—— MEMORIA——

Proyecto DAM

CURSO 2O23/2O24

Jose Antonio Jiménez Villarroya

Contenido

Idea del proyecto

La aplicación permite realizar búsquedas específicas de películas, libros, videojuegos y música con los cuales el usuario puede interactuar para hacer reviews respecto al medio que haya consumido y determinar a qué otros medios le recuerda. También incluye una funcionalidad social, de forma que los usuarios pueden añadir comentarios sobre las reviews de otros usuarios y seguir a otros usuarios porque les han interesado sus reviews o les gustan sus recomendaciones, pudiendo ver información pública de estos usuarios y otras reviews que hayan hecho.

La aplicación implementada utiliza las APIs de Google Books (https://developers.google.com/books/docs/v1/using?hl=es-419), la de Spotify (https://developer.spotify.com/documentation/web-api), la de TMDB (https://developer.themoviedb.org/reference/intro/getting-started) y la de IGDB (https://api-docs.igdb.com/#get ng-started) para proporcionar a los usuarios acceso a información acerca de películas, libros, videojuegos y música de todo el mundo.

Respecto a las notificaciones, los usuarios pueden recibir notificaciones (en un feed de notificaciones) cuando un usuario les siga o responda a una de sus reviews.

Esta aplicación está destinada a usuarios que quieran compartir sus gustos sobre diferentes medios que han consumido con toda la gente posible así como que compartan con ellos esto mismo.

Modelo de Negocio

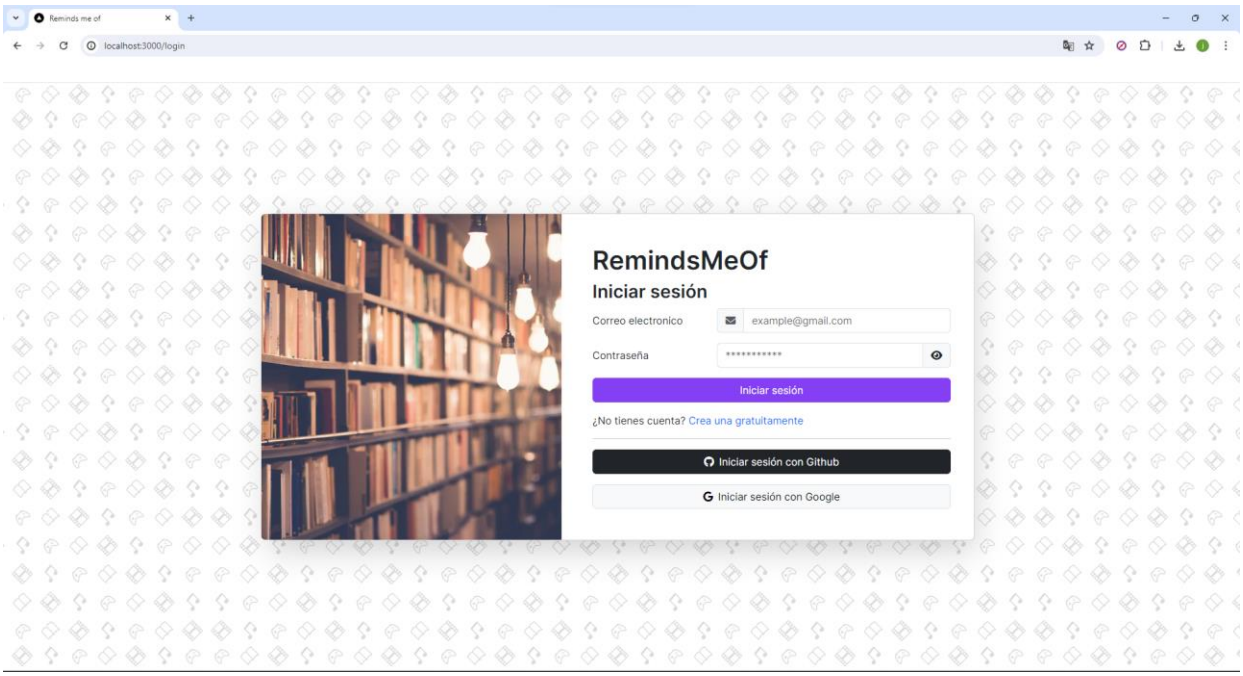
La aplicación ofrece una experiencia única al centralizar la información sobre películas, libros, videojuegos y música en una sola plataforma. Permite a los usuarios no solo buscar y descubrir nuevos contenidos, sino también interactuar y compartir sus experiencias con una comunidad.

El público serian amantes del cine, la literatura, los videojuegos y la música que buscan una plataforma centralizada para gestionar sus intereses y usuarios que disfrutan compartiendo y leyendo reviews y recomendaciones sobre sus medios de entretenimiento favoritos.

