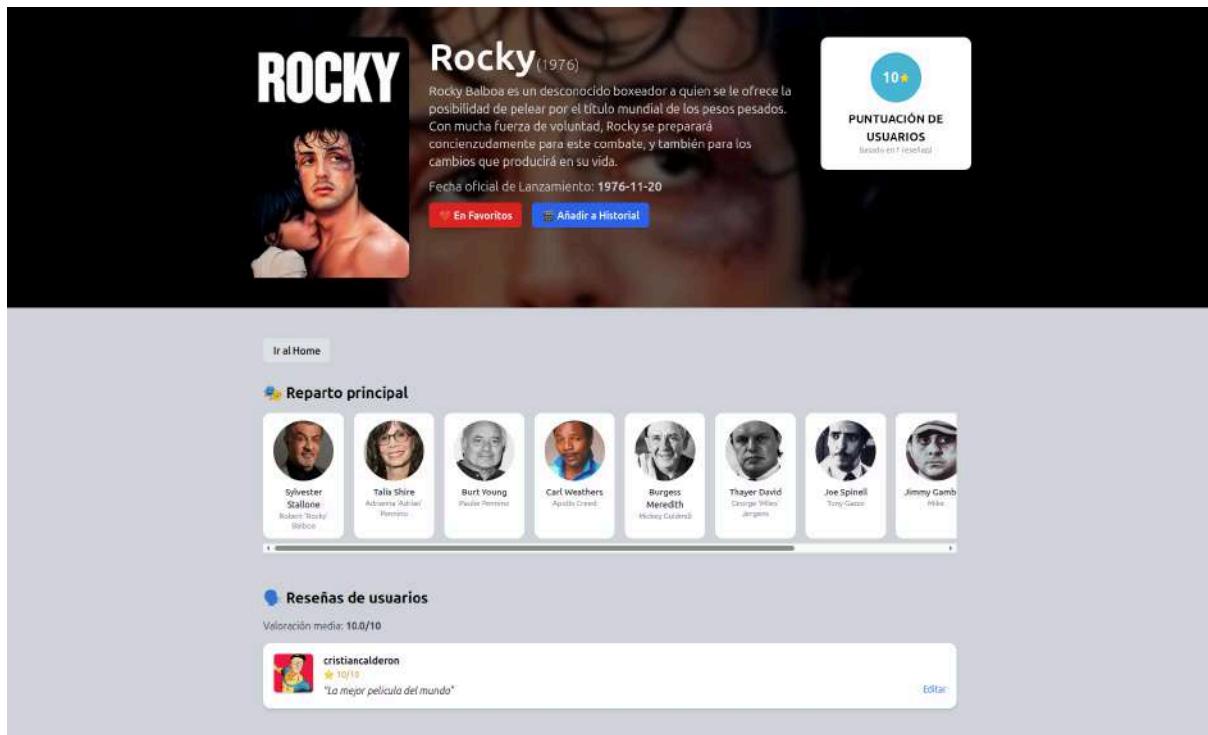
Esta interfaz muestra información detallada sobre un contenido específico (película o serie), permitiendo al usuario consultar y gestionar sus acciones relacionadas con ese contenido. Incluye los siguientes elementos:

- Título de la obra.
- Año de lanzamiento obtenido a partir de la fecha de estreno proporcionada por la API de TMDb.
- Sinopsis breve que resume el argumento.
- Puntuación media calculada con base en todas las calificaciones realizadas por los usuarios.
- Listado de actores principales que participaron en la obra, también proporcionado por la API externa.

Además, la interfaz proporciona las siguientes funcionalidades:

- Botón “En Favoritos”: Permite al usuario agregar el contenido a su lista de favoritos. Si ya está agregado, este botón puede mostrar una opción para eliminarlo.
- Botón “Añadir a Historial”: Guarda el contenido como visualizado por el usuario en su historial personal.
- Botón “Ir al Home”: Redirige al usuario de vuelta a la pantalla principal o perfil.
- Botón “Añadir reseña” / “Editar reseña”:
 - Si el usuario no ha realizado una reseña previa, se le permite crear una nueva puntuación y comentario.
 - Si ya existe una reseña del usuario, se le ofrece la opción de modificar el comentario o la calificación.

