
This is the **published version** of the bachelor thesis:

Heinze Silva, Benaía Johannes. *Plataforma social de lectura con integración de clubes mediante Discord*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334392>

under the terms of the  license.

Plataforma social de lectura con integración de clubes mediante Discord

Benaía J. Heinze Silva

28 de junio de 2026

Resumen— Proyecto para el diseño, desarrollo e implementación de una aplicación web destinada a la centralización de actividades relacionadas con la lectura como actividad de ocio. Se promueve una experiencia más humana, centrada en la gestión de librerías personales, en la valoración de los libros leídos, en la participación en clubes de lectura a través de Discord y en la interacción social de sus usuarios a través de la recomendación personalizada de libros sin sesgos comerciales. Todo esto se consigue gracias a la búsqueda de libros mediante APIs externas. El sistema se ha desarrollado mediante una arquitectura cliente-servidor, incluyendo funcionalidades de verificación, personalización del perfil y gestión de caché para agilizar la carga de la aplicación. El resultado es una aplicación funcional que cumple con los criterios definidos al inicio del proyecto, donde se busca fomentar la cultura y la participación en la lectura a través de su comunidad.

Palabras clave— Aplicación, Club de lectura, Discord, Libros, Node.js, PostgreSQL, React, Recomendaciones, Red social, Valoraciones, Web

Abstract— Project for the design, development and implementation of a web application aimed at the centralization of activities related to reading as a leisure activity. A more human experience is promoted, focused on the management of personal libraries, the evaluation of books read, the evaluation of books read and the participation in reading clubs through Discord and the social interaction of its users through the personalized recommendation of books without commercial biases. All this is achieved thanks to the search for books through external APIs. The system has been developed through a client-server architecture, including verification functionalities, profile customization and cache management to streamline the application's loading. The result is a functional application that meets the criteria defined at the beginning of the project, where it seeks to promote culture and the participation in reading through its community.

Keywords— App, Book club, Books, Discord, Node.js, PostgreSQL, Ratings, React, Recommendations, Social network, Web

1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

EN los últimos años, el consumo de literatura ha aumentado entre los jóvenes en Europa [1] y España [2], mientras que otras investigaciones muestran una tendencia a la baja en regiones como Estados Unidos [3]. Asimismo, la literatura ha experimentado cambios sustanciales debido a la digitalización y el auge de las redes

sociales, incluidas aquellas orientadas específicamente a la lectura. Sin embargo, muchas de estas aplicaciones y plataformas, que tienen como público objetivo a las personas que leen por ocio, presentan algún tipo de limitación, como la necesidad de interactuar entre lectores.

Actualmente, las comunidades en línea establecidas en Instagram, X (anteriormente Twitter), Tiktok u otras plataformas se han convertido en los principales entornos de interacción social en línea. Sin embargo, las dinámicas de lectura y recomendaciones dentro de este espacio suelen depender de mensajes dispersos y carecen de un sistema estructurado que permita atender todas o varias necesidades de estas comunidades.

Ante este contexto, el presente proyecto propone el diseño e implementación de una plataforma social centrada de forma exclusiva en la lectura que permita a los usuarios

• E-mail de contacto: heinzesilvabeni@gmail.com
 • Mención: Ingeniería del Software
 • Trabajo tutorizado por: Yolanda Benítez Fernández (Departamento de ciencias de la computación)
 • Curso 2025/26

registrar sus libros, compartir sus reseñas, descubrir nuevas lecturas mediante un sistema de recomendaciones y, lo más importante, crear y gestionar clubes de lectura integrados con Discord.

El objetivo no es únicamente desarrollar una aplicación funcional, sino fomentar el descubrimiento y la participación comunitaria en la lectura en entornos sociales ya existentes.

2 ESTADO DEL ARTE

Hoy en día, ya existen diversas plataformas orientadas al descubrimiento y valoración de libros:

- **Goodreads** [4] es la plataforma más consolidada en el ámbito social de la lectura. Permite a los usuarios registrar libros, realizar reseñas y revisar las de otros usuarios, así como conseguir recomendaciones, teóricamente basadas en sus preferencias [5]. Sin embargo, pertenece a Amazon, lo que resulta en un sistema de recomendaciones manipulado debido a su dominación sobre el mercado de la lectura digital. Por otra parte, la aplicación no presenta innovaciones e ignora las peticiones de sus usuarios [6]. De igual manera, Amazon se utiliza como el centro principal para revisar opiniones de libros. También contiene grupos para comentar libros mediante mensajes de texto, pero carece de usuarios y se trata de la funcionalidad menos utilizada de la aplicación.
- **StoryGraph** es una solución parecida a Goodreads y, además, pone a disposición del usuario estadísticas y recomendaciones más detalladas y personalizadas. Sin embargo, su integración con otras aplicaciones es prácticamente inexistente, a pesar de que ofrecen una API para que sea posible. Cabe recalcar que fue creada en 2019 y en los últimos años ha habido un crecimiento en los usuarios, alcanzando los 5 millones de usuarios registrados a principios de 2026 [7], pero sigue sin competir con los más de 150 millones de usuarios de Goodreads.
- **Bookum** está más orientada a clubes, pero no ha acabado de despegar y no tiene suficiente base de usuarios como para poder encontrar lo que uno busca.
- **Fable** también incorpora grupos de lectura y, a diferencia de Goodreads, pone a disposición de los usuarios una selección más extensa, pero se centra únicamente en la comunicación mediante texto. Han tenido problemas de discriminación con el uso de Inteligencia Artificial que ha afectado a su imagen. Aparte, solo está disponible como aplicación móvil, por lo que no es compatible con otros dispositivos [8].
- **LibraryThing** [9] es una de las plataformas más veteranas, abierta desde 2005 y con cerca de tres millones de usuarios. Su mayor fortaleza es la catalogación: ofrece un control sobre los metadatos de cada libro muy superior al del resto de aplicaciones, con acceso a la *Library of Congress* y a miles de bibliotecas. Sin embargo, su interfaz resulta poco intuitiva y compleja para los nuevos usuarios, su experiencia móvil está

poco cuidada y la versión gratuita incluye publicidad, además de limitar el número de libros que se pueden catalogar.

- **Discord** y otras plataformas de comunicación por voz se han convertido en el nuevo espacio de encuentro para numerosos clubes de lectura, que crean canales específicos para comentar sus lecturas. Sin embargo, no son funcionalidades nativas y, por lo tanto, son soluciones creadas por los propios usuarios. De todas formas, la migración de usuarios a estas plataformas evidencia la necesidad de crear un programa que integre estas funciones de base. Además, tiene el problema de no poder comentar libros en un idioma diferente al del país en el que se realizan.

Como se puede observar, ninguna de las soluciones contempla un sistema de recomendación propio ni la integración de algún chat de voz/vídeo para las gestiones de clubes de lectura.

3 OBJETIVOS

El objetivo principal de este proyecto ha consistido en desarrollar una plataforma social de lectura que centralice la gestión de libros, las reseñas, las recomendaciones personalizadas y los clubes de lectura integrados con Discord. A partir de esta idea, se han definido los objetivos del proyecto y sus prioridades:

TABLA 1. PRIORIDAD DE LOS OBJETIVOS

ID	Objetivo	Prioridad
O1	Desarrollar una plataforma social de lectura que centralice las actividades de lectura (compartir reseñas, recibir recomendaciones, gestionar clubes de lectura...)	Crítica
O2	Integrar la plataforma con Discord para dar soporte a clubes de lectura.	Crítica
O3	Recomendar las obras de forma personalizada y fomentar la diversidad de géneros.	Crítica
O4	Crear un espacio de interacción entre usuarios para fomentar la actividad de lectura.	Alta
O5	Garantizar el acceso a la plataforma desde diferentes dispositivos y su sincronización entre dispositivos.	Alta
O6	Diseñar una interfaz clara, accesible y centrada en la experiencia de usuario.	Media
O7	Garantizar la seguridad y privacidad de todas las partes involucradas.	Media

4 METODOLOGÍA

Se ha escogido la metodología *Waterfall* [10] como la más adecuada.