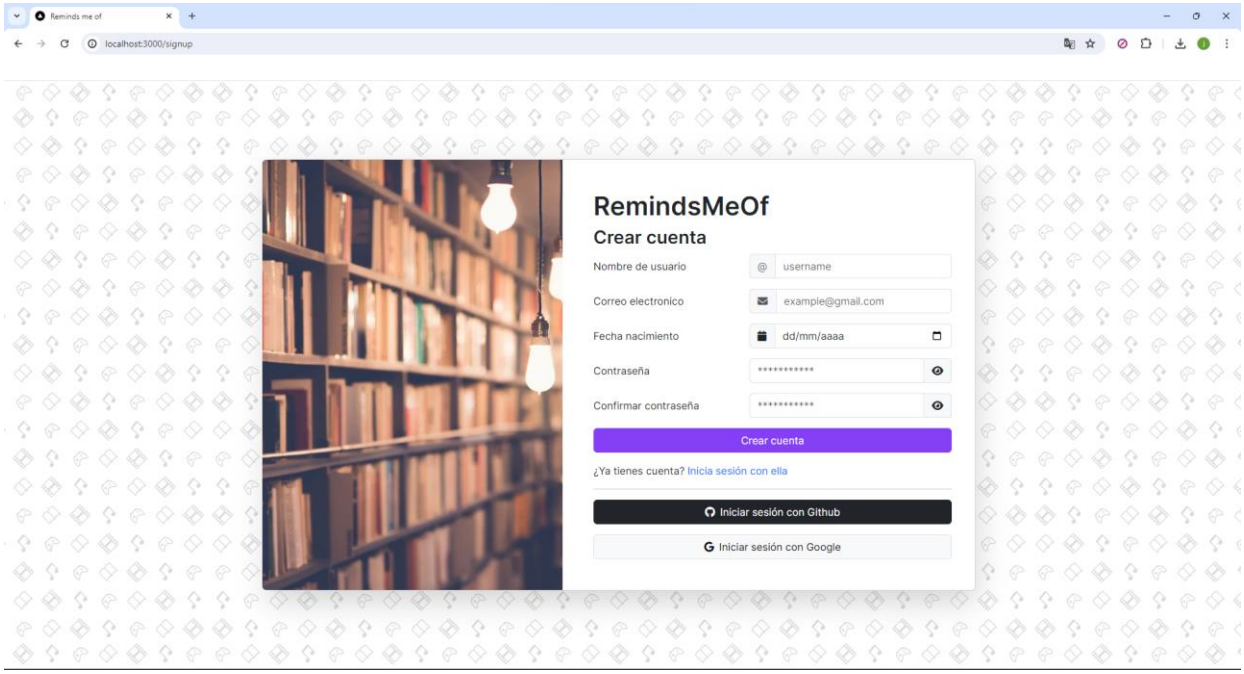
Como fuente de ingresos se puede ofrecer un modelo freemium: acceso gratuito con funcionalidades básicas y una suscripción premium para características avanzadas, publicidad dentro de la versión gratuita, comisiones por recomendaciones afiliadas a tiendas de libros, música, películas y videojuegos o alguna plataforma existente para gestionar su contenido.

