

BrisaacFlix

Gestor de Rankings de Películas y Series

**Brian Fair Suarez Porras
Jhon Isaac Medina Mendoza**

GRUPO #1

Juan Carlos Mariño Morantes

**CAMPUSLANDS
SALON Artemis-U2
RUTA Node.js
FLORIDABLANCA
2025
BrisaacFlix**

1. SITUACIÓN PROBLEMA

El consumo de contenidos geek (películas, series, anime) en la comunidad hispanohablante ha crecido de forma sostenida. Existen múltiples plataformas de streaming y redes sociales, pero la información útil para decidir qué ver (calidad de las reseñas, relevancia del ranking, señales sociales y curaduría) está fragmentada, es inconsistente y suele estar mezclada con ruido (spam, bots, sesgos de calificación, títulos duplicados o desactualizados).

Problema Actual:

Descubrimiento pobre: cuesta encontrar recomendaciones confiables y actualizadas por categoría geek (Anime, Ciencia Ficción, Superhéroes, Fantasía, etc.).

- **Ratings poco representativos:** muchos portales usan promedios simples que no ponderan la calidad de la reseña, el apoyo social (likes/dislikes) ni la frescura temporal.
- **Moderación limitada:** falta un flujo de aprobación para nuevas fichas y controles contra títulos repetidos o información incompleta.
- **Experiencias separadas:** el frontend y el backend de recursos educativos a menudo no simulan un entorno productivo real (autenticación robusta, límites de tasa, transacciones de BD, documentación de API y métricas de calidad).

Oportunidad

Construir **BrisaacFlix**, una aplicación **full-stack** con **Node.js + Express** (backend) y **React** (frontend), que proporcione:

- **Ranking ponderado** que combine calificación numérica (1–10), apoyo social (likes/dislikes) y factor tiempo (premiando reseñas recientes sin sesgar injustamente clásicos).
- **Moderación:** flujo de aprobación de títulos y gestión de categorías solo para administradores.
- **Autenticación segura** (JWT con passport-jwt, bcrypt) y limitación de tasa (express-rate-limit).
- **Validaciones robustas** (express-validator) y transacciones reales en MongoDB para mantener consistencia.
- **API versionada y documentada con Swagger**, consumida por un frontend independiente.

Referentes (benchmark)

- **IMDb**: base masiva y metadatos; enfoque generalista y promedios simples. Falta foco en categorías geek y señal social explícita sobre calidad de reseñas.
- **Rotten Tomatoes**: separación crítica/audiencia; menos granularidad en subgéneros geek y comunidad de reseñas propia limitada.
- **MyAnimeList**: fuerte en anime/manga; no cubre integralmente cine/series occidentales ni pondera señales sociales cruzadas.
- **Letterboxd**: componente social y listas; menos estructura en **roles/flujo de aprobación** y ranking ponderado por señales sociales.

2. Levantamiento de Requerimientos

2.1 Requerimientos Funcionales BackEnd

Los **requerimientos funcionales** definen las características que la aplicación debe cumplir para resolver la situación problema planteada.

Gestión de Usuarios

- **RF-01**: El sistema debe permitir el registro de usuarios con datos básicos (nombre, correo, contraseña).
- **RF-02**: El sistema debe permitir el inicio de sesión de usuarios registrados mediante **JWT**.
- **RF-03**: El sistema debe diferenciar entre roles: **usuario** y **administrador**.
- **RF-04**: Los administradores deben poder aprobar nuevas películas/series y gestionar categorías.

Autenticación y Seguridad

- **RF-05**: El sistema debe autenticar a los usuarios mediante **passport-jwt** y **jsonwebtoken**.
- **RF-06**: Las contraseñas deben almacenarse de manera segura utilizando **bcrypt**.
- **RF-07**: El sistema debe limitar las peticiones de los usuarios con **express-rate-limit**.

Gestión de Categorías

- **RF-08:** Los administradores deben poder realizar **CRUD** de categorías (crear, leer, actualizar, eliminar).
- **RF-09:** El sistema debe validar que no existan categorías duplicadas.

Gestión de Películas y Series

- **RF-10:** El sistema debe permitir a los usuarios proponer películas/series con atributos mínimos: título, descripción, categoría, año, imagen opcional.
- **RF-11:** Los administradores deben aprobar las películas/series antes de que sean visibles para los demás usuarios.
- **RF-12:** El sistema debe validar que no existan títulos repetidos.

Gestión de Reseñas y Ratings

- **RF-13:** Los usuarios deben poder crear, editar y eliminar reseñas sobre películas/series.
- **RF-14:** Cada reseña debe incluir: título, comentario y una calificación numérica entre 1 y 10.
- **RF-15:** Los usuarios deben poder dar **like** o **dislike** a las reseñas de otros, pero no a las propias.
- **RF-16:** El sistema debe calcular un **ranking ponderado** de las películas/series en función de:
 - Calificaciones numéricas.
 - Likes/dislikes de las reseñas.
 - Fecha de publicación de la reseña.

Ranking y Listados

- **RF-17:** El sistema debe listar las películas/series ordenadas por popularidad y ranking.
- **RF-18:** El sistema debe permitir filtrar películas/series por categoría.
- **RF-19:** El sistema debe mostrar la vista de detalle de una película/serie con información y reseñas asociadas.

Validaciones y Transacciones

- **RF-20:** El sistema debe validar los datos de entrada en cada endpoint usando **express-validator**.
- **RF-21:** El sistema debe usar **transacciones reales en MongoDB** para operaciones críticas como:
 - Creación de reseñas con rating inicial.
 - Gestión de likes/dislikes.

API y Documentación

- **RF-22:** El sistema debe exponer una **API REST** versionada siguiendo **semver**.
- **RF-23:** La API debe estar documentada utilizando **swagger-ui-express**.
- **RF-24:** El sistema debe permitir la conexión desde el frontend mediante **CORS** configurado correctamente.

2.2 Requerimientos No Funcionales BackEnd

Los **requerimientos no funcionales** establecen las restricciones y cualidades que la aplicación debe cumplir.

Rendimiento

- **RNF-01:** El tiempo de respuesta p95 para consultas de listados no debe superar los **300 ms**.

Seguridad

- **RNF-02:** Las contraseñas deben almacenarse con **hash seguro** utilizando bcrypt con salt.
- **RNF-03:** El sistema debe invalidar tokens JWT expirados y prevenir accesos no autorizados.
- **RNF-04:** El backend debe contar con **rate limiting** para prevenir ataques de fuerza bruta.

Escalabilidad

- **RNF-05:** La arquitectura del sistema debe ser modular para permitir la incorporación de nuevas funcionalidades sin afectar las existentes.

Usabilidad

- **RNF-06:** La interfaz debe mostrar mensajes claros de validación y error provenientes del backend.

Documentación y Transparencia

- **RNF-07:** Todos los endpoints deben estar documentados con **Swagger**.

2.3. Requerimientos FrontEnd

Página de Inicio (Landing / Exploración)

- **HU-01:** El sistema debe mostrar automáticamente un listado inicial de películas y series obtenidas desde la API, accesible para todos los visitantes.
- **HU-02:** El sistema debe permitir al usuario seleccionar una categoría desde un menú o lista y actualizar el listado de películas en pantalla.
- **HU-03:** El sistema debe permitir ordenar el listado según criterios como popularidad, calificación o ranking ponderado, consumiendo datos desde la API.
- **HU-04:** El sistema debe mostrar una vista de detalle que incluya información de la película (título, sinopsis, categoría, año, calificación promedio y reseñas asociadas).

Autenticación y Sesión

- **HU-05:** El sistema debe permitir registrar usuarios almacenando su información en la base de datos. Las contraseñas deben ser cifradas y el correo electrónico debe ser único para evitar duplicados.
- **HU-06:** El sistema debe permitir iniciar sesión validando las credenciales del usuario y generar un token JWT para autenticar las siguientes peticiones.
- **HU-07:** El sistema debe permitir la persistencia de la sesión mientras el token JWT sea válido y renovar la autenticación automáticamente en el frontend.

- **HU-08:** El sistema debe mostrar mensajes de validación en el frontend provenientes del backend para indicar errores en el proceso de registro o inicio de sesión.

Gestión de Contenido (Películas / Series)

- **HU-09:** El sistema debe mostrar un panel de propuestas recibidas donde el administrador pueda aprobar o rechazar cada una de ellas.
- **HU-10:** El sistema debe permitir operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre películas y series desde una interfaz de administrador.
- **HU-11:** El sistema debe mostrar un formulario donde el usuario pueda enviar los datos básicos de una película/serie para que el administrador la revise.
- **HU-12:** El sistema debe sincronizar en tiempo real el catálogo público cada vez que el administrador apruebe una propuesta.