Dado que el proyecto es individual y se requiere de la entrega sistemática de informes y comunicación con la tutora, lo más adecuado ha sido una aproximación estructurada de forma secuencial en fases diferenciadas, ya que era necesario tener bien definida, desarrollada y acabada la fase anterior para poder pasar a la siguiente. En casos muy excepcionales, y solo cuando era completamente necesario, se haría una revisión de las fases anteriores, siempre y cuando afectara al resto del proyecto.

Esta metodología lineal ha facilitado tanto establecer una planificación clara desde el comienzo como el seguimiento del progreso respecto al plan inicial.

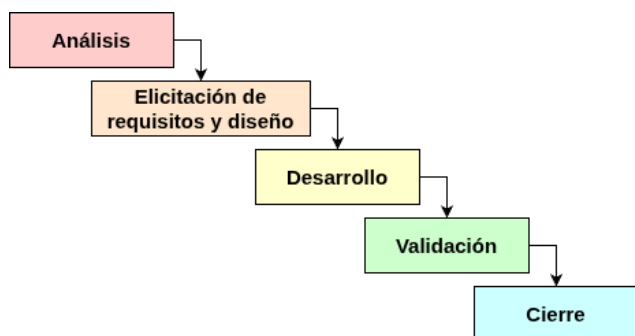


Fig. 1: Diagrama Waterfall

5 PLANIFICACIÓN

El proyecto se ha dividido en cinco fases, cada una enfocada en tareas específicas que se tienen que realizar para poder conseguir los objetivos de cada entrega. Para pasar a la siguiente fase, será necesario haber acabado la anterior, como se ha mencionado en el apartado de la metodología. Cada fase tiene asignado un periodo temporal, indicado junto a su nombre en la siguiente tabla:

TABLA 2. PLANIFICACIÓN DEL PROYECTO

Fase	Tareas e hitos
Fase 1: Análisis 09/02/2026 – 06/03/2026	
1	- Definición del alcance y objetivos. - Estudio de plataformas existentes. - Organización del proyecto. - Elaboración del informe inicial. <i>Hito:</i> Reunión inicial. <i>Hito:</i> Entregable informe inicial.
Fase 2: Requisitos y diseño 07/03/2026 – 09/04/2026	
2	- Elicitación de requisitos funcionales y no funcionales. - Caso de uso. <i>Hito:</i> Reunión sobre requisitos.
3	- Diseño de la arquitectura del sistema. - Modelo de datos y diseño de la base de datos. - Prototipo. - Elaboración del informe de progreso I. <i>Hito:</i> Entregable informe de progreso I.

Fase	Tareas e hitos
Fase 3: Desarrollo 11/04/2026 – 22/05/2026	
4	- Registro e inicio de sesión. - Registro de libros. - Reseñas de libros. - Interacción con las reseñas.
5	- Sistema de recomendaciones. - Integración con Discord y bot. <i>Hito:</i> Reunión sobre desarrollo.
6	- Desarrollo frontend. - Integración backend-frontend. - Elaboración del informe de progreso II. <i>Hito:</i> Entregable informe de progreso II.
Fase 4: Validación 23/05/2026 – 07/06/2026	
7	- Validación de requisitos. - Pruebas de usuario. - Validación del sistema de recomendaciones. - Test de integración con Discord. <i>Hito:</i> Sistema validado.
8	- Elaboración propuesta informe final. <i>Hito:</i> Reunión sobre informe final. <i>Hito:</i> Entregable propuesta informe final.
Fase 5: Cierre 08/06/2026 – 28/06/2026	
9	- Organización del dossier digital. - Preparación de presentación. <i>Hito:</i> Reunión final. <i>Hito:</i> Entregable final.

Esta planificación es el resultado de haber sufrido algunas desviaciones a lo largo del desarrollo.

6 REQUISITOS Y DISEÑO

En la segunda fase del proyecto se ha realizado la elicitación de requisitos [11]. Para ello se ha utilizado la metodología MoSCoW [12] con el objetivo de identificar y priorizar la información más relevante para el proyecto. Este proceso ha permitido determinar sus necesidades y establecer las bases para el diseño de la aplicación (Véase tabla 3 y 4).

6.1. Casos de uso

Después de identificar todos los requisitos de la aplicación, se ha elaborado un diagrama de casos de uso para poder visualizar cuales eran los papeles que debían asumir los diferentes usuarios del sistema y que funcionalidades eran necesarias para ello. El diagrama establece las bases del diseño técnico de la aplicación y define con precisión las funcionalidades críticas que deberían poder realizar los distintos usuarios del sistema. Se han considerado funcionalidades prácticas, como el registro e inicio de sesión, o la búsqueda de libros, y funcionalidades sociales, como la creación de clubes de lectura o el seguimiento entre usuarios. En los anexos se incluyen los diagramas de casos de uso, separados en uso general de la aplicación (véase fig. 10) y en clubes de lectura con Discord (véase fig. 11).

6.2. Tecnologías y arquitectura

Se han considerado distintas tecnologías tanto para el *front-end* y el *back-end* como para la base de datos.

Dado que el objetivo es el desarrollo de una *webapp*, la primera opción era utilizar una solución tradicional basada en HTML, CSS y JavaScript. Sin embargo, en este proyecto las funcionalidades especificadas en los requisitos han supuesto un aumento en su complejidad.

También era posible plantear un *back-end* con tecnologías como PHP o una aplicación más monolítica con menor separación entre cliente y servidor. No obstante, se requería de la capacidad de integrarse con el servicio externo de Discord y sus bots. Esto último es de vital importancia para el proyecto, ya que para Node.js existe una librería específica (*discord.js* [13]) para interactuar con la API de Discord.

En cuanto a persistencia de datos, se podían considerar tanto bases de datos relacionales como no relacionales. De todos modos, el problema está formado por entidades muy interconectadas, por lo que para este tipo de estructuras es más adecuado utilizar una base de datos relacional.

Por otro lado, se ha decidido utilizar React para el desarrollo del *front-end* ya que se adecúa más a aplicaciones web centradas en la experiencia de usuario, gracias a que permite construir la interfaz mediante componentes reutilizables [14], facilitando así su mantenimiento.

Para la lógica del servidor, se ha escogido Node.js [15], ya que se adapta bien a aplicaciones web con múltiples operaciones concurrentes. A su vez, y uno de los motivos más importantes, la coherencia con el *front-end* es muy alta: al utilizar JavaScript tanto en cliente como en servidor, se reduce la complejidad del proyecto y se simplifica su desarrollo.

Por último, se ha seleccionado PostgreSQL como base de datos ya que se trata de una base de datos relacional robusta y es perfecta para la estructura del sistema. La aplicación maneja relaciones claras entre entidades y con una baja complejidad. La combinación React + Node.js + PostgreSQL se considera la más adecuada, no solo por la existencia de múltiples proyectos que ya usan esta misma combinación de tecnologías [16], sino también por la coherencia con las necesidades del proyecto.

Por lo tanto, la arquitectura propuesta para el sistema se basa en un modelo cliente-servidor de tres capas, compuesto por una capa de presentación, una capa de lógica y una capa de persistencia, que permite separar claramente la lógica general del sistema.

En la capa de presentación se sitúa el *front-end* desarrollado con React, encargado de la interacción con el usuario y de la representación de la información en la interfaz. El objetivo es ofrecer una experiencia intuitiva y accesible, adaptada a diferentes resoluciones.

La capa de lógica se implementa mediante un *back-end* en Node.js, cuya responsabilidad es procesar las peticiones entre cliente y servidor. A su vez, contiene todas las funcionalidades de lógica, ya sea validación de credenciales, gestión de usuarios, etc.

La capa de persistencia se apoya en PostgreSQL ya que el sistema debe garantizar consistencia de datos, sincronización entre dispositivos y una gestión segura de los datos personales.

De forma paralela, la integración con Discord se ha plan-

teado como un subsistema externo conectado al *back-end*. Esto permite mantener la plataforma web como núcleo principal del sistema y tratar Discord como una integración especializada.

