

# Plan de gestión, análisis, diseño y memoria del proyecto Bookly



Bookly

06. Susan Kare

[Organización de Github](#)

Luna Zhou Chen ([812103](#)), Lucía Vázquez Martín ([871886](#)), Iván Vayad Díaz ([874762](#)), Javier Susín Villacampa ([872847](#)), Diego Simón Íñiguez ([869800](#)), Ariana Porroche Llorén ([874055](#)), Irene Pascual Albericio ([871627](#)) y Alex Giménez Garrido ([874113](#))

14/05/2025

# Documentación del informe

Revisiones del documento:

| Miembro                        | Versión | Fecha      |
|--------------------------------|---------|------------|
| Alex Giménez                   | 0.1     | 18-02-2025 |
| Ariana Porroche                | 0.2     | 02-03-2025 |
| Alex Giménez                   | 0.3     | 04-03-2025 |
| Ariana Porroche                | 0.4     | 05-03-2025 |
| Alex Giménez                   | 0.5     | 06-03-2025 |
| Ariana Porroche y Alex Giménez | 0.6     | 11-03-2025 |
| Ariana Porroche                | 0.7     | 12-03-2025 |
| Ariana Porroche y Alex Giménez | 0.8     | 13-03-2025 |
| Alex Giménez                   | 0.9     | 14-03-2025 |
| Ariana Porroche                | 0.10    | 15-03-2025 |
| Alex Giménez                   | 0.11    | 24-03-2025 |
| Alex Giménez                   | 0.12    | 11-04-2025 |
| Ariana Porroche                | 0.13    | 19-04-2025 |
| Alex Giménez                   | 0.14    | 20-04-2025 |
| Alex Giménez                   | 0.15    | 27-04-2025 |
| Alex Giménez y Ariana Porroche | 0.16    | 11-05-2025 |
| Ariana Porroche                | 0.17    | 14-05-2025 |

[D.1]. Tabla de revisiones del documento.

Identificador único del informe: **PGADM\_06**

Este informe consta de **65 páginas** y de unas **15.350 palabras**.

Última modificación del documento en **fecha 14-05-2025** y **versión** del documento **0.17**.

# Resumen

Este documento detalla el desarrollo del proyecto Bookly, una plataforma digital de lectura de libros electrónicos en formato PDF. Describimos los objetivos, el alcance y las funcionalidades clave del sistema, incluyendo la gestión de bibliotecas virtuales, la sincronización de progreso de lectura entre dispositivos y herramientas de interacción social. Establecemos los hitos principales del proyecto y los entregables asociados, organizados en dos iteraciones de desarrollo.

La memoria documenta el proceso de planificación, ejecución y control del proyecto, abordando decisiones técnicas, metodologías aplicadas y ajustes realizados durante su desarrollo. Hemos incluido el análisis de requisitos, diseño del sistema, estrategias de desarrollo y pruebas implementadas para garantizar la funcionalidad y usabilidad del sistema. Presentamos los resultados clave, como la implementación de cuentas de usuario, la gestión de lectura de libros y de listas de libros personalizadas para cada usuario, la optimización y apariencia de la interfaz y la sincronización entre web y móvil.

Además, discutimos los desafíos enfrentados durante el desarrollo, las soluciones implementadas y las lecciones aprendidas en el proceso. Finalmente, ofrecemos conclusiones basadas en los aprendizajes obtenidos a lo largo del proyecto.