Gestión de Reseñas

- **HU-13:** El sistema debe permitir al usuario autenticado crear y guardar una reseña de una película o serie, asociándola al contenido correspondiente.
- **HU-14:** El sistema debe permitir al usuario autenticado acceder a sus reseñas previas, modificarlas o eliminarlas de forma permanente.
- **HU-15:** El sistema debe permitir a los usuarios autenticados marcar con un like o dislike una reseña publicada.
- **HU-16:** El sistema debe mostrar en cada reseña un contador visible de likes y dislikes acumulados.
- **HU-17:** El sistema debe retener las reseñas creadas por los usuarios en estado pendiente y permitir al administrador aprobarlas o rechazarlas antes de publicarlas.

Gestión de Categorías

- **HU-18:** El sistema debe permitir al administrador gestionar las categorías, asegurando que se reflejen en la plataforma para el correcto filtrado de películas y series.
- **HU-19:** El sistema debe mostrar un menú o listado de categorías que permita a los usuarios filtrar y visualizar películas o series que correspondan a la categoría seleccionada.

Panel de Usuario / Dashboard

- **HU-20:** El sistema debe permitir que cada usuario vea en un panel todas las reseñas que ha creado, con opciones de edición o eliminación.
- **HU-21:** El sistema debe mostrar una sección de perfil donde se reflejen los datos básicos del usuario (nombre, correo, fecha de registro, etc.).
- **HU-22:** El sistema debe permitir al usuario editar sus datos desde el panel de perfil y guardar los cambios en la base de datos.
- **HU-23:** El sistema debe permitir al usuario eliminar su cuenta de manera definitiva desde el panel de usuario.

Experiencia de Usuario

- **HU-24:** El sistema debe mostrar en pantalla los mensajes de error y validación enviados por el backend, de manera clara y entendible para el usuario.
- **HU-25:** El sistema debe adaptar automáticamente el diseño y disposición de la interfaz según el tamaño de pantalla del dispositivo.

3. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN BACKEND

HISTORIA DE USUARIO - Gestión de Usuarios			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-01	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Registro de usuarios		
Descripción			
Como usuario quiero registrarme en el sistema proporcionando mis datos básicos (nombre, correo, contraseña) para poder crear una cuenta y acceder a las funcionalidades de la plataforma.			
Funcionalidad			
El sistema debe permitir registrar usuarios almacenando su información en la base de datos. Las contraseñas deben ser cifradas y los correos electrónicos únicos para evitar duplicidad.			
Criterios de aceptación	1. El sistema debe permitir que un usuario se registre con nombre, correo y contraseña. 2. El sistema debe validar que el correo no esté previamente registrado. 3. El sistema debe almacenar la contraseña cifrada con bcrypt.		
Restricciones			
<div><input type="checkbox"/> No se deben almacenar contraseñas en texto plano.</div> <div><input type="checkbox"/> El correo electrónico debe ser único en la base de datos.</div>			

HISTORIA DE USUARIO - Gestión de Usuarios			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-02	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Inicio de sesión		
Descripción			
Como usuario quiero iniciar sesión en el sistema usando mis credenciales para poder acceder a mis funcionalidades personalizadas.			
Funcionalidad			
El sistema debe autenticar usuarios mediante JWT, generando un token válido para sesiones seguras.			
Criterios de aceptación	<div>1. El sistema debe validar correo y contraseña contra la base de datos.</div> <div>2. El sistema debe generar un token JWT válido tras autenticación exitosa.</div> <div>3. El token debe expirar después de un tiempo definido.</div>		
Restricciones			
<div>Solo los usuarios previamente registrados podrán iniciar sesión.</div> <div>El token debe ser firmado con una clave secreta segura.</div>			

HISTORIA DE USUARIO - Gestión de Usuarios			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-03	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Gestión de roles		
Descripción			
Como administrador quiero que el sistema diferencie entre roles (usuario y administrador) para gestionar permisos de acceso a funcionalidades críticas.			
Funcionalidad			
El sistema debe incluir roles en el modelo de usuario y validar permisos al momento de ejecutar acciones específicas.			
Criterios de aceptación	<div>1. El sistema debe asignar por defecto el rol "usuario" en el registro.</div> <div>2. El sistema debe permitir asignar el rol "administrador" solo desde la base de datos o por otro administrador.</div> <div>3. El sistema debe restringir funcionalidades según el rol.</div>		
Restricciones			
<div><input type="checkbox"/> Un usuario no puede autoproclamarse administrador.</div> <div><input type="checkbox"/> Los endpoints sensibles deben validar rol antes de ejecutar acciones.</div>			

HISTORIA DE USUARIO - Gestión de Usuarios			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-04	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Gestión de películas/series y categorías		
Descripción			
Como administrador quiero aprobar nuevas películas/series y gestionar categorías para controlar el contenido disponible en el sistema.			
Funcionalidad			
El sistema debe permitir que los administradores aprueben, rechacen o eliminen propuestas de películas/series. Además, gestionar categorías mediante operaciones CRUD.			
Criterios de aceptación	<div>1. El sistema debe permitir que el administrador apruebe/rechace películas/series.</div> <div>2. El sistema debe permitir que el administrador cree, edite y elimine categorías.</div> <div>3. El sistema debe validar que no existan categorías duplicadas.</div>		
Restricciones			
<div><input type="checkbox"/> Solo administradores pueden realizar estas acciones.</div> <div><input type="checkbox"/> No se deben registrar categorías con nombres duplicados.</div>			

HISTORIA DE USUARIO - Autenticación y Seguridad			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-01	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Autenticación con JWT		
Descripción			
Como sistema debo autenticar a los usuarios mediante passport-jwt y jsonwebtoken para garantizar la seguridad de las sesiones.			
Funcionalidad			
El sistema debe verificar tokens JWT en cada petición protegida, asegurando que el usuario tenga una sesión válida.			
Criterios de aceptación	<div>1. El sistema debe rechazar peticiones sin token válido.</div> <div>2. El sistema debe permitir acceso solo si el token está vigente.</div> <div>3. El sistema debe identificar al usuario autenticado a partir del token.</div>		
Restricciones			
<div><input type="checkbox"/> Los tokens deben tener tiempo de expiración.</div> <div><input type="checkbox"/> Se debe usar una clave secreta segura para firmar los tokens.</div>			

HISTORIA DE USUARIO - Autenticación y Seguridad			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-02	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Seguridad en almacenamiento de contraseñas		
Descripción			
Como sistema debo almacenar las contraseñas de los usuarios de forma segura para proteger la información sensible.			
Funcionalidad			
El sistema debe usar bcrypt para cifrar contraseñas antes de almacenarlas en la base de datos.			
Criterios de aceptación	<div>1. El sistema debe cifrar contraseñas con bcrypt usando salt.</div> <div>2. El sistema no debe almacenar contraseñas en texto plano.</div> <div>3. El sistema debe validar contraseñas cifradas al inicio de sesión.</div>		
Restricciones			
<div>• La longitud mínima de contraseña debe ser de 8 caracteres.</div>			

HISTORIA DE USUARIO - Autenticación y Seguridad			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-03	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Limitación de peticiones		
Descripción			
Como sistema debo limitar la cantidad de peticiones que un usuario puede realizar para evitar ataques de fuerza bruta y denegación de servicio.			
Funcionalidad			
El sistema debe implementar express-rate-limit para restringir el número de solicitudes permitidas por IP en un periodo definido.			
Criterios de aceptación	<div>1. El sistema debe bloquear al usuario temporalmente al exceder el límite.</div> <div>2. El sistema debe registrar intentos fallidos de autenticación.</div> <div>3. El sistema debe enviar un mensaje de error al exceder las peticiones permitidas.</div>		
Restricciones			
<div>• Los límites deben configurarse según las buenas prácticas de seguridad.</div>			

HISTORIA DE USUARIO - Gestión de Categorías			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-05	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	CRUD de categorías		
Descripción			
Como administrador quiero poder crear, leer, actualizar y eliminar categorías para organizar las películas y series del sistema.			
Funcionalidad			
El sistema debe permitir operaciones CRUD sobre categorías en la base de datos.			
Criterios de aceptación	<div>1. El sistema debe permitir que el administrador cree nuevas categorías.</div> <div>2. El sistema debe permitir que el administrador actualice categorías existentes.</div> <div>3. El sistema debe permitir que el administrador elimine categorías.</div>		
Restricciones			
<div>• No se deben crear categorías duplicadas.</div>			