El diagrama de arquitectura se encuentra en los anexos (véase fig. 15)

6.3. Modelo de datos y de base de datos

Para poder hacer una estructura coherente en la aplicación, y que todo se relacionara de forma óptima, se han definido las entidades que conforman la aplicación, así como los datos de cada entidad. Para ello primero se ha realizado un modelo de datos UML conceptual para así poder definir el modelo de la base de datos.

6.3.1. Modelo UML

El diseño de la base de datos se ha desarrollado a partir de un modelo conceptual representado en un diagrama de clases UML, considerando las necesidades de la aplicación así como las funcionalidades definidas en la fase de análisis. Entonces, se han creado distintas clases para organizar y preservar la información de forma coherente.

6.3.2. Modelo entidad-relación

Una vez definido el modelo mediante el diagrama de clases, se ha utilizado para crear el modelo relacional. Este proceso implica transformar las clases en tablas, sus atributos en columnas y las relaciones en claves foráneas. En los anexos se incluye el diagrama partido en tres para su correcta visualización (véase figs. 12, 13 y 14).

Cabe mencionar que este modelo evolucionó respecto al diseño inicial, ya que no se adaptaba a las necesidades de todas las funcionalidades, ya sea por la falta de clases entre relaciones o la falta de atributos para ciertos aspectos del programa.

6.4. Prototipo

Se ha desarrollado un prototipo de la aplicación usando Figma en base a los requisitos recogidos, el diagrama de casos de uso y todo el análisis del proyecto. Esta herramienta permite previsualizar la estructura y el flujo de navegación de las diferentes pantallas antes de empezar el desarrollo.

El prototipo está pensado para mostrar el funcionamiento básico de la plataforma, incluyendo sus pantallas y funciones básicas. A su vez, permite diseñar las interfaces y ver cómo quedaría el resultado final. Por ejemplo, podemos ver a continuación la pantalla de bienvenida y la de clubes de lectura:

7 DESARROLLO

En la fase de desarrollo se han utilizado los requisitos definidos y el diseño para implementar la aplicación y establecer las bases sobre las que opera la plataforma.

7.1. Organización de la aplicación

El entorno de desarrollo se ha configurado en un único repositorio que contiene tanto el código del cliente como el

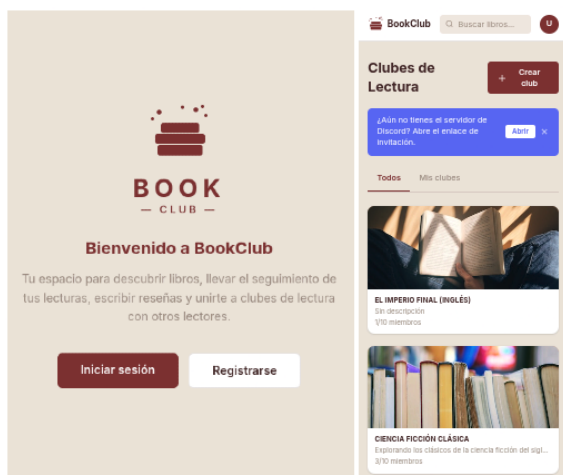


Fig. 2: Páginas prototipo

del servidor, lo que facilita la gestión de dependencias compartidas [17] y permite ejecutar ambas capas con un único comando.

El proyecto se organiza en dos carpetas principales:

- **client:** aplicación web construida con React y Vite, con rutas gestionadas mediante React Router [18].
 - **pages:** las dieciocho pantallas de la aplicación (biblioteca personal, perfil, clubes de lectura, recomendaciones, estadísticas, etc.).
 - **components:** componentes reutilizables (barra de navegación, tarjetas de libros, sistema de valoración por estrellas...).
 - **context:** gestión global de la sesión de usuario mediante Context API.
 - **utils:** funciones auxiliares, como el cálculo del progreso de lectura.
- **server:** servidor de API construido con Express.js sobre Node.js.
 - **routes:** once módulos de endpoints REST (autenticación, libros, biblioteca, reseñas, clubes, listas, recomendaciones, usuarios, géneros, favoritos).
 - **middleware:** autenticación basada en JWT para proteger los endpoints privados.
 - **db:** conexión a PostgreSQL y esquemas SQL de inicialización.
 - **discord:** integración con el bot de Discord (`discord.js`).
 - **utils:** funciones auxiliares del servidor, principalmente normalización de géneros literarios.

Esta organización ha facilitado la implementación y la validación progresiva de las funcionalidades, asegurando una integración fluida de los diversos componentes del sistema.

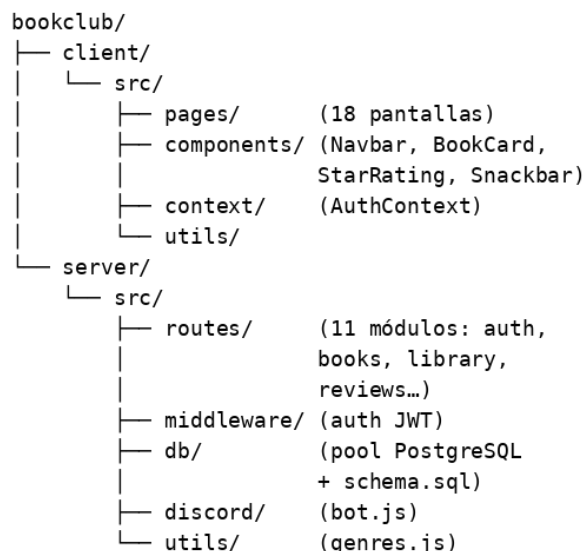


Fig. 3: Árbol ASCII de la estructura de ficheros

7.2. Integración con el catálogo de libros

Para evitar tener que construir y mantener un catálogo propio de libros, se ha optado por integrar la plataforma con una plataforma externa para poder recoger los datos necesarios.

La primera opción fue hacer uso de OpenLibrary, ya que era la opción *open-source* más completa y no requería autenticación. Al realizar una búsqueda, el servidor consulta su API, devuelve los resultados al cliente y los guarda automáticamente en la base de datos para tener un catálogo de libros activos de forma local.

No obstante, presentaba varios problemas. Al tratarse de un catálogo colaborativo donde puede contribuir cualquier usuario, los datos no siempre son fiables y es habitual encontrar entradas incompletas o poco normalizadas, lo que dificulta su almacenamiento y la gestión del catálogo. A esto se le suma que el contenido está mayoritariamente en inglés, que algunos títulos no aparecían al buscarse y que la API tiene una limitación en la carga de datos, por lo que los tiempos de respuesta eran demasiado lentos en cada consulta.

Por estos motivos, se decidió migrar a Dilve [19], la base de datos bibliográfica oficial de la Federación de Gremios de Editores de España. Al ser un servicio oficial alimentado por las propias editoriales, sus datos son más fiables y están mucho más normalizados, lo que facilita almacenarlos en la base de datos y mantener el catálogo actualizado con las últimas novedades.

La desventaja es que solo dispone de libros en español, por lo que el catálogo queda limitado a este mercado. Además, requiere autenticación para acceder al servicio, que se debe solicitar directamente al servicio.

7.3. Almacenamiento de imágenes

Para la gestión de las imágenes subidas por los usuarios, como las fotos de perfil o las imágenes de los clubes de lectura, se ha integrado Cloudinary, un servicio externo de almacenamiento y entrega de imágenes. Su uso se ha limi-

tado únicamente a este tipo de contenido debido a su coste, mientras que el resto de imágenes, como las portadas de los libros, se obtienen directamente de la fuente.

7.4. Sistema de Recomendaciones

El motor de recomendaciones genera hasta veinte sugerencias personalizadas para cada usuario combinando múltiples fuentes de datos. Con el fin de minimizar el tiempo de respuesta, todas las consultas independientes se ejecutan en paralelo:

- **Autores favoritos:** Busca en la base de datos libros de autores de los que el usuario ha leído tres o más títulos.
- **Géneros preferidos:** Busca títulos de los géneros que el usuario ha configurado como preferidos en su perfil.
- **Géneros valorados positivamente:** Busca títulos de géneros donde el usuario ha dado puntuaciones iguales o superiores a cuatro estrellas.
- **Filtrado colaborativo:** Identifica usuarios con un historial de lectura similar y sugiere libros que estos han leído pero el usuario actual no.
- **Co-lectores:** Recomienda títulos frecuentemente leídos por los mismos usuarios que han leído los mismos libros que el usuario actual.
- **Populares en la plataforma:** Incorpora los libros mejor valorados dentro de los géneros preferidos del usuario como complemento.