# Tabla de contenidos

|                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------|----|
| Resumen.....                                                                                                        | 1  |
| 1. Introducción.....                                                                                                | 1  |
| 2. Organización del proyecto.....                                                                                   | 3  |
| 3. Plan de gestión del proyecto.....                                                                                | 5  |
| 3.1. Inicio del proyecto.....                                                                                       | 5  |
| 3.1.1. Asignación de recursos.....                                                                                  | 5  |
| 3.1.2. Riesgos del proyecto.....                                                                                    | 5  |
| 3.1.3. Formación inicial del equipo.....                                                                            | 5  |
| 3.2. Control de configuraciones, construcción del software y aseguramiento de la calidad del producto.....          | 7  |
| 3.2.1. Responsables.....                                                                                            | 7  |
| 3.2.2. Convenciones de nombrado y estándares de código.....                                                         | 7  |
| 3.2.3. Repositorios de control de versiones.....                                                                    | 7  |
| 3.2.4. Métodos, herramientas y técnicas.....                                                                        | 8  |
| 3.2.5. Cambios en código y documentación.....                                                                       | 8  |
| 3.2.6. Construcción e integración del software.....                                                                 | 9  |
| 3.2.7. Despliegue del software.....                                                                                 | 9  |
| 3.2.8. Guías para diseño de software y diseño gráfico de GUI.....                                                   | 10 |
| 3.2.9. Actividades para el control de calidad.....                                                                  | 10 |
| 3.2.10. Entrega de resultados al cliente.....                                                                       | 11 |
| 3.3. Calendario del proyecto, división del trabajo, coordinación, comunicaciones, monitorización y seguimiento..... | 12 |
| 3.3.1. Responsables.....                                                                                            | 12 |
| 3.3.2. Diagrama de Gantt.....                                                                                       | 12 |
| 3.3.3. División del trabajo.....                                                                                    | 15 |
| 3.3.4. Comunicación interna.....                                                                                    | 16 |
| 3.3.5. Determinación de tareas a realizar y reparto.....                                                            | 16 |
| 3.3.6. Medidas de progreso y monitorización de estado del proyecto.....                                             | 17 |
| 3.3.7. Gestión del equipo humano.....                                                                               | 17 |
| 4. Análisis y diseño del sistema.....                                                                               | 18 |
| 4.1. Análisis de requisitos.....                                                                                    | 18 |
| 4.1.1. Stakeholders.....                                                                                            | 18 |
| 4.1.2. Diccionario de datos.....                                                                                    | 18 |
| 4.1.3. Requisitos funcionales.....                                                                                  | 19 |
| 4.1.4. Requisitos no funcionales.....                                                                               | 21 |
| 4.2. Diseño del sistema.....                                                                                        | 23 |
| 4.2.1. Vistas de módulos.....                                                                                       | 23 |
| 4.2.2. Vista de componentes y conectores.....                                                                       | 25 |
| 4.2.3. Vista de despliegue.....                                                                                     | 27 |
| 4.2.4. Diseño de la base de datos.....                                                                              | 28 |
| 5. Memoria del proyecto.....                                                                                        | 30 |
| 5.1. Inicio del proyecto.....                                                                                       | 30 |

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| 5.2. Ejecución y control del proyecto.....                                           | 31 |
| 5.2.1. Funcionamiento del reparto de trabajo.....                                    | 31 |
| 5.2.2. Funcionamiento de la comunicación interna.....                                | 31 |
| 5.2.3. Funcionamiento de la medición del progreso y el conocimiento del avance...    | 31 |
| 5.2.4. Ajustes al calendario inicial.....                                            | 32 |
| 5.2.5. Adecuación de las herramientas y tecnologías empleadas.....                   | 32 |
| 5.2.6. Funcionamiento de procesos de control del código, construcción y despliegue.. | 33 |
| 5.2.7. Pruebas del software.....                                                     | 33 |
| 5.3. Cierre del proyecto.....                                                        | 34 |
| 5.3.1. Comparación estimaciones iniciales con resultados finales.....                | 34 |
| 5.3.2. Riesgos presentes.....                                                        | 35 |
| 5.3.3. Lecciones aprendidas sobre herramientas y tecnologías.....                    | 35 |
| 5.3.4. Recopilación de esfuerzos.....                                                | 37 |
| Conclusiones.....                                                                    | 38 |
| Bibliografía y referencias.....                                                      | 41 |
| Apéndice I: Análisis de riesgos.....                                                 | 42 |
| Anexo I. Resultados de la revisión intermedia.....                                   | 43 |
| Anexo II. Presentación final.....                                                    | 44 |
| Glosarios.....                                                                       | 56 |
| Índice de figuras.....                                                               | 59 |
| Índice de tablas.....                                                                | 60 |

# 1. Introducción

El proyecto trata sobre una plataforma digital diseñada para la **lectura y gestión de libros electrónicos** en formato PDF. La aplicación permitirá a los usuarios acceder a una **biblioteca virtual** en la que podrán organizar sus libros, **realizar búsquedas**, **gestionar listas de lectura** personalizadas y **compartir opiniones** mediante valoraciones y comentarios. Además, el sistema incluirá funcionalidades de **sincronización de progreso** de lectura entre dispositivos, **interacción social** mediante un foro en el que los usuarios pueden lanzar preguntas y responder preguntas de otros usuarios. También se podrán visualizar las **estadísticas** sobre usuarios y libros más leídos mensualmente y anualmente.