Figura 3.5. Pantalla información personal de película o serie



Fuente: Elaboración propia

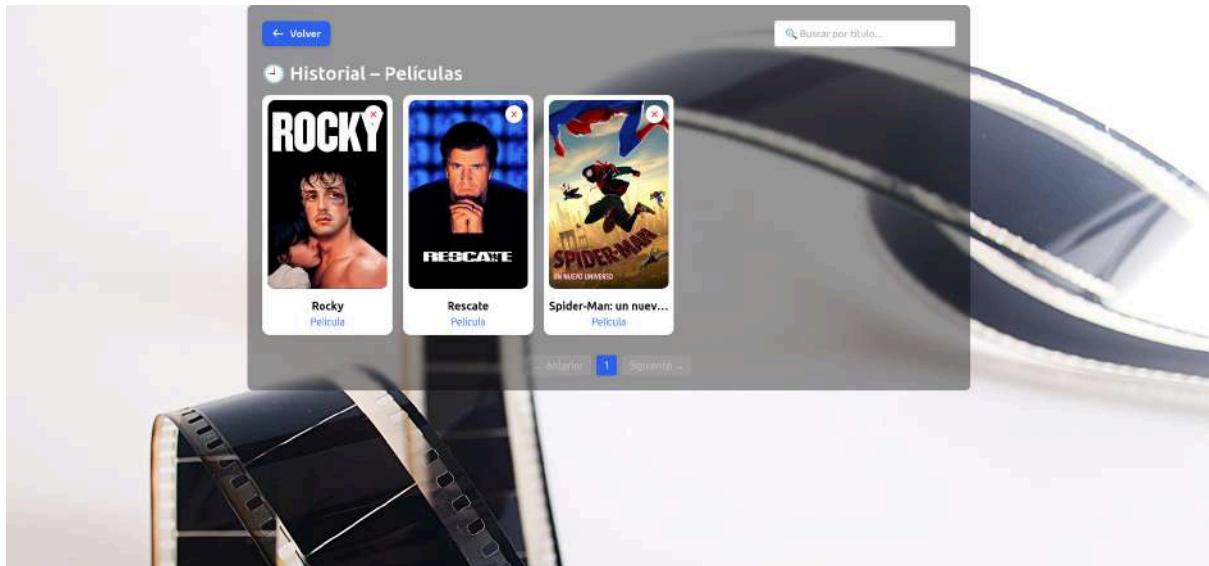
Pantalla de “Historial de Películas y Series” y “Favoritos”

Ambas pantallas comparten una estructura visual y funcional similar gracias a la reutilización de componentes en el desarrollo. Estas secciones están diseñadas para ofrecer una experiencia intuitiva al usuario, permitiéndole gestionar su contenido visualizado o guardado de forma eficiente.

Cada pantalla presenta los siguientes elementos:

- Botón de volver: Ubicado en la parte superior, permite regresar fácilmente a la pantalla anterior.
- Buscador interno: Facilita la localización de títulos específicos dentro del historial o favoritos, especialmente útil cuando el usuario tiene muchos elementos guardados.
- Sistema de paginación: Divide el contenido en páginas para mejorar la navegación y el rendimiento de carga.
- Botón de eliminación: Cada tarjeta de contenido cuenta con un botón que permite al usuario quitar esa película o serie del historial o de favoritos.

Figura 3.6. Pantalla Historial



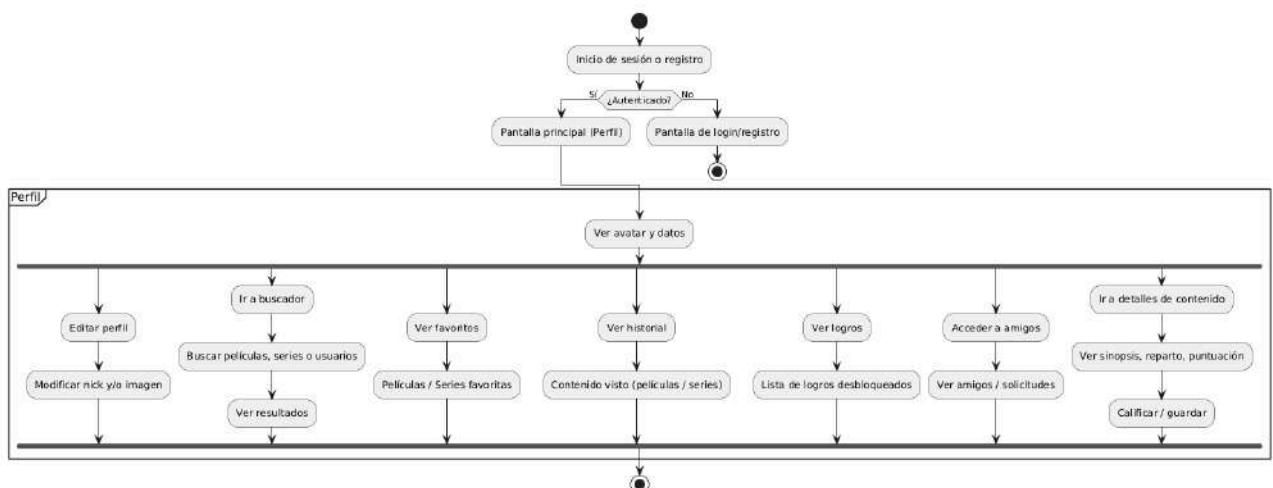
Fuente: **Elaboración propia**

3.1.2. Diseño navegacional

El diseño navegacional de **CineChamp** establece la forma en que los usuarios se desplazan entre las distintas pantallas y funcionalidades de la aplicación. Se ha optado por un flujo simple, utilizando menús y accesos visibles a las funcionalidades clave desde la pantalla principal.