Prototipado de pantallas

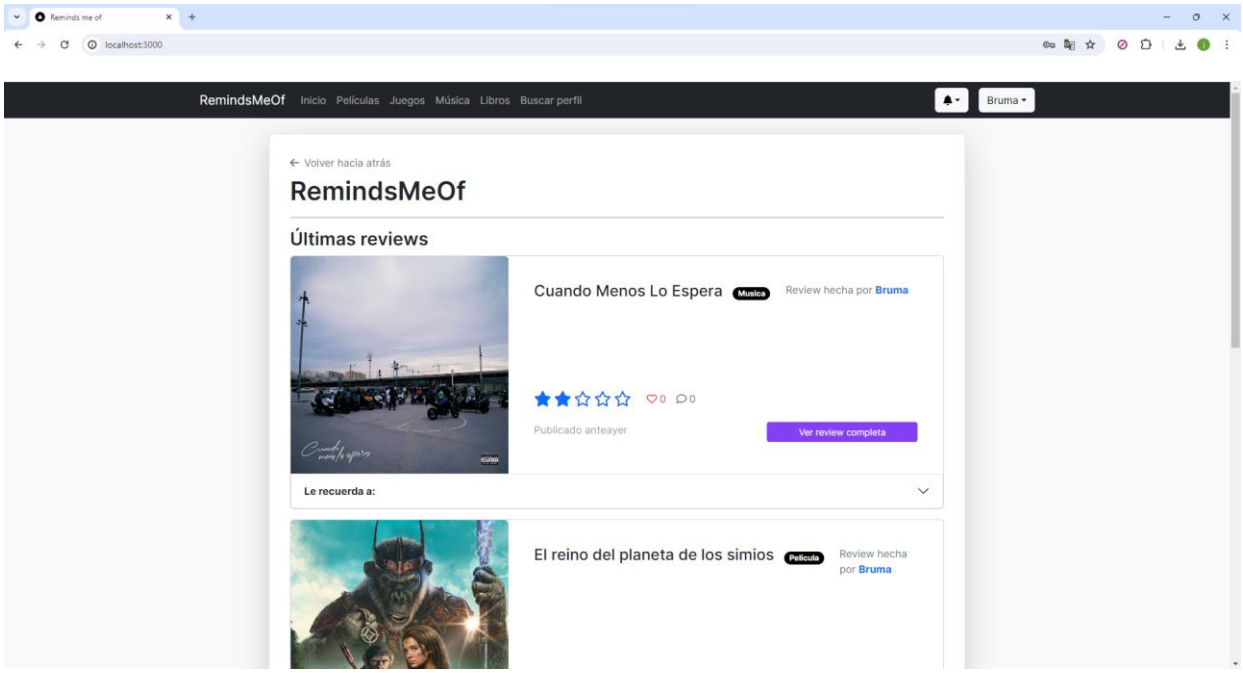
Login



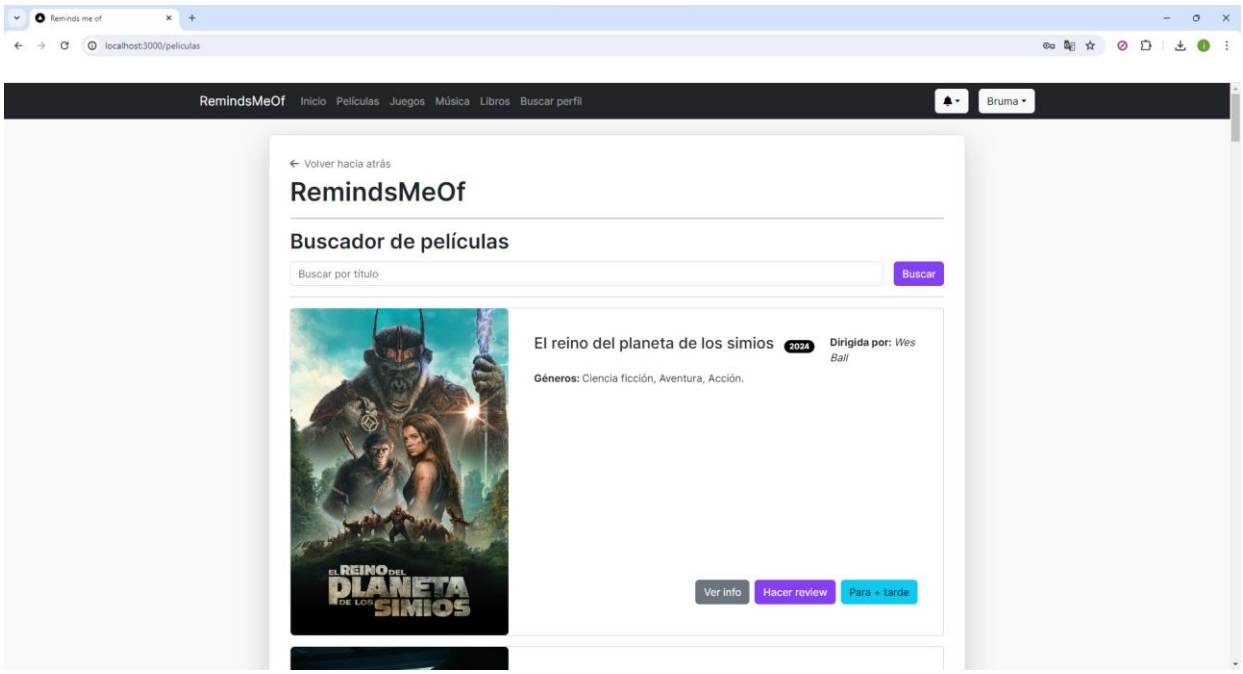
Registro



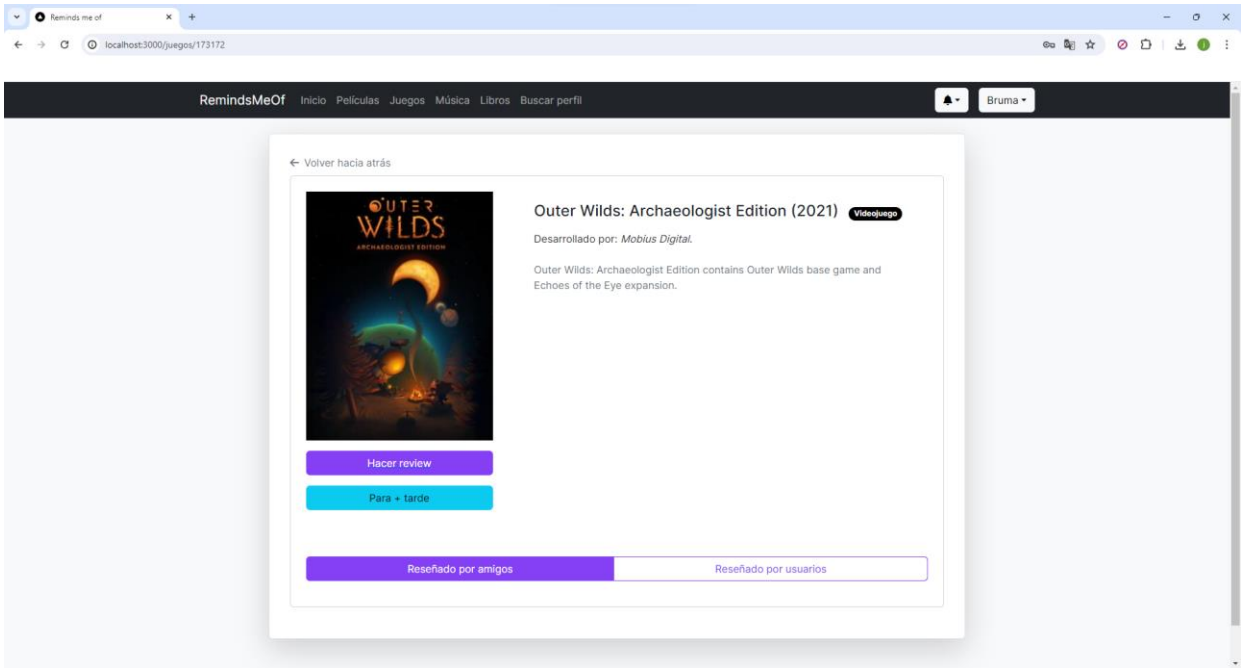
Pantalla de incio



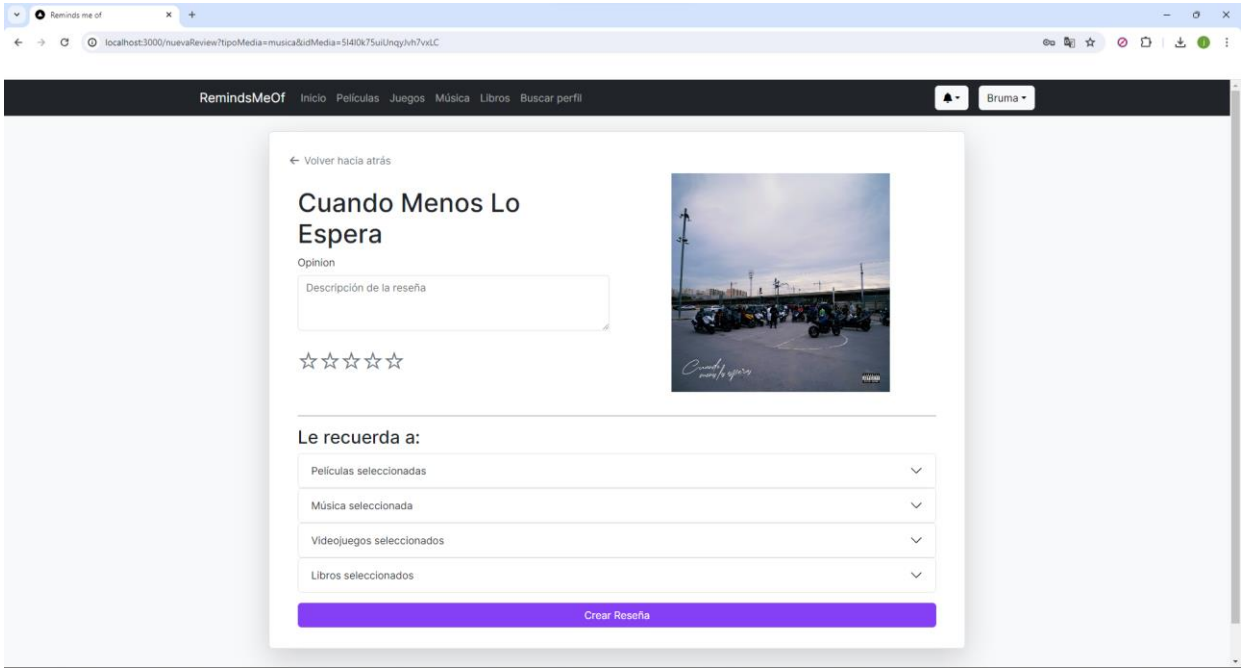
Pantalla peliculas



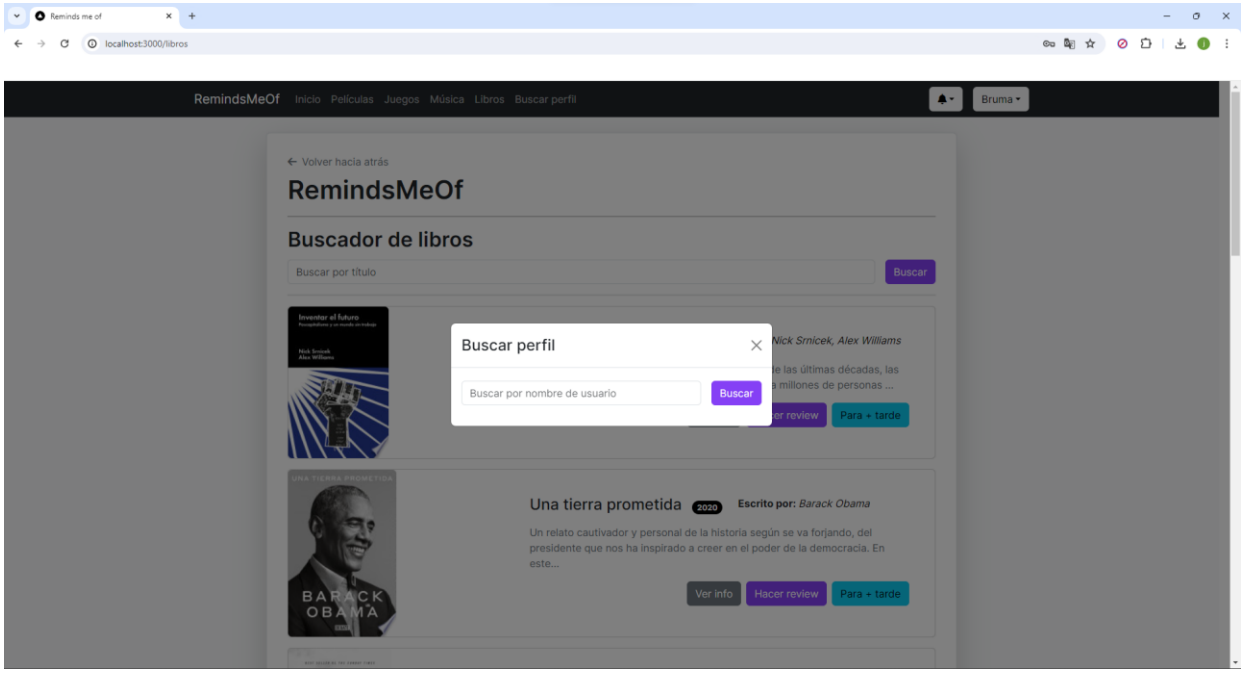
Pantalla botón ver info



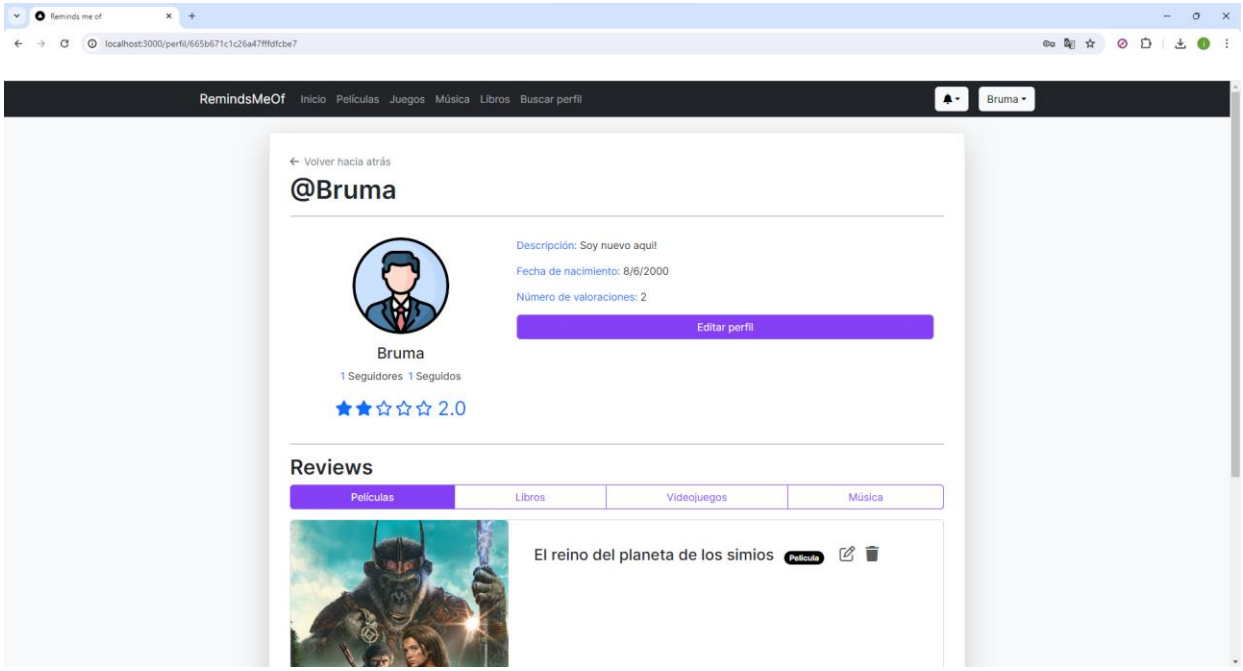
Pantalla hacer review



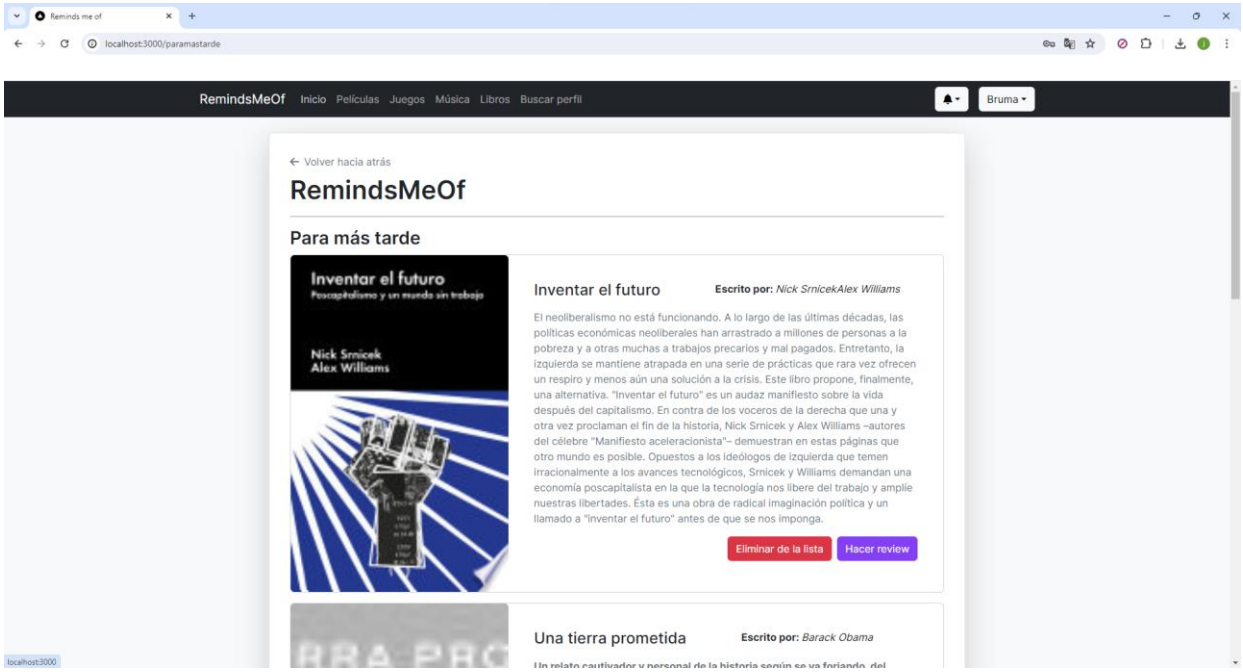
Buscador perfil



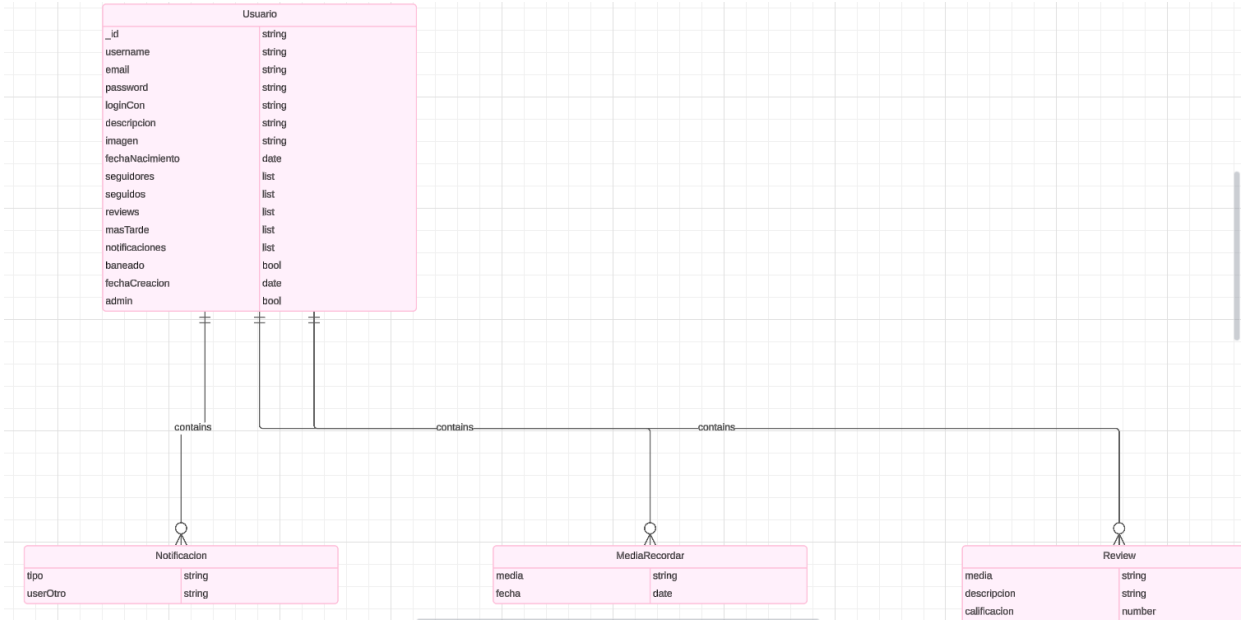
Pantalla perfil

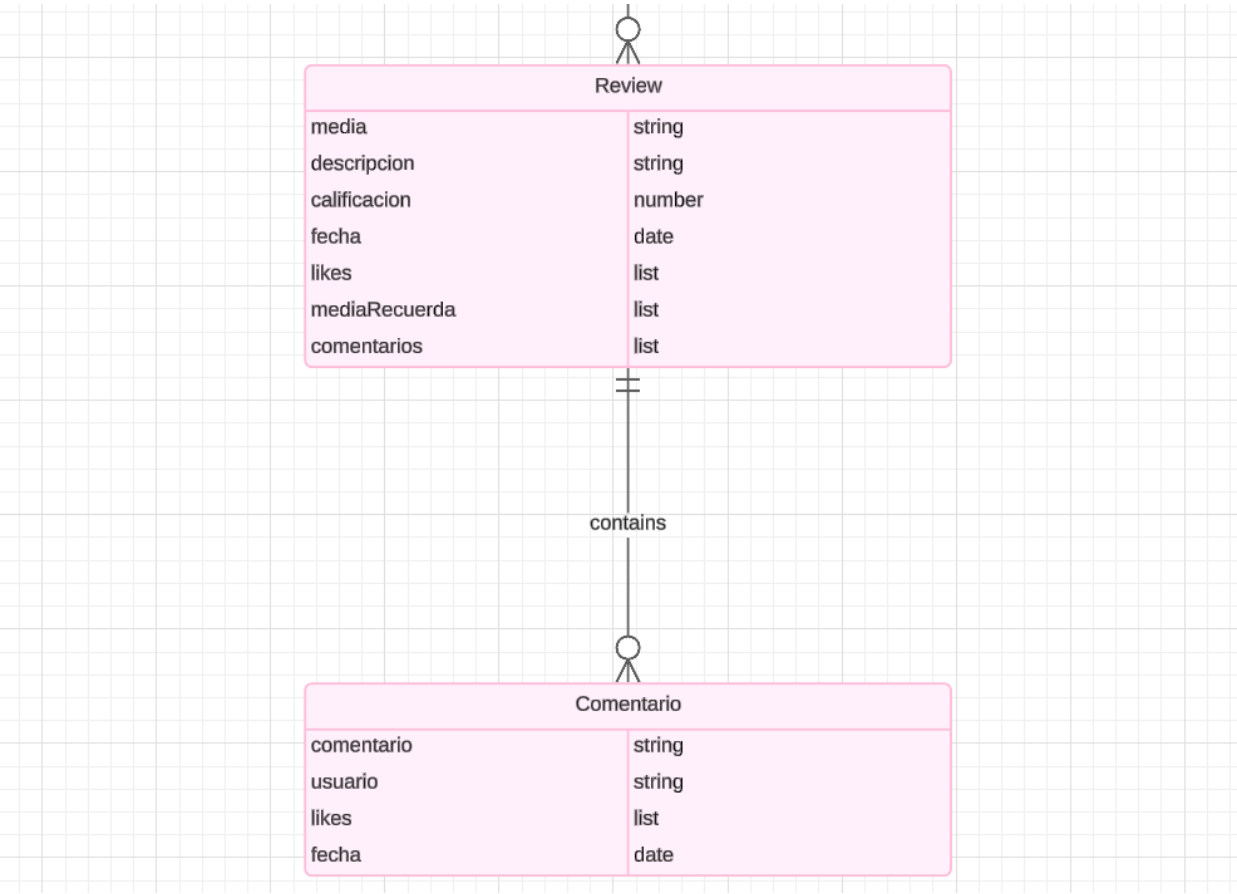


Pantalla Para mas tarde



Digrama Entidad Relación





Usuario: La entidad principal que contiene atributos básicos como username, email, password, etc., y referencias a las subentidades embebidas.
Review: Subentidad embebida dentro de Usuario, que contiene información sobre las reviews de medios.
Comentario: Subentidad embebida dentro de Review, que contiene comentarios sobre las reviews.