HISTORIA DE USUARIO - Gestión de Categorías			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-04	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Validación de categorías duplicadas		
Descripción			
Como sistema debo validar que no existan categorías duplicadas para garantizar consistencia en la base de datos.			
Funcionalidad			
El sistema debe verificar nombres de categorías antes de su inserción o actualización.			
Criterios de aceptación	1. El sistema debe rechazar categorías con nombres duplicados. 2. El sistema debe permitir solo categorías con nombres únicos.		
Restricciones			
• La validación debe ser case-insensitive.			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-06	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Propuesta de películas/series		
Descripción			
Como usuario quiero proponer películas o series con atributos mínimos para contribuir con contenido en la plataforma.			
Funcionalidad			
El sistema debe permitir que un usuario envíe una propuesta de película/serie con título, descripción, categoría, año y opcionalmente imagen.			
Criterios de aceptación	1. El sistema debe permitir que el usuario registre propuestas con campos mínimos. 2. El sistema debe almacenar las propuestas en estado pendiente hasta aprobación.		
Restricciones			
• Las propuestas no deben publicarse sin aprobación de administrador.			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-07	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Aprobación de películas/series		
Descripción			
Como administrador quiero aprobar o rechazar las propuestas de películas/series para garantizar calidad y consistencia en la plataforma.			
Funcionalidad			
El sistema debe permitir que los administradores revisen las propuestas y cambien su estado a aprobado o rechazado.			
Criterios de aceptación	<div>1. El sistema debe mostrar propuestas pendientes a los administradores.</div> <div>2. El sistema debe permitir aprobar o rechazar propuestas.</div> <div>3. El sistema debe notificar al usuario del estado de su propuesta.</div>		
Restricciones			
<div>Solo los administradores pueden aprobar/rechazar propuestas.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-05	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Validación de títulos repetidos		
Descripción			
Como sistema debo validar que no existan títulos repetidos de películas o series para evitar duplicados.			
Funcionalidad			
El sistema debe verificar títulos de películas y series antes de insertarlos en la base de datos.			
Criterios de aceptación	<div>1. El sistema debe rechazar títulos ya registrados.</div> <div>2. El sistema debe permitir títulos únicos sin importar mayúsculas/minúsculas.</div>		
Restricciones			
<div>• Validación case-insensitive.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-08	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	CRUD de reseñas		
Descripción			
Como usuario quiero poder crear, editar y eliminar reseñas sobre películas/series para compartir mi opinión.			
Funcionalidad			
El sistema debe permitir que los usuarios realicen operaciones CRUD sobre reseñas asociadas a películas/series.			
Criterios de aceptación	<div>1. El sistema debe permitir crear reseñas con título, comentario y calificación.</div> <div>2. El sistema debe permitir editar solo reseñas creadas por el usuario.</div> <div>3. El sistema debe permitir eliminar reseñas propias.</div>		
Restricciones			
<div>• Un usuario no puede modificar reseñas de otros.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-09	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Estructura de reseñas		
Descripción			
Como usuario quiero que cada reseña incluya un título, comentario y calificación numérica entre 1 y 10 para expresar mejor mi opinión.			
Funcionalidad			
El sistema debe estructurar las reseñas con los atributos requeridos.			
Criterios de aceptación	<div>1. El sistema debe validar que la calificación esté entre 1 y 10.</div> <div>2. El sistema debe exigir título y comentario en cada reseña.</div>		
Restricciones			
<div>• Calificaciones fuera del rango deben ser rechazadas.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-10	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Likes/dislikes en reseñas		
Descripción			
Como usuario quiero dar like o dislike a reseñas de otros para expresar si me resultan útiles o no.			
Funcionalidad			
El sistema debe permitir que un usuario vote reseñas de terceros, pero no las propias.			
Criterios de aceptación	<div>1. El sistema debe permitir a un usuario dar like/dislike a reseñas de otros.</div> <div>2. El sistema debe impedir que un usuario vote su propia reseña.</div> <div>3. El sistema debe permitir actualizar o retirar un voto.</div>		
Restricciones			
<div>• Un usuario no puede emitir múltiples votos sobre la misma reseña.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-06	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Ranking ponderado		
Descripción			
Como sistema debo calcular un ranking ponderado de películas/series según calificaciones, likes/dislikes y fecha de reseña para mostrar popularidad real.			
Funcionalidad			
El sistema debe procesar automáticamente los datos de reseñas y votos para generar un ranking de películas/series.			
Criterios de aceptación	<div>1. El sistema debe ponderar calificaciones, likes/dislikes y fecha.</div> <div>2. El sistema debe actualizar el ranking en tiempo real o bajo demanda.</div> <div>3. El sistema debe mostrar las películas/series mejor posicionadas primero.</div>		
Restricciones			
<div>• El algoritmo de ranking debe ser consistente y documentado.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-11	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Listado por popularidad		
Descripción			
Como usuario quiero ver un listado de películas/series ordenadas por popularidad y ranking para conocer las más destacadas.			
Funcionalidad			
El sistema debe listar automáticamente películas/series con base en su ranking.			
Criterios de aceptación	1. El sistema debe ordenar el listado por ranking. 2. El sistema debe incluir solo contenido aprobado.		
Restricciones			
• El listado no debe incluir propuestas no aprobadas.			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-12	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Filtrado por categoría		
Descripción			
Como usuario quiero filtrar películas/series por categoría para encontrar contenido de mi interés.			
Funcionalidad			
El sistema debe permitir aplicar filtros de categoría en el listado de películas/series.			
Criterios de aceptación	<div>1. El sistema debe listar solo películas/series de la categoría seleccionada.</div> <div>2. El sistema debe combinar filtros con ranking.</div>		
Restricciones			
<div>Solo categorías existentes podrán usarse como filtros.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF-13	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Vista de detalle		
Descripción			
Como usuario quiero ver la vista detallada de una película/serie con información y reseñas asociadas para conocer su calidad.			
Funcionalidad			
El sistema debe mostrar datos completos de la película/serie, incluyendo reseñas y calificaciones.			
Criterios de aceptación	<div>1. El sistema debe mostrar detalles de título, descripción, año, categoría.</div> <div>2. El sistema debe listar reseñas asociadas.</div>		
Restricciones			
<div>Solo películas/series aprobadas deben tener vista de detalle.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-07	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Validación de entrada		
Descripción			
Como sistema debo validar datos en cada endpoint para garantizar consistencia y seguridad en la información.			
Funcionalidad			
El sistema debe usar express-validator para validar los datos de entrada antes de procesarlos.			
Criterios de aceptación	<div>1. El sistema debe rechazar datos inválidos o incompletos.</div> <div>2. El sistema debe retornar mensajes claros de error en validación.</div>		
Restricciones			
<div>• Los datos inválidos no deben llegar a la base de datos.</div>			

HISTORIA DE USUARIO - Gestión de Películas y Series			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-08	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Transacciones críticas		
Descripción			
Como sistema debo usar transacciones reales en MongoDB para operaciones críticas como creación de reseñas con rating inicial y gestión de likes/dislikes.			
Funcionalidad			
El sistema debe garantizar que las operaciones críticas sean atómicas, asegurando consistencia.			
Criterios de aceptación	<div>1. El sistema debe ejecutar reseñas y calificaciones en una única transacción.</div> <div>2. El sistema debe revertir transacciones en caso de error.</div>		
Restricciones			
<div>• Deben usarse transacciones ACID en MongoDB.</div>			

HISTORIA DE USUARIO - API y Documentación			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-09	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	API REST versionada		
Descripción			
Como sistema debo exponer una API REST versionada siguiendo semver para mantener compatibilidad y escalabilidad.			
Funcionalidad			
El sistema debe ofrecer endpoints versionados (/api/v1, /api/v2).			
Criterios de aceptación	<div>1. El sistema debe exponer endpoints bajo una versión definida.</div> <div>2. El sistema debe mantener compatibilidad con versiones anteriores.</div>		
Restricciones			
<div>• Los cambios incompatibles deben reflejarse en una nueva versión.</div>			