El propósito principal del proyecto es proporcionar una experiencia de lectura fluida y optimizada en una plataforma **accesible desde dispositivos web y móviles**. La aplicación estará diseñada para usuarios registrados y no registrados, permitiendo la gestión eficiente de colecciones de libros, la exploración de nuevas lecturas y la interacción con otros lectores. El sistema utilizará tecnologías modernas para garantizar tiempos de carga rápidos, compatibilidad con diferentes dispositivos y una interfaz intuitiva.

Los objetivos principales del proyecto son los siguientes:

- Facilitar la **lectura de libros en formato PDF** con opciones de navegación, zoom y marcadores.
- Permitir la organización de libros en **listas personalizadas y predefinidas**.
- **Sincronizar el progreso** de lectura entre dispositivos.
- Brindar herramientas de **interacción social**, como valoraciones, reseñas y un foro de discusión.
- Ofrecer **estadísticas personalizadas** sobre los hábitos de lectura de los usuarios.

El desarrollo del sistema se dividirá en dos iteraciones: una entrega intermedia y una entrega final. Hemos dividido las funcionalidades en dos grupos, asignando las más relevantes y críticas a la entrega intermedia, y las demás a la entrega final.

Por otra parte, los entregables finales serán:

- Presentación final del proyecto
- Software web en funcionamiento en un servidor.
- Archivo APK (*Android Package*) para la versión de móvil.
- Documentación sobre el funcionamiento del sistema y sobre la estructura del código.

Este documento está compuesto de varias secciones a continuación de esta introducción, las cuales describen detalladamente el proyecto. La **sección 2** (Organización del Proyecto) detalla el equipo de trabajo, sus roles y responsabilidades. La **sección 3** (Plan de Gestión del Proyecto) expone los procesos clave para su ejecución, incluyendo el inicio, control de configuraciones y aseguramiento de calidad. Posteriormente, la **sección 4** (Análisis y Diseño del Sistema) abarca los requisitos técnicos, la arquitectura y composición de la solución

dada. La **sección 5** (Memoria del Proyecto) documenta la ejecución, control y cierre del mismo, seguido de la sección de **Conclusiones**, que resume los aprendizajes obtenidos y reflexiones finales. Finalmente, el documento incorpora una **bibliografía** con las referencias citadas, un **apéndice** con el análisis de riesgos realizado, un **anexo** con los resultados de la revisión intermedia, un **glosario** de términos con términos y acrónimos utilizados en este documento, y finalmente el **índice de figuras** y el **índice de tablas**, donde para cada una de ellas dejamos claro a qué corresponde cada tabla y cada figura y en qué página se sitúan.

## 2. Organización del proyecto

El equipo está compuesto por 8 miembros, los cuales han adquirido roles y responsabilidades diferentes, aunque el trabajo se ha distribuido de forma que varios integrantes realicen tareas variadas y centradas en un aspecto en concreto.

Al principio de la primera iteración del proyecto, nos centramos en desarrollar el backend para que los equipos del frontend pudieran avanzar. La distribución seguida fue:

| Backend                                                      | Frontend-móvil               | Frontend-web                |
|--------------------------------------------------------------|------------------------------|-----------------------------|
| Javier Susín<br>Diego Simón<br>Irene Pascual<br>Alex Giménez | Luna Zhou<br>Ariana Porroche | Iván Vayad<br>Lucía Vázquez |

[2.1]. Tabla de subgrupos de trabajo primera fase

Para el resto del proyecto, redistribuimos los equipos resultando en la siguiente división:

| Backend                     | Frontend-móvil                               | Frontend-web                                 |
|-----------------------------|----------------------------------------------|----------------------------------------------|
| Javier Susín<br>Diego Simón | Luna Zhou<br>Ariana Porroche<br>Alex Giménez | Iván Vayad<br>Lucía Vázquez<br>Irene Pascual |