El objetivo principal de este diseño es facilitar la experiencia de usuario y minimizar los pasos necesarios para realizar acciones frecuentes, como buscar contenido, consultar su perfil o interactuar con otros usuarios.

Figura 3.7. Diagrama de flujo navegacional



Fuente: **Elaboración propia**

3.2. Fase de pruebas

Esta fase permite verificar la funcionalidad de la aplicación web **CineChamp** concluida en su totalidad después de haber pasado por sus diferentes fases, obteniendo así diferentes resultados del funcionamiento, interactividad y consistencia que serán necesarios para el cumplimiento de la hipótesis planteada empleando las pruebas de caja blanca y las pruebas de caja negra.

- Pruebas de caja blanca, centradas en la lógica interna del código.
- Pruebas de caja negra, enfocadas en el comportamiento externo del sistema desde la perspectiva del usuario.

3.2.1. Etapa de Ejecución

Como se mencionó anteriormente se utilizará las pruebas de caja blanca que se centra en hacer test al ejecutar el software en base al código fuente, por el contrario, las pruebas de caja negra no se tiene acceso al código se orienta a ejecutar directamente la aplicación e interacción con sus interfaces.

3.2.2. Pruebas de caja blanca

A continuación se analizará las funcionalidades de la aplicación mediante test de código y casos de pruebas:

- Prueba de caja blanca 1 : Login y Registro

Tabla 3.1. Caso de prueba: Login y Registro

Caso de Prueba: 1	Ingreso a la aplicación			
Propósito	Verificar si se carga correctamente la pantalla principal			
Prerrequisitos	Usuario registrado y activo			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
	Comprobación de redirección tras login	Se redirige al perfil del usuario	Se redirige al perfil del usuario	Éxito
Pasos	Llenar el formulario de registro y presionar el botón Registrar			

Fuente: **Elaboración propia**

Tabla 3.2. Caso de prueba: Edición de perfil

Caso de Prueba: 2	Edición de perfil de usuario			
Propósito	Verificar si el sistema actualiza correctamente los datos del perfil (nick e imagen)			
Prerrequisitos	Usuario autenticado			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
Nuevo nick: "CineFan123", nueva imagen cargada	Actualización de datos del perfil	Se guardan los nuevos datos y se muestran en el perfil	Se guardan los nuevos datos y se muestran en el perfil	Éxito
Pasos	Acceder a la pantalla de perfil, presionar “Editar perfil”, introducir nuevo nick y/o imagen, presionar “Guardar cambios”			

Fuente: **Elaboración propia**

Tabla 3.3. Caso de prueba: Agregar contenido a favoritos

Caso de Prueba: 3	Agregar contenido a favoritos			
Propósito	Verificar que el contenido seleccionado se agregue correctamente a la lista de favoritos del usuario			
Prerrequisitos	Usuario autenticado, contenido disponible con ID válido (ej. id_tmdb = 603)			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
ID de contenido: 603 (ej. Matrix)	Funcionalidad de marcación como favorito	El contenido aparece en la sección “Tus películas/series favoritas” del usuario	El contenido aparece en la sección “Tus películas/series favoritas” del usuario	Éxito
Pasos	Ingresar a la ficha del contenido, presionar el botón “Agregar a favoritos”, ir al perfil y verificar si aparece en favoritos			

Fuente: **Elaboración propia**

Tabla 3.4. Caso de prueba: Eliminar contenido del historial

Caso de Prueba: 4	Eliminar contenido del historial			
Propósito	Verificar si el sistema elimina correctamente un elemento del historial del usuario			
Prerrequisitos	Usuario autenticado y con contenido previamente guardado como visto			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
ID de contenido: 672 (ej. Tenet)	Gestión del historial	El contenido desaparece del historial del usuario	El contenido desaparece del historial del usuario	Éxito
Pasos	Ir al perfil, localizar el contenido, presionar el botón “Eliminar del historial”, confirmar acción			

Fuente: **Elaboración propia**

Tabla 3.5. Caso de prueba: Calificar contenido

Caso de Prueba: 5	Calificar contenido			
Propósito	Verificar si el sistema elimina correctamente un elemento del historial del usuario			
Prerrequisitos	Usuario autenticado			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
ID TMDb: 155, Puntuación: 9, Comentario: “Excelente narrativa”	Gestión de calificaciones	La calificación y el comentario se guardan y se muestran en el perfil del usuario	La calificación y el comentario se muestran correctamente	Éxito
Pasos	Acceder al contenido, introducir la puntuación y comentario, presionar “Enviar”, verificar en perfil			