Los resultados se puntúan, se agrupan por género y se diversifican por autor, de manera que la lista final no quede dominada por un solo género ni un solo escritor. Los elementos descartados por el usuario se excluyen automáticamente durante treinta días.

Para mejorar el rendimiento en usos sucesivos, el servidor mantiene las recomendaciones en memoria con una vida útil de cinco minutos por usuario. La memoria caché se invalida automáticamente cuando el usuario descarta alguna recomendación, garantizando que los resultados reflejen siempre las preferencias actuales.

7.5. Integración con Discord

La plataforma se integra con Discord para enriquecer la experiencia de los clubes de lectura. El componente principal:

- **Bot de Discord:** Implementado con la biblioteca `discord.js` [20], el bot gestiona automáticamente el servidor de Discord asociado a la plataforma. Cuando se crea un club de lectura, el bot puede crear los canales de texto y de voz correspondientes en el servidor de Discord, permitiendo a los miembros conversar fuera de la plataforma web. El bot se inicia al arrancar el servidor y se ejecuta de manera concurrente con la API.

7.6. Verificación por Correo Electrónico

Para implementar la verificación de identidad mediante correo electrónico, se ha integrado el servicio Resend [21], que permite el envío de mensajes transaccionales a través de su API. El servidor utiliza el paquete oficial `Resend` junto con `Nodemailer` [22] para construir y enviar los mensajes HTML.

El flujo de verificación funciona de la manera siguiente: en el momento del registro, el servidor genera un *token* UUID único, lo guarda con una caducidad de un día y envía un correo a la dirección facilitada con un enlace de confirmación. El usuario no puede iniciar sesión hasta que no haya confirmado el correo.

El sistema de restablecimiento de contraseña sigue el mismo patrón: el usuario solicita el restablecimiento, el servidor comprueba la existencia del correo en la base de datos, genera un *token* de recuperación con caducidad de una hora y envía un correo con el enlace correspondiente. Al acceder al enlace, el servidor valida el *token*, actualiza la contraseña e invalida el *token* inmediatamente.

7.7. Clubes de lectura e integración con Discord

Los clubes de lectura son, probablemente, la funcionalidad que mejor define el proyecto. Permiten crear un club o unirse a uno existente, hacer un seguimiento del progreso de cada miembro, organizar discusiones y fijar objetivos comunes, así como invitar a otros usuarios. Para acceder a ellos hay que tener la cuenta vinculada con Discord y unirse al servidor de la aplicación, donde viven los canales de texto y voz asociados a cada club.

Cuando se crea un club nuevo, un bot implementado con `discord.js` se encarga de generar automáticamente el canal correspondiente en el servidor y de enlazarlo con el club de la plataforma. El mismo bot publica notificaciones cuando cambia el libro activo y permite consultar el progreso de los miembros directamente desde Discord, sin necesidad de entrar en la aplicación.

7.8. Despliegue

Para el despliegue en local se optó por Cloudflare Tunnel [23], pero para la versión final se ha optado por Railway [24] tanto para el *back-end* como para la base de datos, de esta manera la página web queda desplegada y los usuarios pueden acceder cuando lo deseen.

8 RESULTADOS

Tras el desarrollo del proyecto, se ha conseguido obtener como resultado una aplicación funcional que implementa la mayoría de los requisitos definidos. Se ha cumplido con su objetivo principal, que pretendía desarrollar una aplicación de lectura centralizada. Todo esto se ha conseguido mediante la integración de diferentes tecnologías, la gestión de datos, la interacción con el usuario y la aplicación de técnicas y patrones de desarrollo que se han aprendido durante el grado. Desde la funcionalidad más básica como lo es el registro e inicio de sesión por parte del usuario hasta la integración de la aplicación con un bot de Discord persona-

lizado, se ha logrado su implementación e integración con los demás sistemas.

La primera funcionalidad creada, dentro de la aplicación, ha sido la del sistema de gestión de usuarios, que incluye su registro, inicio de sesión y gestión propia por el usuario. Durante el registro se solicitan los datos básicos del usuario como el nombre de usuario, correo electrónico, contraseña, etc.

Aparte, permite seleccionar los géneros literarios preferidos del usuario, que ayudará a ofrecerle recomendaciones personalizadas. Una vez registrado, el usuario recibe un correo de verificación que ha de confirmar para poder usar la aplicación.

A partir de los géneros seleccionados por el usuario, junto con su historial de lectura, valoraciones y actividad de otros usuarios, la aplicación es capaz de ofrecer recomendaciones personalizadas que se ajustan a los gustos personales del lector. También puede descartar las recomendaciones que no le interesan para poder ajustar con mayor precisión sus recomendaciones.

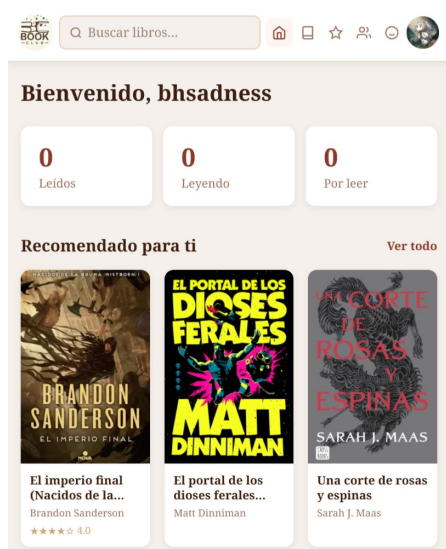


Fig. 4: Pantalla de inicio

Adicionalmente, se ha implementado un sistema de autenticación con Discord, que permite al usuario vincular su cuenta de la aplicación con la de Discord, de manera que pueda interactuar con todo el apartado de clubes de lectura que se explicará más adelante. Esto se consigue a través del flujo OAuth2 de Discord.

Otra de las funcionalidades más importantes del proyecto es el catálogo de libros. Se ha conseguido implementar mediante la integración con la API externa de Dilve, que permite obtener un catálogo amplio de libros. Cada libro muestra su propia ficha personalizada con información como el título, autor, sinopsis, etc.



Fig. 5: Pantalla de búsqueda

Dentro de la ficha de cada libro, el usuario puede dejar su propia reseña junto con una valoración numérica en forma de estrellas, además de poder leer las reseñas de otros usuarios, votar su utilidad y comentar sobre ellas para generar un hilo de discusión sobre el libro, lo que fomenta la interacción dentro de la comunidad.

Aparte, cada usuario dispone de una biblioteca personal en la que puede organizar los libros que está leyendo, los que ya ha leído y los que quiere leer en un futuro. También permite la creación de bibliotecas personalizadas. Esto permite mantener un seguimiento de su actividad como lector.



Fig. 6: Pantalla de biblioteca personal

También se ha implementado un sistema social que permite al usuario seguir a otros usuarios y ver en el *feed* la actividad reciente de estos, como las reseñas que han realizado o los libros que han añadido a alguna de sus librerías, lo que fomenta la interacción en la comunidad lectora.



Fig. 7: Pantalla de actividad

Cada usuario dispone de una sección de estadísticas personales en las que puede consultar datos sobre su propia actividad, como el número de libros leídos, los géneros leídos más frecuentes o la evolución de su actividad a lo largo del tiempo.



Fig. 8: Pantalla de estadísticas

Por último, la funcionalidad más característica de la aplicación es la de los clubes de lectura, que permiten al usuario crear o unirse a un club. Dentro de cada uno, se puede realizar un seguimiento del progreso de lectura de cada miembro, organizar discusiones y objetivos de lectura, así como invitar a otros usuarios a participar. Para poder acceder a esta funcionalidad es necesario vincular la cuenta de la aplicación con la de Discord, y unirse al servidor dedicado de la aplicación, ya que se utilizan los canales de texto/voz para poder hacer las discusiones pertinentes al club.