[2.2]. Tabla de subgrupos de trabajo segunda fase

Por otra parte, los roles y responsabilidades de cada uno se resumen en la siguiente tabla:

| Miembro       | Rol                                                                                                                       | Responsabilidad                                                                                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Lucía Vázquez | Desarrolladora frontend web; directora del desarrollo web, desarrolladora pruebas web; responsable calidad producto       | Desarrolladora y diseñadora de pantallas y funciones en web.<br>Verificar que el progreso del desarrollo web es correcto, se cumplen los plazos y los objetivos.<br>Desarrollar las pruebas de web.<br>Garantizar calidad del software. |
| Iván Vayad    | Desarrollador frontend web; encargado de la demostración web; desarrollador pruebas web; responsable comunicación interna | Desarrollador y diseñador de pantallas y funciones en web.<br>Desarrollador de la demostración del sistema en web.<br>Desarrollar las pruebas web.<br>Gestionar comunicaciones diarias y realizar actas post reuniones.                 |
| Javier Susín  | Desarrollador backend y frontend web; director de despliegue y de backend;                                                | Integración de backend con frontend.<br>Desarrollar funciones backend.<br>Desarrollar la lectura de libros en web.                                                                                                                      |

|                 |                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | responsable control configuraciones y gestión repositorios.                                                                                               | Desarrollar y verificar el despliegue de backend, web y móvil.<br>Verificar que el progreso del desarrollo de backend es correcto, se cumplen los plazos y los objetivos.<br>Gestión de las versiones del código utilizando GitHub                                                                                                                                                                                                                                                       |
| Diego Simón     | Desarrollador backend, frontend web; despliegue web y backend; desarrollador pruebas backend; responsable control configuraciones y gestión repositorios. | Integración de backend con frontend.<br>Desarrollar funciones backend.<br>Documentador del uso de GitHub.<br>Desarrollar el despliegue de web y backend.<br>Desarrollar las pruebas de backend.<br>Gestión de las versiones del código utilizando GitHub                                                                                                                                                                                                                                 |
| Irene Pascual   | Desarrolladora de backend y frontend web; realizadora de presentaciones; responsable coordinación interna                                                 | Desarrollar funciones backend.<br>Desarrollar y diseñar pantallas en web.<br>Crear la presentación final del proyecto.<br>Organizar reuniones semanales y revisar progreso para solventar dudas.                                                                                                                                                                                                                                                                                         |
| Alex Giménez    | Desarrollador de backend y frontend móvil; despliegue móvil; documentador; desarrollador pruebas móvil; responsable calendario y seguimiento del progreso | Gestión de la documentación del proyecto.<br>Desarrollar y diseñar pantallas en móvil.<br>Desarrollar funciones backend.<br>Desarrollar el despliegue de móvil.<br>Desarrollar las pruebas de móvil.<br>Realizar planificación general y actualizar diagrama de Gantt.                                                                                                                                                                                                                   |
| Luna Zhou       | Desarrolladora frontend móvil; encargada de la demostración móvil; desarrolladora pruebas móvil; responsable división del trabajo                         | Desarrolladora y diseñadora de pantallas y funciones en móvil.<br>Desarrolladora de la demostración del sistema en móvil.<br>Desarrollar las pruebas de móvil.<br>Asignar tareas a miembros del equipo y gestionar GitHub Issues.                                                                                                                                                                                                                                                        |
| Ariana Porroche | Directora del proyecto y de frontend móvil; desarrolladora frontend móvil; documentadora; responsable calidad producto; responsable de software           | Decidir bloques de tareas significativos y asignarlos a los demás miembros.<br>Verificar que el progreso del desarrollo móvil es correcto, se cumplen los plazos y los objetivos.<br>Gestión de la documentación del proyecto.<br>Desarrolladora y diseñadora de pantallas y funciones en móvil.<br>Integración de backend con frontend.<br>Desarrollar y asegurar la lectura de libros en móvil.<br>Garantizar la calidad del software.<br>Garantizar producto final cumple requisitos. |

[2.3]. Tabla de roles y responsabilidades por miembro del equipo.

## 3. Plan de gestión del proyecto

### 3.1. Inicio del proyecto

#### 3.1.1. Asignación de recursos

Para el desarrollo del proyecto, necesitaremos varios proveedores de servicios. Será necesario contar con un **almacenamiento en la nube** para los libros, imágenes y documentación, así como una **base de datos en la nube** para gestionar la información del sistema. También requeriremos herramientas para el **diseño de la interfaz** de usuario y un **servidor donde desplegar** nuestra aplicación al completo, tanto backend como frontend.