Fuente: **Elaboración propia**

Tabla 3.6. Caso de prueba: Enviar solicitud de amistad

Caso de Prueba: 6	Agregar amigo			
Propósito	Verificar si se registra correctamente una solicitud de amistad entre dos usuarios			
Prerrequisitos	Usuario autenticado, segundo usuario existente en la base de datos			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
ID del segundo usuario: 21	Gestión de relaciones sociales	La solicitud se registra con estado “pendiente”	Se registra solicitud con estado “pendiente” en la tabla amigos	Éxito
Pasos	Buscar usuario por nombre, presionar “Agregar amigo”, verificar en base de datos o interfaz de solicitudes			

Fuente: **Elaboración propia**

Tabla 3.7. Caso de prueba: Desbloqueo de logro por visualizaciones

Caso de Prueba: 7	Desbloqueo automático de logro al ver contenido			
Propósito	Verificar si se desbloquea un logro al cumplir el criterio de películas vistas			
Prerrequisitos	Usuario autenticado con al menos 9 películas vistas previamente			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
Visualización de 1 nueva película (ej. ID TMDb: 2049)	Lógica de logros automática	Se otorga el logro “Nuevo Freak” al alcanzar 10 películas vistas	Logro desbloqueado y visible en el perfil del usuario	Éxito
Pasos	Ver película, esperar verificación automática del sistema, ir a perfil/logros y comprobar desbloqueo			

Fuente: **Elaboración propia**

Tabla 3.8. Caso de prueba: Caso de prueba: Buscar contenido audiovisual

Caso de Prueba: 8	Buscar películas, series o usuarios			
Propósito	Verificar que el buscador unificado devuelve resultados correctos según el término introducido			
Prerrequisitos	Usuario autenticado			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
Término de búsqueda: "Matrix"	Función de búsqueda general	Se muestran resultados relacionados con películas, series y usuarios con el término "Matrix"	Resultados mostrados correctamente en pantalla	Éxito
Pasos	Ir al buscador, escribir "Matrix", presionar buscar, verificar los resultados en pantalla			

Fuente: **Elaboración propia**

Tabla 3.9. Caso de prueba: Cierre de sesión

Caso de Prueba: 9	Cerrar sesión			
Propósito	Verificar si el sistema cierra correctamente la sesión del usuario y lo redirige al login			
Prerrequisitos	Usuario autenticado			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
Acción del usuario: clic en "Cerrar sesión"	Manejo de autenticación	El usuario es redirigido a la pantalla de login y se borra el token local	Se redirige al login y se borra el token del almacenamiento	Éxito
Pasos	Estando logueado, hacer clic en "Cerrar sesión" en el header, confirmar que vuelve al login			

Fuente: **Elaboración propia**

Tabla 3.10. Caso de prueba: Ver el detalle de una película o serie desde su ID

Caso de Prueba: 10	Visualización de detalles de contenido por ID			
Propósito	Verificar que al acceder a un contenido específico mediante su ID, se muestran correctamente sus datos			
Prerrequisitos	Usuario autenticado, contenido con ID TMDb válido			
Datos de entrada	Atributo del sistema	Resultado esperado	Resultado Obtenido	Resultado
ID TMDb: 603 (The Matrix)	Consulta a la API externa TMDb + renderización en frontend	Se muestran los datos del contenido: sinopsis, imagen, reparto, botones de acción	Todos los datos son cargados correctamente en la vista	Éxito
Pasos	Desde búsqueda o lista de contenido, hacer clic en Matrix, verificar que se renderiza la vista con todos los datos relevantes extraídos de la API			

Fuente: **Elaboración propia**

3.2.2. Pruebas de caja negra

A continuación se analizará desde el punto de vista funcional sin tomar en cuenta la estructura interna de la aplicación web es decir el código fuente, se evaluará los aspectos externos del sistema como interfaces, integridad o rendimiento del sistema empleando un determinado conjunto de datos de entrada y de esta manera observar las salidas que se producen para lo cual a continuación se presenta las siguientes pantallas de prueba.