HISTORIA DE USUARIO - API y Documentación			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RNF-01	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Documentación con Swagger		
Descripción			
Como sistema debo documentar la API utilizando swagger-ui-express para facilitar la comprensión y pruebas de los endpoints.			
Funcionalidad			
El sistema debe generar documentación interactiva disponible en un endpoint (/docs).			
Criterios de aceptación	<div>1. El sistema debe mostrar todos los endpoints documentados en Swagger.</div> <div>2. El sistema debe permitir ejecutar pruebas desde la interfaz Swagger.</div>		
Restricciones			
<div>• La documentación debe actualizarse con cada cambio de API.</div>			

HISTORIA DE USUARIO - API y Documentación			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	AS-10	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Configuración de CORS		
Descripción			
Como sistema debo permitir la conexión desde el frontend mediante CORS correctamente configurado para garantizar seguridad y accesibilidad.			
Funcionalidad			
El sistema debe habilitar CORS para solicitudes desde dominios específicos.			
Criterios de aceptación	<div>1. El sistema debe permitir conexiones desde el dominio del frontend.</div> <div>2. El sistema debe rechazar peticiones de orígenes no autorizados.</div>		
Restricciones			
<div>• No se debe habilitar CORS global para todos los orígenes en producción.</div>			

4. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN FRONTEND

HISTORIA DE USUARIO – Página de Inicio			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-01	Actor	Usuario visitante
NOMBRE DEL REQUERIMIENTO	Listado inicial de películas/series		
Descripción			
Como usuario visitante quiero ver un listado de películas/series precargadas en la primera página, para conocer el contenido disponible sin necesidad de registrarme.			
Funcionalidad			
El sistema debe mostrar automáticamente un listado inicial de películas y series obtenidas desde la API, accesible para todos los visitantes.			
Criterios de aceptación	<div>1. El sistema debe mostrar un listado visible en la página de inicio con al menos título, imagen y calificación general.</div> <div>2. El sistema debe consumir la API para obtener los datos iniciales.</div> <div>3. El usuario debe poder visualizar el contenido sin estar registrado ni iniciar sesión.</div>		
Restricciones			
<div><div>- El listado debe cargarse dinámicamente desde el backend o una API externa.</div><div>- No se debe requerir autenticación para visualizar el contenido inicial.</div></div>			

HISTORIA DE USUARIO – Página de Inicio			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-02	Actor	Usuario visitante
NOMBRE DEL REQUERIMIENTO	Filtrado por categoría		
Descripción			
Como usuario visitante quiero poder filtrar las películas por categoría, para encontrar más rápido el contenido de mi interés.			
Funcionalidad			
El sistema debe permitir al usuario seleccionar una categoría desde un menú o lista y actualizar el listado de películas en pantalla.			
Criterios de aceptación	<div>1. El sistema debe mostrar un menú o lista de categorías disponibles.</div> <div>2. Al seleccionar una categoría, el listado debe actualizarse con solo las películas/series que correspondan.</div> <div>3. El filtro debe aplicarse sin necesidad de recargar manualmente toda la página.</div>		
Restricciones			
<div><div></div><div>- Solo se deben mostrar categorías aprobadas por un administrador.</div></div> <div><div></div><div>- El filtrado debe estar sincronizado con los datos de la API.</div></div>			

HISTORIA DE USUARIO – Página de Inicio			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-03	Actor	Usuario visitante
NOMBRE DEL REQUERIMIENTO	Ordenamiento por popularidad o ranking		
Descripción			
Como usuario visitante quiero poder ordenar el listado por popularidad o ranking, para identificar las películas mejor valoradas.			
Funcionalidad			
El sistema debe permitir ordenar el listado según criterios como popularidad, calificación o ranking ponderado, consumiendo datos desde la API.			
Criterios de aceptación	<div>1. El usuario debe poder elegir entre distintos criterios de orden (ej. popularidad, calificación, ranking).</div> <div>2. El sistema debe actualizar el listado automáticamente tras seleccionar un criterio.</div> <div>3. El orden aplicado debe mantenerse mientras el usuario navega en la página actual.</div>		
Restricciones			
<div><div></div><div>- El ranking debe calcularse de acuerdo con la lógica definida en el backend.</div></div> <div><div></div><div>- El frontend solo debe mostrar los resultados ordenados, sin recalcular rankings.</div></div>			

HISTORIA DE USUARIO – Página de Inicio			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-04	Actor	Usuario visitante
NOMBRE DEL REQUERIMIENTO	Vista de detalle de película		
Descripción			
Como usuario visitante quiero poder entrar al detalle de una película, para ver su sinopsis, calificación y reseñas.			
Funcionalidad			
El sistema debe mostrar una vista de detalle que incluya información de la película (título, sinopsis, categoría, año, calificación promedio y reseñas asociadas).			
Criterios de aceptación	<div>1. El sistema debe mostrar la información detallada de la película seleccionada.</div> <div>2. El detalle debe incluir las reseñas ya existentes y su calificación.</div> <div>3. El usuario debe poder regresar fácilmente al listado general desde la vista de detalle.</div>		
Restricciones			
<div><div></div><div>- El detalle debe obtenerse dinámicamente desde la API.</div></div> <div><div></div><div>- No se debe permitir la creación de reseñas sin haber iniciado sesión.</div></div>			

HISTORIA DE USUARIO – Página de Inicio			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-05	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Registro de usuario		
Descripción			
Como usuario quiero poder registrarme proporcionando nombre, correo y contraseña, para tener una cuenta en la plataforma.			
Funcionalidad			
El sistema debe permitir registrar usuarios almacenando su información en la base de datos. Las contraseñas deben ser cifradas y el correo electrónico debe ser único para evitar duplicados.			
Criterios de aceptación	<div>1. El sistema debe permitir el registro de usuarios con nombre, correo y contraseña.</div> <div>2. El sistema debe validar que el correo no se encuentre previamente registrado.</div> <div>3. La contraseña debe almacenarse de manera segura usando bcrypt.</div>		
Restricciones			
<div><div></div><div>- No se deben almacenar contraseñas en texto plano.</div></div> <div><div></div><div>- El correo electrónico debe ser único en la base de datos.</div></div>			

HISTORIA DE USUARIO – Autenticación			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-06	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Inicio de sesión		
Descripción			
Como usuario quiero poder iniciar sesión con mi correo y contraseña, para acceder a funcionalidades restringidas como crear reseñas.			
Funcionalidad			
El sistema debe permitir iniciar sesión validando las credenciales del usuario y generar un token JWT para autenticar las siguientes peticiones.			
Criterios de aceptación	<div>1. El sistema debe validar correo y contraseña ingresados.</div> <div>2. El sistema debe devolver un token JWT válido en caso de credenciales correctas.</div> <div>3. El sistema debe rechazar el inicio de sesión si las credenciales no son válidas.</div> <div>4. El diseño de la página de inicio de sesión debe de ser amigable e intuitivo con el usuario para su fácil comprensión.</div>		
Restricciones			
<div><div></div><div>- El token JWT debe tener un tiempo de expiración definido.</div></div> <div><div></div><div>- No se debe permitir el acceso sin un token válido.</div></div>			

HISTORIA DE USUARIO – Autenticación			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-07	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Mantener sesión activa		
Descripción			
Como usuario autenticado quiero que mi sesión se mantenga activa hasta cerrar sesión, para no tener que loguearme cada vez.			
Funcionalidad			
El sistema debe permitir la persistencia de la sesión mientras el token JWT sea válido y renovar la autenticación automáticamente en el frontend.			
Criterios de aceptación	<div>1. El sistema debe mantener la sesión activa mientras el token JWT no haya expirado</div> <div>2. El usuario debe poder cerrar sesión manualmente desde la interfaz.</div> <div>3. El sistema debe redirigir al login cuando el token haya caducado.</div> <div>4. Se debe de mostrar un mensaje amigable al usuario para direccionarlo a que inicie sesión nuevamente.</div>		
Restricciones			
<div><div></div><div>- El token no debe almacenarse en texto plano en el navegador (debe usarse almacenamiento seguro como localStorage/sessionStorage).</div><div>- Al expirar el token, el usuario debe autenticarse nuevamente.</div></div>			