Fig. 9: Pantalla de creación de un club de lectura

En resumen, los resultados obtenidos cumplen con lo especificado durante las fases iniciales del proyecto y dan lugar a una aplicación funcional y coherente con la visión original que se tenía sobre él. No solo eso, sino que se puede considerar una base sobre la que seguir desarrollando esta misma idea. En el anexo se encuentran un par más de pantallas (véase fig. 16).

9 VALIDACIÓN

Con el objetivo de comprobar que la aplicación cumplía con los requisitos definidos y que ofrecía la experiencia de uso deseado, se ha llevado a cabo un proceso de validación basado en pruebas de usabilidad con usuarios reales.

9.1. Diseño de las pruebas

La validación se ha realizado mediante un cuestionario estructurado, con una participación por parte de 13 usuarios (7 desde ordenador y 6 desde dispositivo móvil), a los que se les proporcionó acceso a la aplicación desplegada. De esta manera, cada usuario pudo interactuar libremente con la plataforma en un entorno real, sin supervisión directa, replicando las condiciones de uso habituales.

El cuestionario se ha organizado en torno a los requisitos del proyecto. Para cada requisito funcional (RF) y no funcional (RNF), el usuario debía indicar uno de los siguientes estados: *funciona correctamente*, *funciona con problemas*, *no funciona* o *no probado*. Esta clasificación ha permitido validar individualmente cada funcionalidad y, a la vez, identificar qué partes no llegaron a probarse. Adicionalmente, se incluyeron varias preguntas abiertas sobre las dificultades encontradas, los aspectos menos claros y las posibles mejoras, así como una valoración global de la aplicación.

9.2. Resultados de las pruebas

La valoración media de la aplicación fue de 8,9 sobre 10, y el consenso general indica que las funcionalidades básicas (registro, inicio de sesión, gestión de la biblioteca, búsqueda de libros, reseñas y recomendaciones) funcionan correctamente y que la aplicación resulta, en su mayoría, intuitiva.

No obstante, las pruebas también permitieron detectar varios aspectos a mejorar:

- **Catálogo de libros:** los usuarios probaron la implementación con la API externa de Open Library, por lo que algunos títulos no aparecían en las búsquedas y el catálogo se encontraba mayoritariamente en inglés, mezclando idiomas en la interfaz. Se resolvió mediante el uso de los datos de Dilve, por lo que se tiene acceso a cualquier libro publicado en España, pero no a libros publicados en otros países o idiomas.
- **Rendimiento:** varios usuarios señalaron que las operaciones que dependen de consultas externas, especialmente la búsqueda, presentan tiempos de carga elevados.
- **Edición de perfil:** algunos usuarios indicaron que no se guardaban los cambios o que no localizaban con facilidad el acceso a su perfil.
- **Adaptación a móvil:** algunas pantallas no se ajustan correctamente en dispositivos móviles y obligan a desplazarse horizontalmente para ver todas las opciones.
- **Clubes de lectura e integración con Discord:** ha sido el apartado más problemático y, a la vez, el menos probado. Es la parte con mayor complejidad y se espera que el usuario ya esté familiarizado con Discord, ya que no depende del diseño de la aplicación. Los que lo habían usado anteriormente tuvieron menos dificultades para hacer uso de esta característica.

Finalmente, para evaluar el rendimiento bajo carga, se han realizado estas pruebas con cinco usuarios concurrentes, durante las cuales la aplicación mantuvo su funcionamiento sin caídas. El único problema ha seguido siendo la carga de datos que dependen de consultas externas, pero eso ha sido un problema independientemente de los usuarios concurrentes.

En resumen, la validación confirma que las funcionalidades cumplen con los requisitos establecidos y que la propuesta ha sido bien recibida por los usuarios, mientras que la gestión de clubes con Discord ha sido la funcionalidad que ha presentado más problemas, junto con las limitaciones del catálogo.

10 LÍNEAS ABIERTAS

Pese a haber cumplido con los objetivos definidos, durante el desarrollo y la validación del proyecto han surgido diversas líneas de mejora que quedan pendientes como trabajo futuro.

En primer lugar, la optimización del rendimiento de las búsquedas de libros es la línea más prioritaria. Como se ha mencionado, la limitación en la carga de datos de la API de Dilve obliga a utilizar una solución parcial basada en el uso

de la caché por el lado del cliente y la paralelización de las consultas. En un futuro se podría implementar la precarga de datos en memoria y el almacenamiento de las portadas en un servicio propio, soluciones descartadas inicialmente por su elevado coste, pero que mejorarían notablemente los tiempos de carga.

En segundo lugar, sería conveniente ampliar el catálogo de libros. Al depender de Dilve, el catálogo queda limitado a libros publicados en España, por lo que no incluye obras de otros países o idiomas. Para solucionarlo, se podrían integrar fuentes de datos adicionales que complementen el catálogo, ofreciendo así una cobertura más amplia.

En tercer lugar, la experiencia en dispositivos móviles también admite mejoras. Durante la validación se detectó que algunas pantallas no se ajustan correctamente y obligan a desplazarse horizontalmente, por lo que se podría revisar el diseño *responsive* para garantizar una correcta adaptación a todas las resoluciones.

En cuarto lugar, hay un margen de mejora bastante amplio en el apartado visual de la aplicación, y podría ser mucho más atractivo para el usuario.

Por último, el apartado de clubes de lectura y su integración con Discord ha sido el más complejo de utilizar para los usuarios. Por ello, se ha de revisar este apartado y facilitar su uso por parte del usuario. También, se puede realizar la implementación contraria, y es que todo el apartado de clubes de lectura se gestione directamente mediante Discord, evitando así tener que hacer uso de la plataforma de forma directa.

11 CONCLUSIONES

El desarrollo de este proyecto de final de grado me ha permitido crear una aplicación completamente funcional centrada en la centralización e interacción de la comunidad lectora, implementando todas las funcionalidades necesarias para ello utilizando tecnologías reales utilizadas en entornos profesionales.

A nivel técnico ha sido un proyecto con varias complicaciones, pero que me ha dado la oportunidad de crear una aplicación utilizando todos los conocimientos que he adquirido durante mi formación como ingeniero informático. La integración con las diferentes tecnologías me ha obligado a tomar decisiones de arquitectura constantemente, utilizando lo aprendido para encontrarles una solución y así poder resolver problemas con los cuales no me había encontrado anteriormente.

También, pese a mantener una planificación estructurada desde el inicio, su cumplimiento ha sido uno de los aspectos más difíciles del proyecto, ya sea por haber otorgado más tiempo y recursos de los que realmente se necesitaban, como por infravalorar la cantidad de recursos necesarios. No obstante, gracias a la base sólida que otorgaba, se ha conseguido obtener un proyecto alineado con los objetivos definidos desde el inicio.

A nivel personal, la integración con diferentes servicios como la API de Dilve, el servicio SMTP de correos o el bot de Discord ha supuesto un reto añadido, pero que me ha permitido aprender sobre nuevas tecnologías y adaptarme a ellas. No solo eso, sino que, gracias a la experiencia adquirida por ser el primer proyecto individual de estas características, va a facilitar el desarrollo del siguiente.

Como conclusión final, el proyecto ha sido una experiencia formativa muy importante y completa, con unos resultados muy positivos y un alto valor personal sobre mi carrera, tanto personal como profesional.

USO DE LA IA GENERATIVA

A lo largo del proyecto, se ha utilizado la IA para mejorar aspectos formales de los informes (como la visualización de tablas), validar los diferentes diagramas en base a las definiciones propias del proyecto, como herramienta de depuración (*debugging*) para encontrar errores en el código, para crear el logotipo de la aplicación y para afinar el apartado visual en base al diseño propio.