Tabla 3.11. Caso de prueba: Login

	CineChamp Inicia sesión en tu cuenta Correo electrónico <input type="text" value="ejemplo@correo.com"/> <input type="button" value="Olvidaste tu contraseña?"/> Contraseña <input type="password"/> Tu contraseña <input type="button" value="Ingresar"/> <small>(No tienes cuenta? Regístrate.)</small>
Caso de Prueba	Login de usuario
Propósito	Verificar que un usuario puede acceder correctamente al sistema introduciendo credenciales válidas
Entradas	Nick: usuario123 Contraseña: 123456
Resultado Esperado	Redirección al perfil del usuario, con todos sus datos cargados
Resultado Obtenido	El sistema redirige correctamente y muestra la información del usuario
Resultado	Éxito
Observaciones	El sistema debe mostrar mensaje de error si las credenciales no son válidas

Fuente: **Elaboración propia**

Tabla 3.12. Caso de prueba: Registro de usuario

Caso de Prueba	Registro de nuevo usuario
Propósito	Verificar que un nuevo usuario puede registrarse correctamente en la plataforma
Entradas	Nick: nuevoUsuario Email: nuevo@cinechamp.com Contraseña: clave1234 Imagen: opcional
Resultado Esperado	El sistema crea el usuario y redirige al login o al perfil
Resultado Obtenido	El usuario es creado correctamente y se redirige
Resultado	Éxito
Observaciones	Si el email o nick ya existe, debe mostrar un mensaje de error claro