MediaRecordar: Subentidad embebida dentro de Usuario, que contiene información sobre medios que el usuario desea recordar para más tarde.
Notificacion: Subentidad embebida dentro de Usuario, que contiene notificaciones recibidas por el usuario.

Relaciones:

- Usuario tiene varias Reviews.
- Review tiene varios Comentarios.
- Usuario tiene varias MediaRecordar.
- Usuario tiene varias Notificaciones.

Arquitectura y stack tecnológico

Frontend

Se ha implementado con React y se han utilizado las siguientes bibliotecas y módulos.

- **React-bootstrap:** biblioteca que proporciona componentes de Bootstrap como componentes React. Permite crear interfaces de usuario utilizando los estilos de Bootstrap de manera más coherente y eficiente en aplicaciones React.
- **Next-Auth:** Permite solucionar la autenticación en aplicaciones Next.js. Proporciona hooks como "usesession" y "signOut" para la gestión del usuario.
- **@github/relative-time-element:** Biblioteca que permite mostrar automáticamente fechas en un formato relativo, como "hace 5 minutos" o "hace 2 días". Facilita mostrar los tiempos de manera más amigable al usuario.
- **html-react-parser:** Biblioteca que permite analizar cadenas HTML y convertirlas en elementos React. Utilizado para renderizar el contenido HTML dinámico de manera segura.
- **next/link:** Módulo de Next.js que proporciona un componente Link para navegar entre páginas de la web. Permite crear enlaces que habilitan la carga de páginas de manera rápida y eficiente.
- **easy-toast-react-bootstrap:** Biblioteca utilizada para mostrar notificaciones (toasts). Integra funcionalidad de toast con los estilos de Bootstrap, facilitando la implementación de mensajes temporales para hacer el feedback al usuario.
- **next/navigation:** Módulo utilizado para la navegación programática en aplicaciones Next.js. Permite redirigir a los usuarios a diferentes rutas dentro de la aplicación.