AGRADECIMIENTOS

Primero de todo, quiero darle las gracias a mi tutora, Yolanda Benítez, por toda la ayuda a lo largo del proyecto, que ha sido lo que ha conseguido que haya podido obtener estos resultados.

También agradecer a todas aquellas personas que han participado en alguna fase de la aplicación, obteniendo así el resultado final. En especial a Anna, por también corregir todos mis errores gramaticales.

REFERENCIAS

- [1] Eurostat, “Younger People and Women in the EU Read More Books,” European Commission, <https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20240809-2>, 2024.
- [2] Ministerio de Cultura y Deporte, “Barómetro de Lectura 2025,” <https://www.cultura.gob.es/actualidad/2026/01/260122-barometro-lectura-2025.html>, 2026.
- [3] D. Batty, “Children Reading Less, New Research Shows,” *The Guardian*, <https://www.theguardian.com/education/2020/feb/29/children-readinglessays-new-research>, 2020.
- [4] Goodreads, “Who Is the Average Goodreads User? You’ll Be Surprised,” https://www.goodreads.com/author_blog_posts/14538341-who-is-the-average-goodreads-user-you-ll-be-surprised, 2017.
- [5] Goodreads, “Goodreads Members’ Top 72 Hit Books of the Year So Far,” <https://www.goodreads.com/blog/show/2302-goodreads-members-top-72-hit-books-of-the-year-so-far>, 2022.
- [6] L. Norberg, “Redesign of Goodreads,” *Medium*, <https://medium.com/design-bootcamp/goodreads-first-redesign-in-20-years-what-changed-and-what-didnt-t-23589497264c>, 2025.
- [7] TheStoryGraph, “X Post Sobre Plataforma de Lectura,” X, <https://x.com/thestorygraph/status/2016996250466344994>, 2026.
- [8] “Is Anyone Aware of Any Online Book Club?,” *Reddit*, https://www.reddit.com/r/books/comments/14if5oo/is_anyone_aware_of_any_online_book_club/, 2023.
- [9] LibraryThing, “About LibraryThing,” <https://www.librarything.com/about>
- [10] Eye on Tech, “What Is the Waterfall Model and How Does It Work?,” *YouTube*, <https://youtu.be/bNLcRdrSQUA>, 2022.
- [11] A. Satpathy, “How to Write Good Software Requirements (with Examples),” *Modern Requirements*, <https://www.modernrequirements.com/blogs/good-software-requirements/>, 2025.
- [12] ProductPlan, “What Is MoSCoW Prioritization? Overview of the MoSCoW Method,” <https://www.productplan.com/glossary/moscow-prioritization/>, 2025.
- [13] discord.js, “Introduction,” <https://discordjs.guide/>, 2026.
- [14] UXPin, “React vs HTML – Can You Spot a Difference?,” *Studio by UXPin*, <https://www.uxpin.com/studio/blog/react-vs-html/>, 2023.
- [15] Node.js, “Introduction to Node.js,” <https://nodejs.org/learn/getting-started/introduction-to-nodejs>, 2026.
- [16] Novice7691, “Fastest Tech Stack for Web App Development in Your Experience,” *Reddit*, https://www.reddit.com/r/ExperiencedDevs/comments/1c2fb7b/fastest_tech_stack_for_web_app_development_in/, 2025.
- [17] npm, “npm Docs,” <https://docs.npmjs.com/>, 2026.
- [18] React Router, “Official Documentation,” <https://reactrouter.com/>, 2026.
- [19] Dilve, “¿Qué es DILVE?,” <https://web.dilve.es/dilve/que-es-dilve/>, 2026.
- [20] Discord, “Discord Developer Platform Documentation,” <https://docs.discord.com/developers/intro>, 2026.
- [21] Resend, “Introduction,” <https://resend.com/docs>, 2026.
- [22] Nodemailer, “Nodemailer Documentation,” <https://nodemailer.com/>, 2026.
- [23] Cloudflare, “Cloudflare Tunnel,” *Cloudflare Docs*, <https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/>, 2026.
- [24] Railway, “Railway Docs,” <https://docs.railway.com/>, 2026.