HISTORIA DE USUARIO – Autenticación			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-08	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Mensajes de error y validación en autenticación		
Descripción			
Como usuario quiero recibir mensajes claros si mis credenciales son inválidas o si falta algún dato en el registro/login, para saber cómo corregirlo.			
Funcionalidad			
El sistema debe mostrar mensajes de validación en el frontend provenientes del backend para indicar errores en el proceso de registro o inicio de sesión.			
Criterios de aceptación	<div>1. El sistema debe mostrar un mensaje si el correo no está registrado.</div> <div>2. El sistema debe mostrar un mensaje si la contraseña no coincide.</div> <div>3. El sistema debe mostrar un mensaje si algún campo obligatorio no fue completado.</div>		
Restricciones			
<div><div></div><div>- No se deben mostrar mensajes que revelen información sensible del sistema.</div><div></div><div>- Los mensajes deben ser comprensibles y en un lenguaje claro para el usuario.</div></div>			

HISTORIA DE USUARIO – Gestión de Contenido: Panel de aprobación			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-09	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Panel para aprobar o rechazar propuestas de películas/series		
Descripción			
Como administrador quiero tener un panel donde pueda aprobar o rechazar las películas propuestas por usuarios, para mantener la calidad del catálogo.			
Funcionalidad			
El sistema debe mostrar un panel de propuestas recibidas donde el administrador pueda aprobar o rechazar cada una de ellas.			
Criterios de aceptación	<div>1. El administrador debe ver un listado de todas las propuestas enviadas por usuarios.</div> <div>2. Cada propuesta debe mostrar al menos: título, imagen, descripción y datos básicos.</div> <div>3. Debe existir un botón de aprobar y otro de rechazar por cada propuesta.</div> <div>4. El sistema debe registrar la decisión tomada por el administrador.</div>		
Restricciones			
<div><div></div><div>- Solo los administradores autenticados pueden acceder a este panel.</div><div>- La información debe provenir de la base de datos y actualizarse en tiempo real.</div></div>			

HISTORIA DE USUARIO – Gestión de Contenido: Panel de aprobación			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-10	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Crear, actualizar o eliminar películas/series desde el frontend		
Descripción			
Como administrador quiero poder crear, actualizar o eliminar películas/series desde el frontend, para gestionar el catálogo de manera sencilla.			
Funcionalidad			
El sistema debe permitir operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre películas y series desde una interfaz de administrador.			
Criterios de aceptación	<div>1. El administrador debe poder registrar una nueva película/serie desde un formulario.</div> <div>2. Debe existir la opción de editar la información de cualquier película/serie.</div> <div>3. Debe existir la opción de eliminar una película/serie.</div> <div>4. Los cambios deben reflejarse de forma inmediata en el catálogo.</div>		
Restricciones			
<div><div></div><div>- Solo los administradores autenticados pueden realizar estas acciones.</div></div> <div><div></div><div>- Todas las operaciones deben validarse antes de guardarse en la base de datos.</div></div>			

HISTORIA DE USUARIO – Propuesta de contenido por usuarios			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-11	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Proponer una nueva película/serie mediante formulario		
Descripción			
Como usuario autenticado quiero poder proponer una nueva película/serie llenando un formulario, para contribuir al catálogo.			
Funcionalidad			
El sistema debe mostrar un formulario donde el usuario pueda enviar los datos básicos de una película/serie para que el administrador la revise.			
Criterios de aceptación	<div>1. El formulario debe incluir campos: título, imagen, descripción, género y año.</div> <div>2. Al enviar, la propuesta debe almacenarse en la base de datos en estado pendiente.</div> <div>3. El usuario debe recibir una confirmación de que su propuesta fue enviada.</div>		
Restricciones			
<div><div></div><div>- Solo los usuarios autenticados pueden proponer nuevas películas/series.</div><div></div><div>- Las propuestas no se muestran automáticamente en el catálogo, deben ser aprobadas por un administrador.</div></div>			

HISTORIA DE USUARIO – Actualización automática del catálogo			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-12	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Actualización automática del listado público tras aprobación		
Descripción			
Como administrador quiero que al aprobar una película/serie se actualice automáticamente en el listado público, para que otros usuarios la vean sin recargar manualmente.			
Funcionalidad			
El sistema debe sincronizar en tiempo real el catálogo público cada vez que el administrador apruebe una propuesta.			
Criterios de aceptación	<div>1. Al aprobar una película/serie, debe añadirse de inmediato al listado público visible.</div> <div>2. Los usuarios visitantes o autenticados deben ver el cambio sin necesidad de recargar la página.</div> <div>3. El estado de la propuesta debe cambiar de pendiente a aprobada en la base de datos.</div>		
Restricciones			
<div><div></div><div>- La funcionalidad debe implementarse con un sistema de actualización en tiempo real (ejemplo: WebSockets o notificaciones push).</div><div>- Solo las películas/series aprobadas deben mostrarse en el catálogo público.</div></div>			

HISTORIAS DE USUARIO – Gestión de Reseñas			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-13	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Crear reseña de película/serie		
Descripción			
Como usuario autenticado quiero poder escribir una reseña de una película con título, comentario y calificación, para compartir mi opinión.			
Funcionalidad			
El sistema debe permitir al usuario autenticado crear y guardar una reseña de una película o serie, asociándola al contenido correspondiente.			
Criterios de aceptación	<div>1. El usuario debe poder ingresar título, comentario y calificación (numérica o en estrellas).</div> <div>2. La reseña debe quedar asociada a la película/serie seleccionada.</div> <div>3. Solo los usuarios autenticados pueden crear reseñas.</div>		
Restricciones			
<div><div></div><div>- La reseña no debe mostrarse públicamente hasta que sea aprobada</div></div> <div><div></div><div>- No se debe permitir reseñas vacías o sin calificación.</div></div>			

HISTORIAS DE USUARIO – Gestión de Reseñas			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-14	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Editar o eliminar reseñas propias		
Descripción			
Como usuario autenticado quiero poder editar o eliminar mis reseñas, para mantenerlas actualizadas o borrarlas si lo deseo.			
Funcionalidad			
El sistema debe permitir al usuario autenticado acceder a sus reseñas previas, modificarlas o eliminarlas de forma permanente.			
Criterios de aceptación	<div>1. Solo el autor de la reseña podrá editar o eliminar su propio contenido.</div> <div>2. Al editar, debe poder modificar el título, comentario y calificación.</div> <div>3. Al eliminar, la reseña desaparecerá del catálogo (o quedará marcada como eliminada en base de datos).</div>		
Restricciones			
<div><div></div><div>- No se deben permitir ediciones que dejen la reseña vacía.</div><div>- El administrador puede seguir viendo las reseñas eliminadas en el backend para auditoría.</div></div>			

HISTORIAS DE USUARIO – Gestión de Reseñas			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-15	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Like/Dislike en reseñas		
Descripción			
Como usuario autenticado quiero poder dar like o dislike a reseñas de otros, para apoyar o no sus comentarios.			
Funcionalidad			
El sistema debe permitir a los usuarios autenticados marcar con un like o dislike una reseña publicada.			
Criterios de aceptación	<div>1. Cada usuario podrá dar solo un voto por reseña (like o dislike).</div> <div>2. El usuario puede cambiar su voto (de like a dislike o viceversa).</div> <div>3. Los votos deben actualizarse en tiempo real en la interfaz.</div>		
Restricciones			
<div><div></div>- Solo usuarios autenticados podrán interactuar con reseñas.</div> <div><div></div>- Un mismo usuario no puede votar varias veces en la misma dirección en una reseña.</div>			

HISTORIAS DE USUARIO – Gestión de Reseñas			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-16	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Visualización de likes/dislikes en reseñas		
Descripción			
Como usuario autenticado quiero ver cuántos likes/dislikes tiene cada reseña, para identificar cuáles son más valoradas.			
Funcionalidad			
El sistema debe mostrar en cada reseña un contador visible de likes y dislikes acumulados.			
Criterios de aceptación	<div>1. El conteo de likes y dislikes debe mostrarse junto a cada reseña.</div> <div>2. El conteo debe actualizarse automáticamente al registrarse un nuevo voto.</div> <div>3. Debe ser visible para todos los usuarios autenticados.</div>		
Restricciones			
<div><div></div><div>- Los conteos deben estar protegidos contra manipulaciones de votos fraudulentos.</div></div>			

HISTORIAS DE USUARIO – Gestión de Reseñas			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-17	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Moderación de reseñas		
Descripción			
Como administrador quiero que las reseñas creadas pasen primero por aprobación antes de mostrarse públicamente, para moderar el contenido.			
Funcionalidad			
El sistema debe retener las reseñas creadas por los usuarios en estado pendiente y permitir al administrador aprobarlas o rechazarlas antes de publicarlas.			
Criterios de aceptación	<div>1. Las reseñas no deben mostrarse en la página pública hasta que sean aprobadas.</div> <div>2. El administrador debe contar con un panel donde pueda ver, aprobar o rechazar reseñas pendientes.</div> <div>3. Una vez aprobada, la reseña debe aparecer automáticamente en la película/serie correspondiente.</div>		
Restricciones			
<div><div></div><div>- Solo los administradores pueden aprobar o rechazar reseñas.</div><div>- Las reseñas rechazadas no deben mostrarse públicamente, pero pueden almacenarse para auditoría.</div></div>			