Además, necesitaremos una **IA** disponible y accesible mediante API para la creación de un asistente en la aplicación. Por último, para la **realización de pruebas** en la aplicación móvil, se tiene previsto usar dispositivos móviles propios, atendiendo a la necesidad de compatibilidad con Android 5.0 o superior, dado que esta es la versión mínima requerida por React-Native.

#### 3.1.2. Riesgos del proyecto

En la **tabla de análisis de riesgos** [\[Al.1\]](#), detallamos los posibles riesgos, así como su probabilidad e impacto, y en la **tabla de estrategias de mitigación** [\[Al.2\]](#) se nombran las posibles estrategias de mitigación de dichos riesgos, así como sus responsables ([ver Apéndice I](#)). Estos son los principales riesgos identificados en el proyecto:

- Algunos integrantes tienen **mucha carga de trabajo**, para lo cual dividiremos el trabajo y adelantaremos gran parte de este.
- Parte del equipo **nunca ha trabajado junta** antes, lo que trataremos de solucionar con una amplia comunicación de grupo, compartiendo los conocimientos y habilidades de cada uno y organizando reuniones.
- Se tiene **gran desconocimiento del cliente**, por lo que compartiremos todas las dudas que tenemos con él y organizamos entrevistas en las que dejamos claros los temas tratados.
- Se ha elegido una **tecnología que no se conoce**, para lo cual realizaremos una formación inicial mediante los integrantes del equipo que sí la conocen y mediante ayuda externa.

#### 3.1.3. Formación inicial del equipo

Nos vamos a formar en varias tecnologías para desarrollar el proyecto de manera efectiva. Como ninguno de nosotros ha trabajado con Vue.js ni React Native, dedicaremos tiempo a aprender su funcionamiento, de forma que el equipo de frontend web profundizará en Vue.js

y el equipo de móvil profundizará en React Native. Los miembros de backend que no hayan utilizado Node.js o Express profundizarán en los mismos para poder gestionar el backend.

Por otra parte, la mayoría del equipo no tiene experiencia previa con GitHub, por lo que quienes ya la conocen prepararán un documento con instrucciones básicas sobre su uso. Esto nos permitirá integrarnos más rápidamente y evitar problemas al gestionar los repositorios del proyecto.

Para adquirir estos conocimientos, utilizaremos tutoriales de YouTube y otros recursos en línea, lo que nos permitirá avanzar a nuestro propio ritmo. A medida que surjan problemas durante el desarrollo, investigaremos soluciones y aprenderemos de la experiencia. También recurriremos a herramientas de inteligencia artificial, como ChatGPT, para resolver dudas y comprender mejor ciertos conceptos. Con este enfoque, nos aseguraremos de adaptarnos rápidamente a las tecnologías que necesitamos.

## 3.2. Control de configuraciones, construcción del software y aseguramiento de la calidad del producto

### 3.2.1. Responsables

- **Responsables del control de configuraciones y gestión de repositorios:** Javier Susín Villacampa y Diego Simón Íñiguez. Gestionarán las versiones del código utilizando GitHub, asegurando que todos los miembros trabajan con la misma estructura y sin conflictos.
- **Responsables del aseguramiento de la calidad del producto:** Ariana Porroche Llorén y Lucía Vázquez Martín. Garantizarán la calidad del software desarrollado, mediante revisiones periódicas del código y pruebas exploratorias.
- **Responsable del software:** Ariana Porroche Llorén. Garantizará que el producto final desarrollado cumple con los requisitos iniciales, así como la entrega del mismo.