ANEXOS

A.1. Diagramas de casos de uso

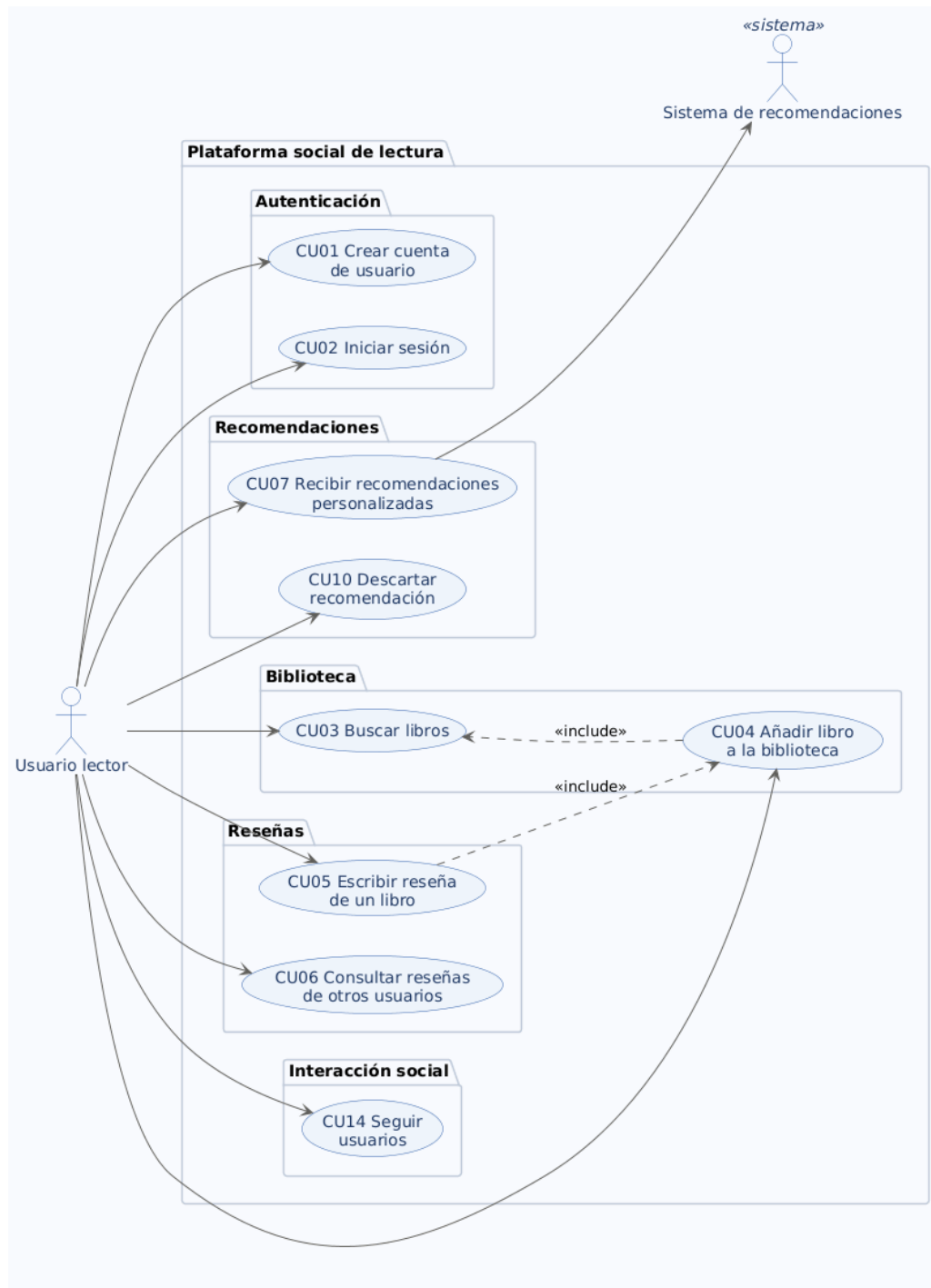


Fig. 10: Diagrama de casos de uso: uso general

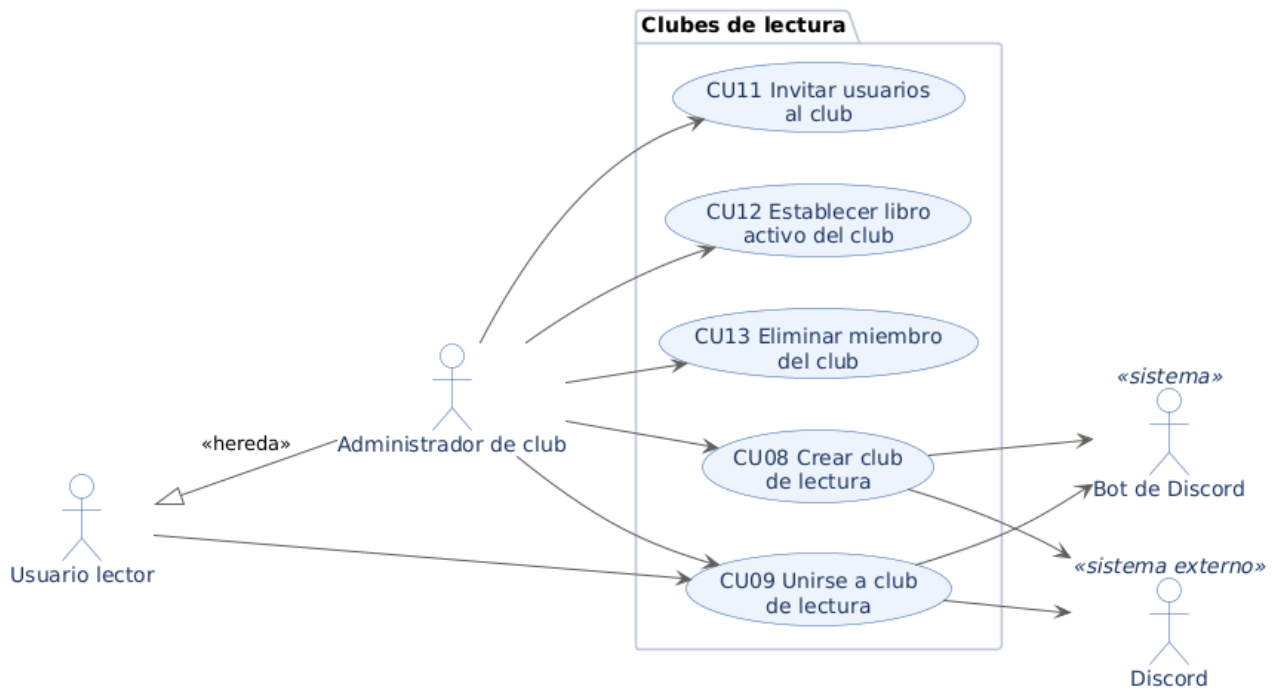


Fig. 11: Diagrama de casos de uso: clubes de lectura

A.2. Diagramas Entidad-Relación

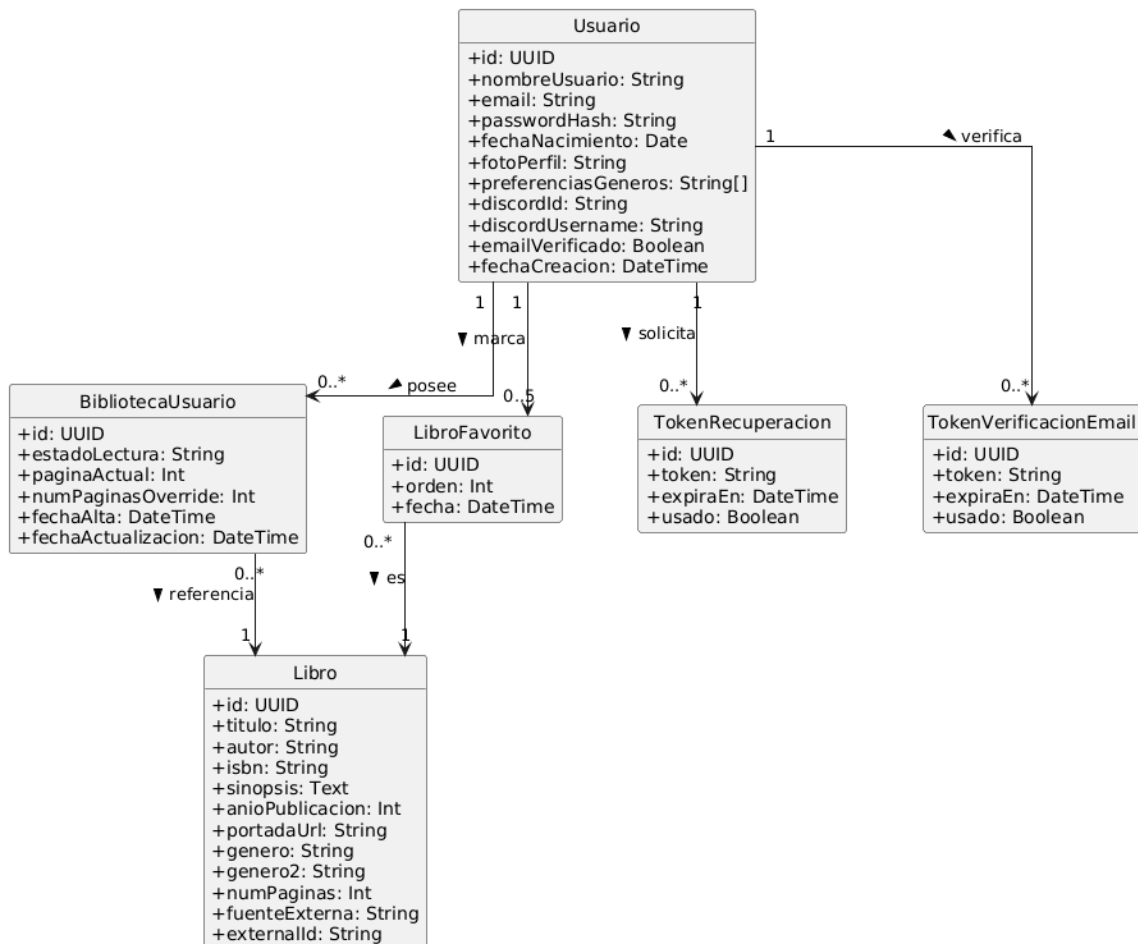


Fig. 12: Diagrama Entidad-Relación: uso general

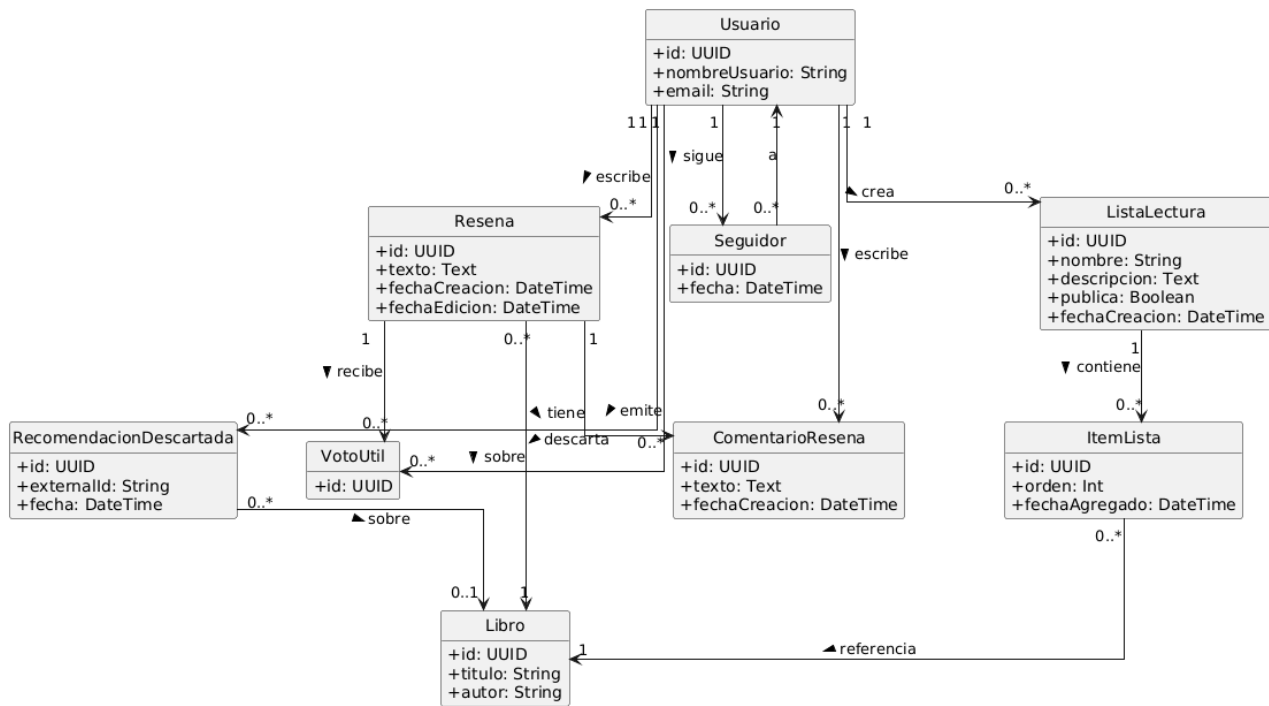