HISTORIAS DE USUARIO – Gestión de Categorías			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-18	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Gestión de categorías en el panel administrativo		
Descripción			
Como administrador quiero poder crear, editar y eliminar categorías desde un panel, para organizar el contenido en la plataforma.			
Funcionalidad			
El sistema debe permitir al administrador gestionar las categorías, asegurando que se reflejen en la plataforma para el correcto filtrado de películas y series.			
Criterios de aceptación	<div>1. El administrador debe poder crear nuevas categorías con un nombre único.</div> <div>2. El administrador debe poder editar categorías existentes modificando su nombre.</div> <div>3. El administrador debe poder eliminar categorías, y el sistema debe manejar correctamente las películas/series asociadas a ellas (por ejemplo, reasignarlas o dejarlas sin categoría).</div> <div>4. Los cambios deben reflejarse inmediatamente en la plataforma.</div>		
Restricciones			
<div><div></div><div>- Solo los administradores pueden acceder al panel de gestión de categorías.</div><div>- No se debe permitir duplicar nombres de categorías.</div><div>- No se debe permitir eliminar categorías críticas del sistema si están configuradas como obligatorias.</div></div>			

HISTORIAS DE USUARIO – Gestión de Categorías			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-19	Actor	Usuario visitante
NOMBRE DEL REQUERIMIENTO	Filtrado por categorías		
Descripción			
Como usuario visitante quiero poder filtrar películas/series por categoría seleccionando desde un menú, para ver contenido específico.			
Funcionalidad			
El sistema debe mostrar un menú o listado de categorías que permita a los usuarios filtrar y visualizar películas o series que correspondan a la categoría seleccionada.			
Criterios de aceptación	<div>1. El sistema debe mostrar todas las categorías disponibles en un menú o listado.</div> <div>2. Al seleccionar una categoría, el sistema debe mostrar únicamente las películas/series correspondientes.</div> <div>3. Debe existir la opción de volver a ver todo el contenido sin filtros.</div>		
Restricciones			
<div><div></div><div>- Solo se mostrarán categorías que estén activas y aprobadas por el administrador.</div></div> <div><div></div><div>- Los usuarios visitantes podrán usar el filtrado sin necesidad de autenticación.</div></div>			

HISTORIAS DE USUARIO – Panel de Usuario / Dashboard			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-20	Actor	Usuario visitante
NOMBRE DEL REQUERIMIENTO	Panel de reseñas del usuario		
Descripción			
Como usuario autenticado, quiero tener un panel donde pueda ver todas mis reseñas creadas, para gestionarlas fácilmente.			
Funcionalidad			
El sistema debe permitir que cada usuario vea en un panel todas las reseñas que ha creado, con opciones de edición o eliminación.			
Criterios de aceptación	<div>1. El panel debe mostrar todas las reseñas asociadas al usuario autenticado.</div> <div>2. El usuario debe tener la opción de editar una reseña existente.</div> <div>3. El usuario debe poder eliminar una reseña creada previamente.</div>		
Restricciones			
<div><div></div><div>- Solo el usuario autenticado podrá ver y gestionar sus propias reseñas.</div></div> <div><div></div><div>- No podrá modificar ni eliminar reseñas de otros usuarios.</div></div>			

HISTORIAS DE USUARIO – Panel de Usuario / Dashboard			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-21	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Visualización de perfil		
Descripción			
Como usuario autenticado, quiero poder ver mi perfil con mis datos básicos, para tener control sobre mi cuenta.			
Funcionalidad			
El sistema debe mostrar una sección de perfil donde se reflejen los datos básicos del usuario (nombre, correo, fecha de registro, etc.).			
Criterios de aceptación	<div>1. El perfil debe mostrar la información básica del usuario.</div> <div>2. -Los datos deben estar protegidos y visibles solo para el usuario autenticado.</div>		
Restricciones			
<div><div></div><div>- No se deben mostrar datos sensibles como contraseñas.</div></div>			

HISTORIAS DE USUARIO – Panel de Usuario / Dashboard			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-22	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Actualización de datos del perfil		
Descripción			
Como usuario autenticado, quiero poder actualizar mis datos (nombre, correo, contraseña), para mantener mi cuenta al día.			
Funcionalidad			
El sistema debe permitir al usuario editar sus datos desde el panel de perfil y guardar los cambios en la base de datos.			
Criterios de aceptación	<div>1. El usuario debe poder actualizar su nombre, correo y contraseña.</div> <div>2. Los cambios deben reflejarse inmediatamente en su perfil.</div> <div>3. El sistema debe validar los datos antes de guardarlos.</div>		
Restricciones			
<div><div></div><div>- La contraseña debe cumplir con criterios de seguridad mínimos (longitud, caracteres).</div></div> <div><div></div><div>El correo no debe estar repetido en la base de datos.</div></div>			

HISTORIAS DE USUARIO – Panel de Usuario / Dashboard			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-23	Actor	Usuario autenticado
NOMBRE DEL REQUERIMIENTO	Eliminación de cuenta		
Descripción			
Como usuario, quiero poder eliminar mi cuenta desde el panel, para darme de baja si lo deseo.			
Funcionalidad			
El sistema debe permitir al usuario eliminar su cuenta de manera definitiva desde el panel de usuario.			
Criterios de aceptación	<div>1. El sistema debe solicitar confirmación antes de eliminar la cuenta.</div> <div>2. Al eliminar la cuenta, todos los datos asociados al usuario deben eliminarse o anonimizarse.</div> <div>3. El usuario debe recibir un mensaje de confirmación de que su cuenta fue eliminada.</div>		
Restricciones			
<div><div></div><div>- Una cuenta eliminada no puede ser recuperada.</div></div> <div><div></div><div>Las reseñas asociadas pueden ser eliminadas o mantenerse anonimizadas según la política de la plataforma.</div></div>			

HISTORIAS DE USUARIO – Experiencia de Usuario			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-24	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Mensajes de error y validación en frontend		
Descripción			
Como usuario, quiero que el frontend muestre mensajes de error o validación que provienen del backend, para entender por qué una acción no se completó.			
Funcionalidad			
El sistema debe mostrar en pantalla los mensajes de error y validación enviados por el backend, de manera clara y entendible para el usuario.			
Criterios de aceptación	<div>1. Los mensajes deben mostrarse en el lugar correspondiente de la interfaz (ejemplo: debajo del campo con error o en un alert general).</div> <div>2. Los mensajes deben ser legibles y evitar lenguaje técnico complejo.</div> <div>3. El sistema debe manejar tanto errores de validación de formularios como errores generales (ejemplo: conexión fallida).</div>		
Restricciones			
<div><div></div><div>- Los mensajes deben mantenerse consistentes en estilo y diseño en toda la plataforma.</div></div> <div><div></div><div>- El sistema no debe mostrar información sensible del backend al usuario.</div></div>			

HISTORIAS DE USUARIO – Experiencia de Usuario			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-25	Actor	Usuario
NOMBRE DEL REQUERIMIENTO	Responsividad del frontend		
Descripción			
Como usuario, quiero que el frontend sea responsivo y usable desde móvil, tablet o PC, para navegar cómodamente desde cualquier dispositivo.			
Funcionalidad			
El sistema debe adaptar automáticamente el diseño y disposición de la interfaz según el tamaño de pantalla del dispositivo.			
Criterios de aceptación	<div>1. El contenido debe ser legible y navegable en pantallas pequeñas sin necesidad de hacer zoom.</div> <div>2. Los menús deben convertirse en menús desplegables o hamburguesa en móviles.</div> <div>3. Las imágenes, tablas y elementos gráficos deben ajustarse de forma proporcional al tamaño de pantalla.</div>		
Restricciones			
<div><div></div><div>- Debe cumplirse con principios de diseño responsivo (CSS grid, flexbox, media queries).</div><div></div><div>Se debe garantizar compatibilidad con navegadores modernos en dispositivos móviles y de escritorio.</div></div>			

5. METODOLOGÍA

Para el desarrollo del proyecto **BrisaacFlix** se aplicó el marco de desarrollo de trabajo **SCRUM**, la cual permitió organizar y planificar el trabajo de manera iterativa e incremental, facilitando la entrega continua de valor y la colaboración del equipo.