- **chart.js:** Biblioteca de JavaScript utilizada para crear gráficos y visualizar datos. Sobre la que se construye "react-chartjs-2", proporcionando una amplia variedad de tipos de gráficos y opciones de personalización.
- **react-chartjs-2:** biblioteca que proporciona una integración entre Chart.js y React, permitiendo crear gráficos y visualizar datos de manera sencilla utilizando componentes React. Soporta múltiples tipos de gráficos.
- crypto: Módulo de Node.js que proporciona funcionalidades de criptografia. En el contexto del frontend, se puede utilizar para la generación de hashes y cifrado de datos.
- **jsonwebtoken:** para crear y verificar tokens JSON Web (JWT), que son utilizados para la autenticación y autorización en aplicaciones web. Permite generar tokens firmados que pueden ser enviados al cliente y verificados en el servidor, garantizando la integridad y autenticidad de los datos transmitidos, lo cual es esencial para la implementación de mecanismos de seguridad en APIs y servicios web.

@fortawesome/react-fontawesome: Biblioteca que integra FontAwesome con React, permitiendo usuarios los iconos como componentes React

- **@fortawesome/free-solid-svg-icons:** Conjunto de iconos gratuitos de FontAwesome en formato SVG, utilizando “@fortawesome/react-fontawesome” para incluir los iconos en la aplicación React.
- **react-google-recaptcha:** Biblioteca que proporciona un componente React para Google reCAPTCHA, utilizado para prevenir el spam y el abuso en páginas web mediante la verificación de usuarios humanos.
- next/font/google: Módulo de Next.js que facilita la inclusión y optimización de fuentes de Google, mejorando la experiencia del usuario.
- **next/script:** Componente de Next.js para añadir y gestionar scripts. Permite controlar la carga y ejecución de scripts de manera óptima y segura.