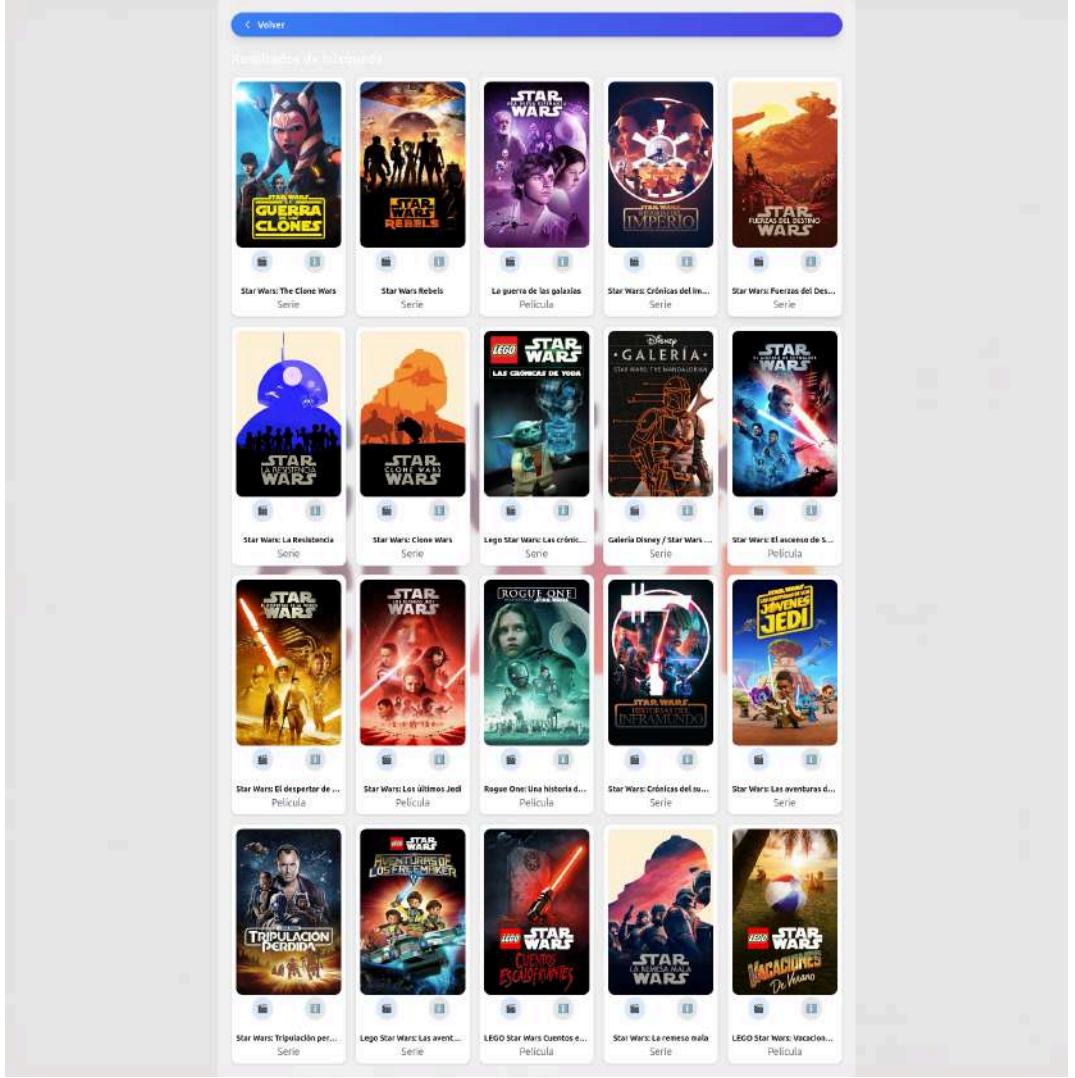
Fuente: **Elaboración propia**

Tabla 3.13. Caso de prueba: Editar perfil

Caso de Prueba	Modificación de datos de perfil
Propósito	Comprobar que el usuario puede editar su nick o imagen de perfil
Entradas	Nuevo Nick: usuarioEditado Imagen: nueva-foto.jpg
Resultado Esperado	Actualización de los datos del perfil en la interfaz
Resultado Obtenido	Los datos se actualizan y se reflejan en la vista
Resultado	Éxito
Observaciones	El nuevo nick debe ser único, en caso contrario mostrar advertencia

Fuente: **Elaboración propia**

Tabla 3.14. Caso de prueba: Búsqueda de contenido



The screenshot shows a search results page for "Star Wars". The results are displayed in a grid format with 5 columns and 6 rows. Each item includes a thumbnail image, the title, and a subtitle indicating if it's a movie or a series.

Caso de Prueba	Buscar película o serie
Propósito	Validar que el sistema muestra resultados relevantes al buscar contenido
Entradas	Término de búsqueda: Star Wars
Resultado Esperado	Se muestran resultados relacionados con la búsqueda desde la API externa
Resultado Obtenido	Los resultados coinciden correctamente
Resultado	Éxito
Observaciones	Se debe mostrar mensaje si no hay resultados encontrados

Fuente: Elaboración propia

Tabla 3.15. Caso de prueba: Visualizar perfil público de otro usuario

Caso de Prueba	Visualizar perfil público de otro usuario
Propósito	Comprobar que un usuario puede ver correctamente el perfil público de otro usuario
Entradas	Click en avatar de otro usuario desde lista de amigos o resultados de búsqueda
Resultado Esperado	Carga del perfil público con avatar, logros, favoritos, historial.
Resultado Obtenido	Resultado esperado
Resultado	Éxito

Fuente: Elaboración propia

Tabla 3.16. Caso de prueba: Enviar solicitud de amistad

Caso de Prueba	Enviar solicitud de amistad
Propósito	Validar que un usuario pueda enviar solicitud de amistad a otro usuario
Entradas	Click en botón 'Aregar amigo' en perfil de otro usuario
Resultado Esperado	Mensaje de confirmación y actualización del estado a 'Solicitud pendiente'
Resultado Obtenido	Resultado esperado
Resultado	Éxito

Fuente: **Elaboración propia**

Tabla 3.17. Caso de prueba: Ver el detalle de una película desde su ID

The screenshot displays the detailed view of a movie from a database. At the top, there's a large thumbnail image of the movie poster for "Spider-Man: un nuevo universo" (2018). Below the poster, the movie title and year are prominently displayed. A detailed synopsis follows, describing the plot involving Miles Morales becoming the new Spider-Man. To the right of the synopsis is a box showing a user rating of 9.0 based on 2 reviews. Below the synopsis, the release date is listed as 2018-12-06, along with two buttons: "En Favoritos" and "Añadir a Historial". Underneath the main title, there's a link to "Ir al Home". The "Reparto principal" section shows the main cast members with their names and character roles: Shameik Moore (Miles Morales), Jake Johnson (Peter B. Parker), Hailee Steinfeld (Gwen Stacy/wives), Mahershala Ali (Uncle Aaron (voice)), Brian Tyree Henry (Henry), Lily Tomlin (Aunt May (voice)), Luna Lauren Velez (Rio (voice)), and Zoë Kravitz (Macy Jane (voice)). Below the cast, there's a section for "Reseñas de usuarios" (User reviews) with a average rating of 9.0/10. Two reviews are shown: one from "ejemplo78" giving it 10/10 and another from "cristiancalderon" giving it 9/10.

Caso de Prueba	Ver el detalle de una película desde su ID
Propósito	Comprobar que se puede acceder a la vista de detalle desde una película específica
Entradas	Click en título o imagen desde lista de películas
Resultado Esperado	Vista detallada con sinopsis, reparto.
Resultado Obtenido	Resultado esperado
Resultado	Éxito

Fuente: Elaboración propia

Tabla 3.18. Caso de prueba: Calificar contenido (película o serie)

Caso de Prueba	Calificar contenido (película o serie)
Propósito	Verificar que un usuario pueda puntuar y comentar una película o serie
Entradas	Click en botón de calificación en la vista de detalle
Resultado Esperado	Puntuación guardada y visible en su historial y en promedio global
Resultado Obtenido	Resultado esperado
Resultado	Éxito