Roles definidos en el equipo:

- **Product Owner (PO) Brian Suarez:** Encargado de definir las prioridades, gestionar el Product Backlog y asegurar que el proyecto cumpla con las necesidades del usuario final.
- **Scrum Master (SM) Isaac Medina:** Responsable de garantizar la correcta aplicación de la metodología, facilitando las reuniones y eliminando impedimentos.
- **Development Team Brian Suarez y Isaac Medina:** Equipo encargado de implementar las funcionalidades tanto en frontend como en backend, asegurando calidad y cumplimiento de las historias de usuario.

Eventos de SCRUM implementados:




- **Sprint Planning:** Se definieron las tareas e historias de usuario a desarrollar en cada sprint, priorizadas por valor al usuario.
- **Daily Stand Up:** Reuniones cortas diarias para sincronizar avances, identificar bloqueos y ajustar el plan de trabajo.
- **Sprint Review:** Al final de cada sprint se presentaron los avances y entregables al equipo y PO para obtener retroalimentación.

6. EVIDENCIA DE PLANTEAMIENTO DE PLATAFORMA DE TRABAJO



Como parte del trabajo colaborativo en equipo, se documentaron y aplicaron las siguientes evidencias:

- **Repositorio del proyecto:**
[GitHub - BrisaacFlix \(Ejemplo de repositorio\)](#)
- **Reuniones SCRUM:**

26 de agosto de 2025

-  **Isaac Medina** ha iniciado una llamada que ha durado unos segundos. 26/8/25, 15:56
-  **Isaac Medina** ha iniciado una llamada que ha durado unos segundos. 26/8/25, 18:37
-  **Isaac Medina** ha iniciado una llamada que ha durado una hora. 26/8/25, 18:39

27 de agosto de 2025

-  **Isaac Medina** ha iniciado una llamada que ha durado unos segundos. 27/8/25, 14:58
-  **Isaac Medina** ha iniciado una llamada que ha durado una hora. 27/8/25, 15:46






Isaac Medina 27/8/25, 17:04






SCRUM_BrisaacFlix.docx
7.09 MB

28 de agosto de 2025

-  **Brian Fair Suarez Porras** ha iniciado una llamada que ha durado una hora. 28/8/25, 15:49
-  Tienes una llamada perdida de **Brian Fair Suarez Porras** que ha durado 3 minutos.
28/8/25, 17:28
-  **Brian Fair Suarez Porras** ha iniciado una llamada que ha durado una hora. 28/8/25, 17:41

29 de agosto de 2025

-  **Brian Fair Suarez Porras** ha iniciado una llamada que ha durado 6 minutos. 29/8/25, 14:26
-  **Brian Fair Suarez Porras** ha iniciado una llamada que ha durado 21 minutos.
29/8/25, 14:34
-  **Isaac Medina** ha iniciado una llamada que ha durado una hora. 29/8/25, 16:39



- **Tablero de trabajo (Scrum Board):**

Se utilizó una herramienta de gestión GitHub Projects donde se documentaron:

- Historias de usuario priorizadas.
- Tareas asignadas con responsables.
- Fechas estimadas de entrega.
- Estados de avance (To Do, In Progress, Done).
- Capturas de las diferentes etapas del proyecto como evidencia del cumplimiento de SCRUM:

jhonisaacMedinaMendoza08 / Projects / BrisaacFlix

BrisaacFlix

Backend Frontend + New view

Filter by keyword or by field

Discard

Todo 13
This item hasn't been started

- BrisaacFlix-backend #1
RF-01 - Registro de usuarios
- BrisaacFlix-backend #2
RF-02 - Inicio de sesión
- BrisaacFlix-backend #3
RF-03 - Gestión de roles
- BrisaacFlix-backend #4
RF-04 - Gestión de películas/series y categorías
- BrisaacFlix-backend #5
RF-05 - CRUD de categorías
- BrisaacFlix-backend #10
RF-06 - Propuesta de películas/series
- BrisaacFlix-backend #11
RF-07 - Aprobación de películas/series
- BrisaacFlix-backend #13
RF-08 - CRUD de reseñas

+ Add Item

Requerimientos Funcionales 10
AS-01 - Autenticación con JWT

- BrisaacFlix-backend #5
AS-01 - Autenticación con JWT
- BrisaacFlix-backend #6
AS-02 - Seguridad en almacenamiento de contraseñas
- BrisaacFlix-backend #7
AS-03 - Limitación de peticiones
- BrisaacFlix-backend #9
AS-04 - Validación de categorías duplicadas
- BrisaacFlix-backend #12
AS-05 - Validación de títulos repetidos
- BrisaacFlix-backend #16
AS-06 - Ranking ponderado
- BrisaacFlix-backend #20
AS-07 - Validación de entrada
- BrisaacFlix-backend #21
AS-08 - Transacciones críticas
- BrisaacFlix-backend #22

+ Add Item

Requerimientos No Funcionales
RNF-01 - Tiempo de respuesta

- BrisaacFlix-backend #25
RNF-01 - Tiempo de respuesta
- BrisaacFlix-backend #26
RNF-02 - Seguridad Contraseña
- BrisaacFlix-backend #27
RNF-03 - Expiración de Tokens
- BrisaacFlix-backend #28
RNF-04 - Express Rate Limit
- BrisaacFlix-backend #29
RNF-05 - Arquitectura Modular
- BrisaacFlix-backend #30
RNF-06 - Mensajes de error
- BrisaacFlix-backend #23
RNF-07 - Documentación con Swagger

+ Add Item

In Progress
This is actively being worked on

+ Add Item

Done
This has been completed

+ Add Item

jhonisaacMedinaMendoza08 / Projects / BrisaacFlix

BrisaacFlix

Backend Frontend + New view

Filter by keyword or by field

Discard Save

Todo 8
This item hasn't been started

- BrisaacFlix-backend #8
RF-05 - CRUD de categorías
- BrisaacFlix-backend #10
RF-06 - Propuesta de películas/series
- BrisaacFlix-backend #11
RF-07 - Aprobación de películas/series
- BrisaacFlix-backend #14
RF-09 - Estructura de reseñas
- BrisaacFlix-backend #15
RF-10 - Likes/dislikes en reseñas
- BrisaacFlix-backend #17
RF-11 - Listado por popularidad
- BrisaacFlix-backend #18
RF-12 - Filtrado por categoría
- BrisaacFlix-backend #19
RF-13 - Vista de detalle

+ Add Item

AS 7
AS-03 - Limitación de peticiones

- BrisaacFlix-backend #7
AS-03 - Limitación de peticiones
- BrisaacFlix-backend #9
AS-04 - Validación de categorías duplicadas
- BrisaacFlix-backend #12
AS-05 - Validación de títulos repetidos
- BrisaacFlix-backend #16
AS-06 - Ranking ponderado
- BrisaacFlix-backend #20
AS-07 - Validación de entrada
- BrisaacFlix-backend #21
AS-08 - Transacciones críticas
- BrisaacFlix-backend #24
AS-10 - Configuración de CORS

+ Add Item

Requerimientos No Funcionales
RNF-01 - Tiempo de respuesta

- BrisaacFlix-backend #25
RNF-01 - Tiempo de respuesta
- BrisaacFlix-backend #26
RNF-02 - Seguridad Contraseña
- BrisaacFlix-backend #28
RNF-04 - Express Rate Limit
- BrisaacFlix-backend #29
RNF-05 - Arquitectura Modular
- BrisaacFlix-backend #30
RNF-06 - Mensajes de error
- BrisaacFlix-backend #23
RNF-07 - Documentación con Swagger