### 3.2.2. Convenciones de nombrado y estándares de código

Los **archivos** seguirán unas directrices para su nombrado, a los archivos de código les daremos nombre en función de su funcionalidad y los organizaremos en carpetas para diferenciar unos de otros, de forma que un archivo referido a la pantalla de iniciar sesión podría tener un nombre tal que "IniciarSesion.js" y en la aplicación móvil, se encontraría en la carpeta "pantallas", mientras que en web se encontraría en la carpeta "components". Cabe destacar que en el apartado web, Vue genera una estructura propia de carpetas a seguir. No seguimos guías específicas de código externas, pero mantenemos una estructura y convención común de grupo que todos debemos respetar para poder comprender el código de cualquier miembro del equipo. Esto garantiza que cualquier miembro pueda encontrar y modificar código independientemente de su módulo de trabajo. Por otra parte, los **archivos de documentación** seguirán la lengua española y su nombre derivará en función del objetivo de dicho documento. Para identificarlos como archivos propios, utilizaremos "06.SK-" como inicio del nombre de dichos documentos, además de que este prefijo estará seguido del nombre como tal del documento.

Además, para el diseño y documentación de la API REST, seguimos el estándar **OpenAPI**, lo que nos permite definir de manera estructurada los endpoints, parámetros, respuestas y autenticación. Esto facilita la integración con otros sistemas y la generación de documentación automática para los desarrolladores.

### 3.2.3. Repositorios de control de versiones

Para este proyecto utilizaremos **GitHub** como base para compartir código, para ello crearemos una organización llamada "**UNIZAR-30226-2025-60**", en la cual crearemos **3 repositorios**: "backend", "frontend-web" y "frontend-movil", los cuales como sus nombres indican, corresponden al código de backend, frontend para web y frontend para móvil respectivamente. En todos ellos seguiremos la misma forma de trabajo, tenemos una rama

principal “main”, mientras que en paralelo tenemos otras ramas en las cuales se avanzan las funcionalidades o se corrigen errores. Cuando la funcionalidad esté desarrollada al completo, se realizará un *merge* con la rama main.

Los permisos de acceso de los miembros del equipo corresponderán a “Owner” en la organización mencionada previamente, de forma que todos puedan acceder los repositorios y editarlos, pudiendo gestionar *issues* en los repositorios y modificar el código y los datos de los repositorios.

Para la creación de la base de datos tendremos un archivo .sql en la ruta “backend/creacionBD.sql”, así como otro archivo .sql para la creación específica de los triggers en la ruta “backend/triggers.sql”. Se añadirán instrucciones para el despliegue en los archivos README.md de los repositorios correspondientes, donde explicaremos cómo lanzar en local la base de datos, la web, lanzar en un emulador la aplicación móvil o cómo generar el APK, además del acceso a la web desde URL.

### 3.2.4. Métodos, herramientas y técnicas

Como herramientas de construcción del software utilizaremos **VSCode** como IDE y **GitHub** para gestión de versiones y construcción. También utilizaremos **Git Flow** como técnica de desarrollo, mediante el uso de ramas en GitHub, además de la división del código en módulos para su reutilización. Además, para probar el software, haremos uso de **Expo Go** para la simulación de un dispositivo móvil en el ordenador, el navegador web **Google Chrome** para la aplicación web, y **Postman** para probar los accesos a la base de datos. Por último, queremos destacar que haremos uso de una **base de datos relacional SQL** y que el **despliegue** desde la nube, permitirá un acceso automático desde GitHub.

### 3.2.5. Cambios en código y documentación

Para la realización de cambios en el código fuente utilizaremos las **ramas de GitHub**, de forma que si alguien quiere realizar un cambio o añadir una nueva funcionalidad, crea una rama, hace los cambios y realiza el **commit**. Los commits no deben ser verificados por nadie antes de ser realizados, salvo por el mismo usuario que ha creado la rama, siendo este el responsable de comprobar que no haya fallos y que no haya afectado a ninguna otra funcionalidad previa. A la hora de crear *issues*, se debe indicar en el título de la misma lo que representa, y si es necesario aclarar su significado se puede añadir una descripción, por otra parte cada *issue* debe tener un estado entre los siguientes:

- **Todo**: tarea pendiente por realizar.
- **In progress**: tarea en proceso de desarrollo.
- **Done**: tarea completada y funcional.

Al completar una funcionalidad, el issue correspondiente se cierra y se asigna a los miembros que han participado en su desarrollo.

Para los cambios en la documentación técnica, **Google Docs** ya cuenta con un historial de versiones, el cual utilizaremos para observar los cambios realizados y volver a alguna versión anterior si es necesario. Además, utilizaremos los comentarios para comunicar dudas, correcciones o secciones a revisar.