Fuente: **Elaboración propia**

Tabla 3.19. Caso de prueba: Agregar contenido a favoritos

Caso de Prueba	Agregar contenido a favoritos
Propósito	Evaluando que se puede añadir correctamente un contenido como favorito
Entradas	Click en ícono de favorito (estrella) desde la vista de detalle
Resultado Esperado	Cambio de ícono, confirmación visual y persistencia de favoritos
Resultado Obtenido	Resultado esperado
Resultado	Éxito

Fuente: Elaboración propia

Capítulo IV

Conclusiones

4. Conclusiones

4.1. Conclusiones generales del proyecto

Durante la realización del proyecto se han aprendido muchos aspectos relacionados con el desarrollo de una aplicación web, como el diseño de interfaces, la estructura del código, la lógica de programación y la organización del contenido. Aunque al inicio se planteó ciertas funcionalidades, no se era plenamente consciente de todo lo que implicaba desarrollarlas. Aun así, el resultado nos deja un buen sabor y se han conseguido llegar a los objetivos esenciales de nuestra aplicación.

A pesar de que algunas ideas iniciales se quedaron fuera por falta de organización o conocimientos al principio, el proyecto ha resultado ser una aplicación sólida, funcional y decentemente estructurada. Al trabajar en una aplicación como esta por primera vez, nos ha servido como una experiencia real de desarrollo profesional.

4.2. Consecuencias de los objetivos

Muchos de los objetivos propuestos se han conseguido: el sistema básico de gestión de contenido funciona correctamente, la navegación es clara y se ha desarrollado una API RESTful para estructurar las operaciones con los datos.

No obstante, algunas funcionalidades más avanzadas como la creación de foros y temas, el sistema de sugerencias, una cartelera dinámica o un sistema de logros personalizado no se llegaron a implementar por falta de tiempo o experiencia técnica.

Esto nos ha hecho ver lo importante que es tener una planificación realista desde el principio, saber priorizar funcionalidades clave y empezar a investigar y organizar el proyecto con mayor antelación.

4.3. Valoración de la metodología y planificación

El trabajo en equipo ha sido una parte muy positiva del proyecto. Cada tarea fue discutida previamente, se acordó quién la realizaría y cómo se repartiría la carga de trabajo. A medida que aparecían problemas o bloqueos, se tomaban decisiones de forma conjunta y flexible. Si alguien se atascaba o una funcionalidad resultaba más compleja de lo esperado, se reorganizaba el esfuerzo para avanzar en conjunto.

Aunque no se siguió una metodología formal como Scrum, sí se llevó a cabo un seguimiento constante, con revisiones diarias del estado del trabajo y ajustes continuos. Esta forma de organización ha demostrado ser eficaz para mantener el ritmo y alcanzar los objetivos establecidos.

4.4. Visión de futuro

En retrospectiva, se ha aprendido que algunas elecciones tecnológicas, como el tipo de almacenamiento o la forma de estructurar los datos, podrían haber sido diferentes. Elegir otras herramientas, aunque hubiera supuesto aprenderlas desde cero, podría haber evitado algunas de las limitaciones que se encontraron más adelante.

En el futuro, se podrían añadir funcionalidades como autenticación de usuarios, eliminar la dependencia de una API externa, implementar completamente el sistema de foros con temas, así como un sistema de compra de entradas o redirección a plataformas donde visualizar las series o películas registradas.

El proyecto cuenta con una base sólida, por lo que continuar su desarrollo sería una excelente oportunidad para seguir aprendiendo y llevar CineChamp a un siguiente nivel.

Capítulo V

Glosario

5. Glosario

API (Application Programming Interface): Conjunto de funciones y procedimientos que permiten la comunicación entre diferentes sistemas de software. En este proyecto se utiliza la API de TMDb para obtener datos actualizados sobre películas y series.

Backend: Parte del desarrollo web que se encarga de la lógica de negocio, procesamiento de datos y conexión con la base de datos. No es visible para el usuario final.

Base de datos: Sistema de almacenamiento estructurado de datos. CineChamp utiliza una base de datos relacional MySQL para guardar información de usuarios, amigos, calificaciones, logros, etc.

Caja blanca (White-box testing): Tipo de prueba de software que evalúa la lógica interna del código, considerando su estructura y flujo.

Caja negra (Black-box testing): Tipo de prueba que se enfoca en la funcionalidad del sistema sin tener en cuenta la lógica interna del código.

Cinefilia: Afición o amor por el cine. CineChamp está dirigido a personas con este perfil.

CSS (Cascading Style Sheets): Lenguaje utilizado para definir el estilo visual de las páginas web, incluyendo colores, fuentes y distribución.

Frontend: Parte del desarrollo web encargada de la interfaz de usuario y la experiencia visual e interactiva. En CineChamp se ha desarrollado con React y TypeScript.