Estas bibliotecas, módulos y componentes permiten crear un aplicación moderna y funcional, aprovechando las ventajas de React y Next.js para construir interfaces de usuarios robustas y eficientes.

Backend

Se ha desarrollado con Node.js y Express, y se han empleado las siguientes bibliotecas y módulos para garantizar un buen funcionamiento y una correcta implementación de la API y el servidor.

- Mongoose: biblioteca de modelado de objetos para MongoDB que se emplea en Node.js. Permite definir esquemas de datos y proporciona una interfaz para interactuar con MongoDB, simplificando la ejecución de operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en los métodos de la API.
- Swagger: Se han utilizado los módulos swagger-jsdoc y swagger-ui-express para documentar y visualizar la API con Swagger. Con swagger-jsdoc se ha escrito la documentación directamente en el código fuente usando comentarios de estilo jsdoc, mientras que swagger-ui-express se ha usado para generar el archivo de especificación Swagger y ponerlo disponible en una ruta específica de la aplicación Expres.
- Dotenv: biblioteca utilizada para cargar variables de entorno desde un archivo .env para no incluir claves de API, contraseñas o direcciones directamente en el código fuente.
- node-fetch: biblioteca utilizada para realizar solicitudes HTTP a las API como la de Google Books o a la de Spotify.
- **Passport:** biblioteca utilizada para la implementación del login social con Google y Github. Concretamente se han empleado los módulos passport-google-ouath20 y passport-github2.(Sin implementar)
- **Winston:** biblioteca utilizada para el registro de información (logging) durante la ejecución del backend.
- **jsonwebtoken:** para crear y verificar tokens JSON Web (JWT), que son utilizados para la autenticación y autorización en aplicaciones web. Permite generar tokens firmados que pueden ser enviados al cliente y verificados en el servidor, garantizando la integridad y autenticidad de los datos transmitidos, lo cual es esencial para la implementación de mecanismos de seguridad en APIs y servicios web.

Implementación

Frontend

La aplicación frontend se ha desarrollado utilizando React y Next.js, siguiendo una estructura modular para mantener el código claro y organizado. Esta estructura facilita el mantenimiento y la escalabilidad de la web. Módulos y componentes de la aplicación:

- Páginas (pages) : Las páginas son componentes React que representan las diferentes vistas de la aplicación. Next.js utiliza enrutamiento basado en sistemas de archivos, donde cada archivo de la carpeta se convierte en una ruta accesible de la aplicación.
- Componentes (components): La carpeta componentes incluye componentes React reutilizables que se utilizan en diferentes partes de la web. Estos componentes encapsulan su lógica y diseño que se podrá reutilizar en múltiples páginas. Ej: Navbar,js para la barra de navegación.
- Contextos (contexts): Proporcionan una forma de compartir valores como el estado de autentificación, la configuración del usuario sin necesidad de pasar explícitamente las propiedades a través de cada nivel.
- Hooks: Encapsulan la lógica reutilizable relacionada con el estado proporcionando una forma limpia y eficiente de gestionar el comportamiento del componente. Ej: useAuth para manejar la autenticación del usuario.

- Estilos: “global.css” se utiliza para definir estilos globales que se afectarán a toda la función.
- Utils: Incluye funciones auxiliares y utilidades que se utilizan en cualquier parte de la web. Ej: obtenerDatos.js sirve para solicitudes a las diferentes APIs (Spotify, The Movie Database, Google Books, IGDB) para obtener la información necesaria.

Backend

La aplicación creada con Node.js y Express utiliza el patrón MVC y cuenta con diversos módulos para mantener una estructura modular. Esto facilita la claridad del código y proporciona un mayor control sobre su implementación.

- controllers: se encarga de manejar la lógica de control de la aplicación.
- models: incluye la representación y gestión de los datos de la aplicación.
- routes: incluye la representación y gestión de los datos de la aplicación.

También se ha incluido el módulo utils donde se incluyen funciones auxiliares. Además, se incluye el módulo de middlewares, que contiene varios métodos para la autenticación de usuarios y el manejo de sesiones sin estado mediante tokens JWT. También se implementa el control de roles para asegurar que solo los administradores puedan realizar ciertas acciones. Estas funciones actúan como middleware, ejecutándose antes de llegar a los controladores finales o rutas específicas.

Trabajo futuro

La aplicación ha sido diseñada para proporcionar a los usuarios una plataforma centralizada donde puedan buscar y revisar películas, libros, videojuegos y música, así como interactuar socialmente mediante comentarios y seguimientos.

Aplicación móvil

Desarrollar versiones nativas de la aplicación para iOS y Android, asegurando una experiencia de usuario optimizada en dispositivos móviles.
Como beneficios: Aumenta la accesibilidad y conveniencia, permitiendo a los usuarios interactuar con la aplicación en cualquier momento y lugar.

Sistema de recompensas

Introducir un sistema de recompensas que incentive a los usuarios a interactuar más con la aplicación, como otorgar puntos por escribir reviews, comentar, y recibir "me gusta" en sus publicaciones.
Como beneficios: Aumenta el compromiso del usuario y promueve la participación activa en la comunidad.

Foros y grupos de interes

Crear foros y grupos de interés donde los usuarios puedan discutir sobre temas específicos relacionados con películas, libros, videojuegos y música.
Como beneficios: Fomenta la creación de subcomunidades dentro de la plataforma, aumentando el sentido de pertenencia y la interacción entre usuarios con intereses similares.