Fig. 13: Diagrama Entidad-Relación: interacción social

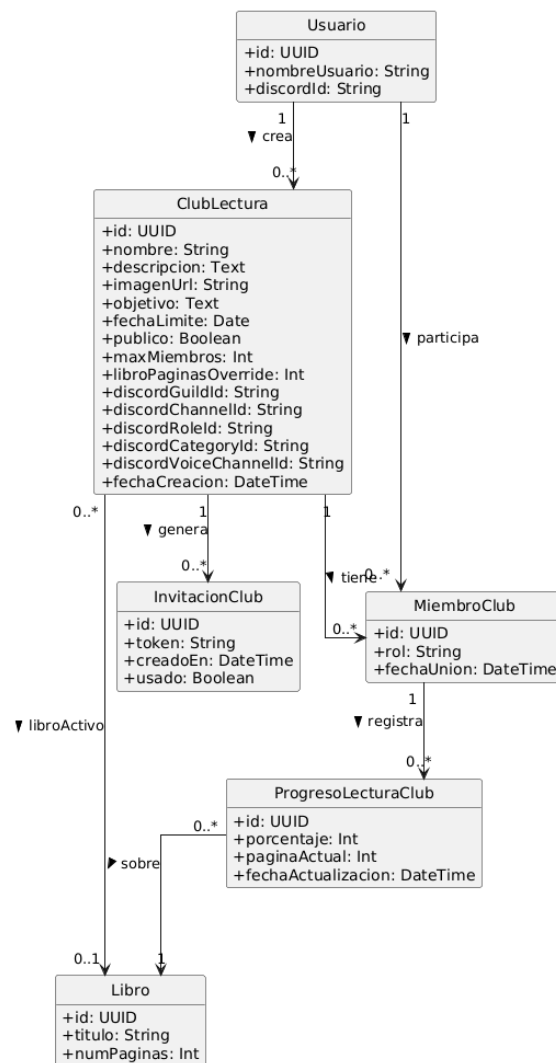


Fig. 14: Diagrama Entidad-Relación: clubes de lectura

A.3. Diagrama de arquitectura

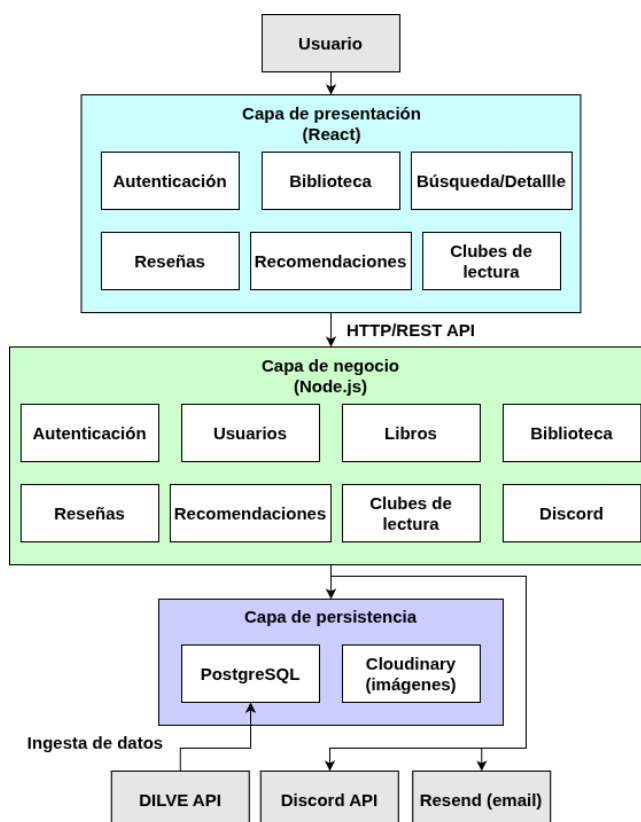


Fig. 15: Diagrama de arquitectura

A.4. Resultados adicionales

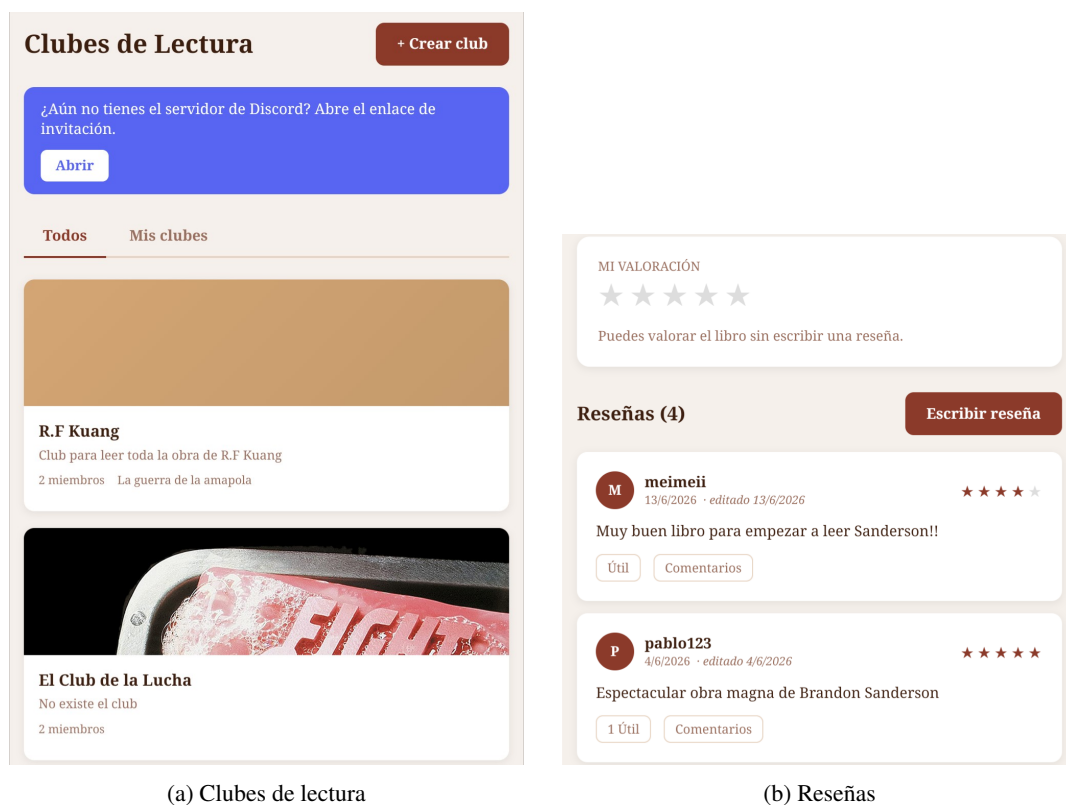


Fig. 16: Pantallas adicionales de la aplicación