Git: Sistema de control de versiones distribuido que permite llevar un seguimiento del historial de cambios realizados en el código fuente del proyecto.

GitHub: Plataforma en línea que aloja repositorios Git, facilitando la colaboración y publicación del código del proyecto.

HTTP (HyperText Transfer Protocol): Protocolo de comunicación utilizado para transferir información en la web entre cliente y servidor.

Kanban: Metodología ágil que permite gestionar proyectos mediante un tablero visual donde se organizan tareas en columnas como "Por hacer", "En progreso" y "Hecho".

MVC (Modelo-Vista-Controlador): Patrón de arquitectura de software que separa la lógica de negocio (Modelo), la interfaz de usuario (Vista) y el control del flujo de la aplicación (Controlador).

MySQL: Sistema de gestión de bases de datos relacional (RDBMS) ampliamente utilizado para aplicaciones web.

Node.js: Entorno de ejecución para JavaScript del lado del servidor, utilizado en el backend de CineChamp.

React: Biblioteca de JavaScript desarrollada por Meta para construir interfaces de usuario reutilizables, basada en componentes.

TMDb (The Movie Database): Base de datos en línea que proporciona información sobre películas, series y celebridades del mundo audiovisual. Su API es utilizada por CineChamp para obtener contenido dinámico.

TypeScript: Lenguaje de programación basado en JavaScript que incorpora tipado estático, utilizado en el desarrollo del frontend para mejorar la robustez del código.

Capítulo VI

Referencias

Bibliográficas

6. Referencias bibliográficas

Metodología Kanban

<https://asana.com/es/resources/what-is-kanban>

<https://kissflow.com/project/agile/kanban-methodology/#:~:text=Kanban%20is%20a%20pull%20system&text=Kanban%20is%20about%20work%20states,moves%20from%20left%20to%20right.>

Figma

Herramienta colaborativa de diseño de interfaces y prototipado visual. Se utilizó para diseñar la interfaz de usuario de la aplicación.

<https://www.figma.com>

Trello

Aplicación web de gestión de proyectos basada en tableros Kanban. Fue utilizada para la organización y seguimiento de tareas durante el desarrollo.

<https://trello.com>

MySQL

Sistema de gestión de bases de datos relacional utilizado para almacenar y gestionar la información de los usuarios, contenidos y relaciones en CineChamp.

<https://www.mysql.com>

React

Librería de JavaScript para construir interfaces de usuario. Fue empleada en el desarrollo del frontend de CineChamp.

<https://react.dev>

TMDb (The Movie Database)

Plataforma que ofrece una API pública con información de películas, series y actores. Fue integrada en CineChamp para acceder a contenido actualizado.

<https://www.themoviedb.org>

Express

Framework minimalista para Node.js que facilita la creación de servidores y rutas. Se utilizó para el desarrollo del backend.

<https://expressjs.com>

Node.js

Entorno de ejecución para JavaScript en el servidor. Fue la tecnología principal para implementar la lógica del backend.

<https://nodejs.org>

GitHub

Plataforma de alojamiento de código fuente y control de versiones basada en Git. Se utilizó para almacenar, versionar y colaborar en el desarrollo del proyecto.

<https://github.com>

Capítulo VII

Anexos

7. Anexos

Estimación de costes de desarrollo y mantenimiento de la plataforma CineChamp:

<https://franciscomorales.es/blog/ver/-cuanto-cuesta-el-desarrollo-de-una-pagina-web-en-esp-ana-/4#:~:text=Dominio%20y%20Hosting>

<https://www.ovhcloud.com/es-es/vps/os/vps-linux/#:~:text=Desde%20,mes>

<https://www.hostinger.com/es/tutoriales/precio-pagina-web#:~:text=Alojamiento%20web%20%20a%2080,30%20%E2%82%AC%2Fhora%E2%80%9330%20%E2%82%AC%2Fhora%20o%20incluido>

<https://www.hostinger.com/es/tutoriales/precio-pagina-web#:~:text=Alojamiento%20web%20%20a%2080,30%20%E2%82%AC%2Fhora%E2%80%9330%20%E2%82%AC%2Fhora%20o%20incluido>

<https://www.cloudflare.com/es-es/plans/#:~:text=Seguridad%2C%20rendimiento%20y%20fiabilidad%20en,Facturaci%C3%B3n%20anual%20o>

Trello:

<https://trello.com/b/c2trhtyU/cinechamp>

Figma:

https://www.figma.com/design/OhEgeZd6A8NHMEM3AoDAEj/CineChamp?node_id=0-1&t=dy00iWddjAMUV7kw-1

Plan de Prevención de riesgos laborales:

https://docs.google.com/document/d/16rb09eEI573K3G0lrt_EmrkjAf6OZX71u-i1HWj2-Xc/edit?tab=t.0