+ Add Item

In Progress 2
This is actively being worked on

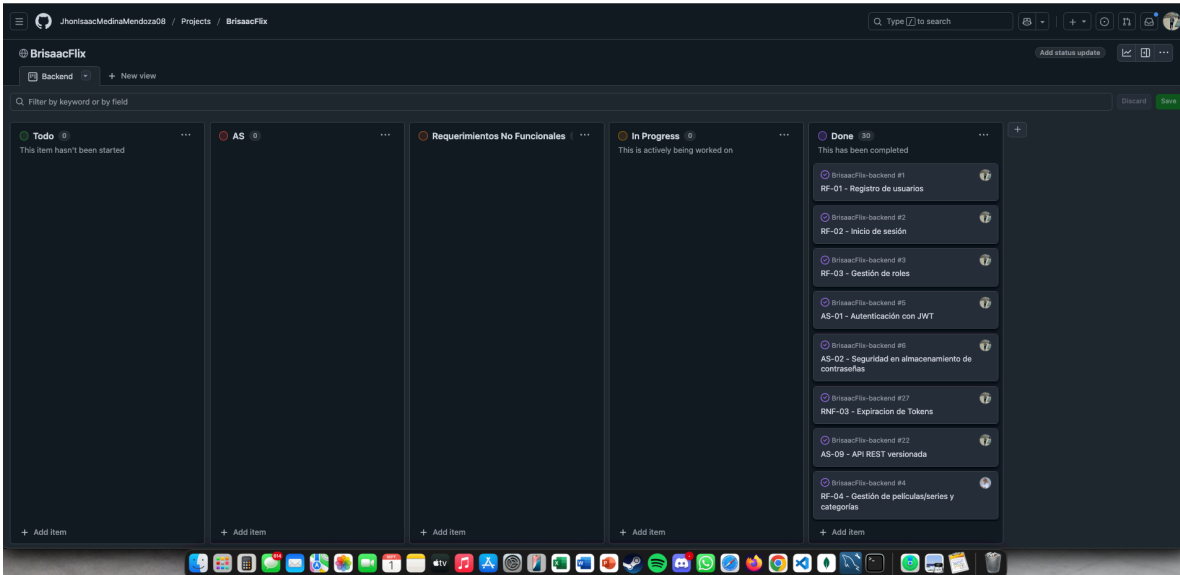
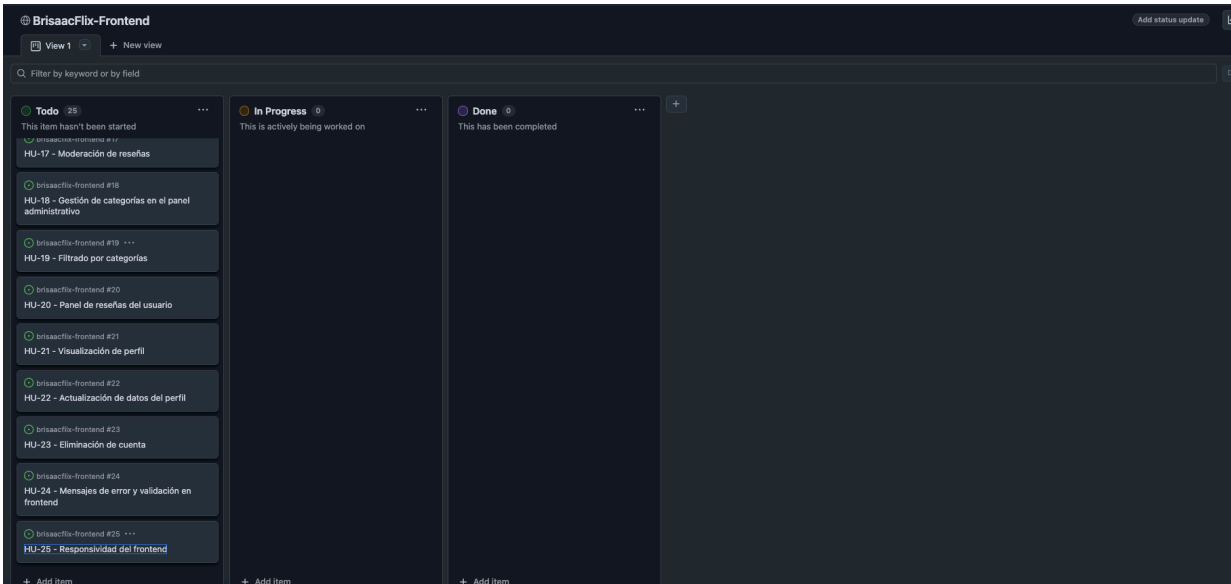
- BrisaacFlix-backend #4
RF-04 - Gestión de películas/series y categorías
- BrisaacFlix-backend #13
RF-08 - CRUD de reseñas

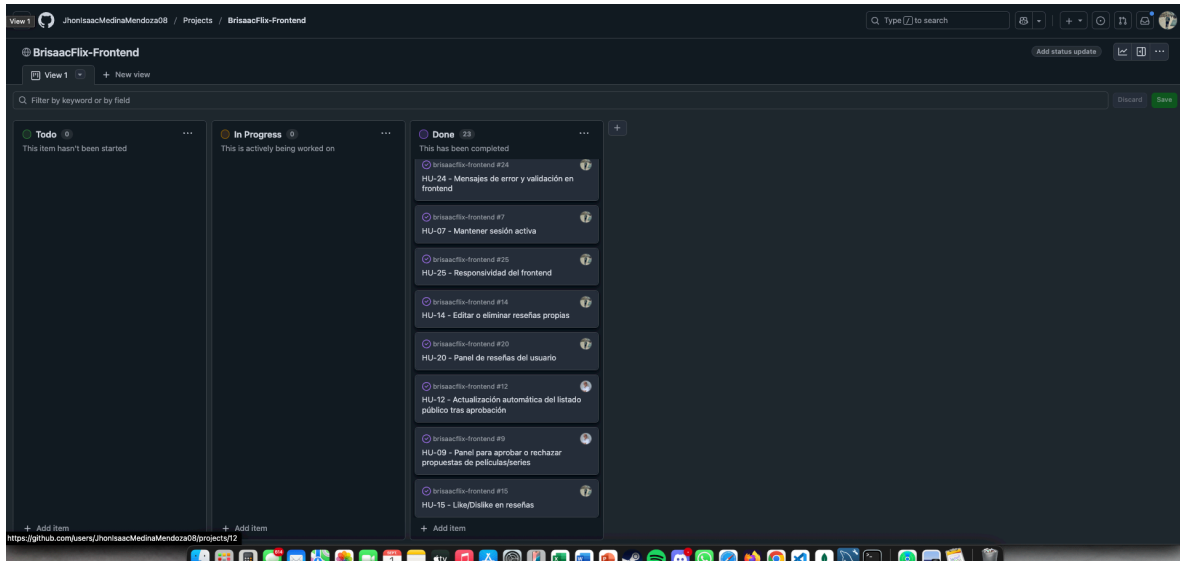
+ Add Item

Done 7
This has been completed

- BrisaacFlix-backend #1
RF-01 - Registro de usuarios
- BrisaacFlix-backend #2
RF-02 - Inicio de sesión
- BrisaacFlix-backend #3
RF-03 - Gestión de roles
- BrisaacFlix-backend #5
AS-01 - Autenticación con JWT
- BrisaacFlix-backend #6
AS-02 - Seguridad en almacenamiento de contraseñas
- BrisaacFlix-backend #27
RNF-03 - Expiración de Tokens
- BrisaacFlix-backend #22
AS-09 - API REST versionada

+ Add Item





- **Documentación del producto final:**
 - Tecnologías utilizadas: **Node.js, Express, MongoDB, JWT, REACT.**
 - Cumplimiento de las funcionalidades:
 - Registro e inicio de sesión de usuarios con autenticación JWT.
 - Gestión de películas, series, categorías y reseñas.
 - Diferenciación de roles (usuario y administrador).
 - Frontend responsive y conectado al backend.
 - Consumo de API externa para precarga de contenido.

7. CONCLUSIONES

Conclusiones generales del documento:

- La aplicación **BrisaacFlix** logró cumplir los objetivos planteados, ofreciendo una plataforma para explorar, calificar y reseñar películas/series.
- La implementación de **SCRUM** facilitó la organización del trabajo en equipo, garantizando entregas incrementales y retroalimentación constante.
- La separación entre backend y frontend permitió una arquitectura clara, escalable y alineada con buenas prácticas.
- Se aseguraron aspectos clave como seguridad (JWT, bcrypt), validaciones (express-validator), rendimiento (rate-limiter) y documentación (Swagger).

Conclusiones de la retrospectiva del Sprint:

- Se destacó la **buena comunicación** y compromiso del equipo como factor clave en el éxito del proyecto.
- Se identificó la necesidad de **mejorar la estimación de tiempos** para ciertas tareas, especialmente en la integración de frontend-backend.

- Se acordó reforzar la **automatización de pruebas** en futuros proyectos para garantizar mayor calidad en cada sprint.
- El uso de un **tablero SCRUM** y reuniones diarias contribuyó a la transparencia y a mantener el foco en los objetivos.