### 3.2.6. Construcción e integración del software

La construcción e integración del software se lleva a cabo mediante integración continua (CI) utilizando GitHub Actions. Se han definido distintos workflows para automatizar el proceso de construcción y verificación del backend, la aplicación web y la aplicación móvil (Expo/Android).

Cada workflow ejecuta una serie de pasos que incluyen: la descarga del repositorio, la instalación de dependencias con Node.js, la ejecución de pruebas automáticas en el backend y la compilación de los proyectos en sus respectivas plataformas. En el caso de la aplicación móvil, también se realiza una fase de preconstrucción con Expo y una compilación del proyecto nativo usando Gradle. Para garantizar consistencia en los entornos de desarrollo, se utiliza la misma versión de Node.js (v20) en todos los trabajos.

Además, se emplean variables de entorno seguras, definidas en los secrets del repositorio, para manejar configuraciones sensibles como credenciales y URL de bases de datos. Se desarrollarán scripts adicionales que aseguren que todos los integrantes del equipo puedan compilar y ejecutar el software localmente bajo las mismas condiciones y dependencias.

### 3.2.7. Despliegue del software

Para el despliegue del software se utilizarán servidores en la nube como Vercel o Render que nos permitirán lanzar a producción integrando directamente de los repositorios de Github. Estas plataformas automatizan la obtención de los últimos commits en el repositorio y los lanzan automáticamente. Para que todo funcione correctamente además tendremos que proporcionar el .env con todo lo necesario para que el sistema funcione correctamente tanto en local como en producción. De esta forma pretendemos que el trabajo se pueda realizar de manera local utilizando el backend local para desarrollar, y en Render o Vercel usando el backend lanzado también de la misma manera en estas plataformas.

El backend es lo primero que desplegaremos. Desde Render o Vercel se facilitará el enlace de acceso [\[1\]](#), que nos permitirá seguir con el desarrollo de la aplicación en ambos clientes. Al desplegarse en estas plataformas, el servidor estará en reposo hasta que reciba una solicitud desde el frontend web o el cliente móvil. Esto significa que no se ejecutará de forma constante, sino que se activará solo cuando un cliente (web o móvil) haga una solicitud. Una vez que se reciba la solicitud, el backend procesará la petición y enviará la respuesta correspondiente.

Como las pruebas y comprobaciones en tiempo real de ambos clientes las haremos en local, usaremos variables de entorno para detectar si estamos en el entorno de desarrollo o de producción. Con respecto al cliente móvil, vamos a hacer uso de la variable “\_\_DEV\_\_”. Cuando esta es false, se hará uso de la conexión a la base de datos Neon a través de Render o Vercel. En caso contrario, se usarán las conexiones locales. El archivo “config.js” leerá la variable “\_\_DEV\_\_” y generará una variable global que contiene el path adecuado. El caso del cliente web es algo distinto, “window.location.hostname” será la variable que nos permitirá comprobar si la actividad del cliente está en desarrollo o en producción, y de la misma forma que en móvil definirá la ruta necesaria.

La aplicación web la lanzaremos también en Render o Vercel, funcionando de la misma forma que el backend y haciendo uso siempre del backend de Render. Como ya se ha explicado se detectan automáticamente los commits y se actualiza el enlace.

Por su parte, el cliente móvil se construirá con el comando “./gradlew assembleRelease”, el cual generará un APK (Android Package) instalable en Android, que hará uso del backend en Render. Pero este APK sí que hay que generarlo manualmente cada vez que hagamos cambios en el código.

### 3.2.8. Guías para diseño de software y diseño gráfico de GUI

Para la creación de **documentos técnicos** utilizaremos **Google Docs**, de forma que sea accesible para todos los integrantes del equipo mediante Google Drive. Por otra parte, los **diagramas** se crearán mediante el uso de **excel** y **draw.io**.

Para el **diseño gráfico de las GUI** (*Graphical User Interface*) utilizaremos la herramienta colaborativa **Figma**. Seguiremos la guía de accesibilidad de WCAG [\[2\]](#) (Web Content Accessibility Guidelines) así como la guía de Material Design, sobre el uso de colores en el diseño de la interfaz de usuario y sus contrastes, siguiendo así la propia guía de diseño de Android [\[3\]](#). En dicha guía se establece una **relación mínima de contraste** entre los colores empleados, la cual debe de ser como mínimo de 4.5:1 para el texto y de 3:1 para el fondo y un icono. Dicha relación de contraste puede ser medida con cualquier página web, aunque debe basarse en el ratio de contraste 2.1 de WCAG.

### 3.2.9. Actividades para el control de calidad

Para realizar un control de calidad apropiado en el producto realizaremos diferentes actividades, en primer lugar el sistema web será sometido a inspecciones por parte de los miembros de los sectores de backend y de móvil, repasando que se cumplen todos los requisitos funcionales y una inspección de interfaz respecto al diseño inicial de GUI. Por otra parte, el sistema móvil será sometido al mismo método con los miembros de los equipos de backend y de web. Por último, ambos sistemas, tanto web como móvil, se someterán a una checklist en la cual se verificarán las pantallas con las que deberían contar ambos sistemas, así como verificación de la similitud entre web y móvil.

Con el objetivo de asegurar que el producto desarrollado cumpla con los requisitos acordados y mantengamos un alto nivel de calidad, llevaremos a cabo diversas actividades estructuradas de control de calidad, abarcando tanto la calidad del producto como la del proceso, tal y como se promueve en los modelos de calidad como el ISO 9001.

- **Revisiones por pares:** realizaremos revisiones cruzadas entre equipos de trabajo. Centraremos las revisiones en asegurar la correspondencia entre lo implementado y los requisitos funcionales definidos, así como la fidelidad al diseño visual.
  - El sistema web será inspeccionado por los miembros de los equipos de backend y móvil.
  - El sistema móvil será revisado por los equipos de backend y web.
- **Checklist de cobertura funcional:** Emplearemos una lista de comprobación basada en los requisitos funcionales (RFs) y no funcionales (RNFs) para verificar que todas las funcionalidades estén implementadas y correctamente integradas. Esto incluye la revisión de pantallas, navegación y similitud entre versiones web y móvil.
- **Revisión de documentación y requisitos:** Los documentos de análisis, diseño y memoria serán sometidos a revisión interna. Validaremos que los requisitos estén correctamente trazados y reflejados en el diseño del sistema. Asimismo, Revisaremos los diagramas del sistema y los procesos y mecanismos descritos, verificando que estén alineados con la arquitectura técnica final.
- **Pruebas manuales:** Realizaremos pruebas exploratorias por parte de los distintos equipos, siguiendo escenarios de uso reales. Estas pruebas permitirán detectar errores de usabilidad, flujos rotos o problemas visuales no detectados en fases anteriores. Para asegurar la calidad de la experiencia de usuario, se probarán distintos dispositivos con diferentes tamaños de pantalla, tanto para web como para móvil.

### 3.2.10. Entrega de resultados al cliente

La entrega de resultados al cliente se realizará en dos fases: una entrega intermedia y una entrega final, conforme a lo establecido en el acuerdo técnico y el calendario de reuniones previamente fijado.

- **Entrega intermedia:** Esta entrega incluirá una versión funcional parcial del sistema, accesible desde un servidor privado. Las funcionalidades previstas son: gestión de cuentas y perfiles de usuario, listado de libros con buscador y filtros por categorías, detalles del libro (sinopsis, autor, valoraciones...), lector de libros en formato PDF, y listas del propio usuario. También se entregará una versión preliminar de la memoria del proyecto, y se realizará una reunión con el cliente para revisar el cumplimiento de los objetivos funcionales y recoger sugerencias de mejora.
- **Entrega final:** En esta fase se entregará el sistema completo, accesible en entorno público y estable. Se incluirán funcionalidades adicionales como listas de otros usuarios, sistema de favoritos, foro, estadísticas de lectura, libros leídos y en curso, y modo claro/oscuro.

En la **entrega final** entregaremos lo siguiente:

- Software web en funcionamiento, desplegado en un servidor web.
- Archivo APK para la versión móvil Android.
- Documentación completa mediante el plan de gestión, análisis, diseño y memoria..
- Presentación final del proyecto.

El **código fuente no será entregado al cliente**, tal y como se acordó previamente. La entrega se centrará en el acceso al sistema, los ejecutables generados y la documentación correspondiente.

### 3.3. Calendario del proyecto, división del trabajo, coordinación, comunicaciones, monitorización y seguimiento

#### 3.3.1. Responsables

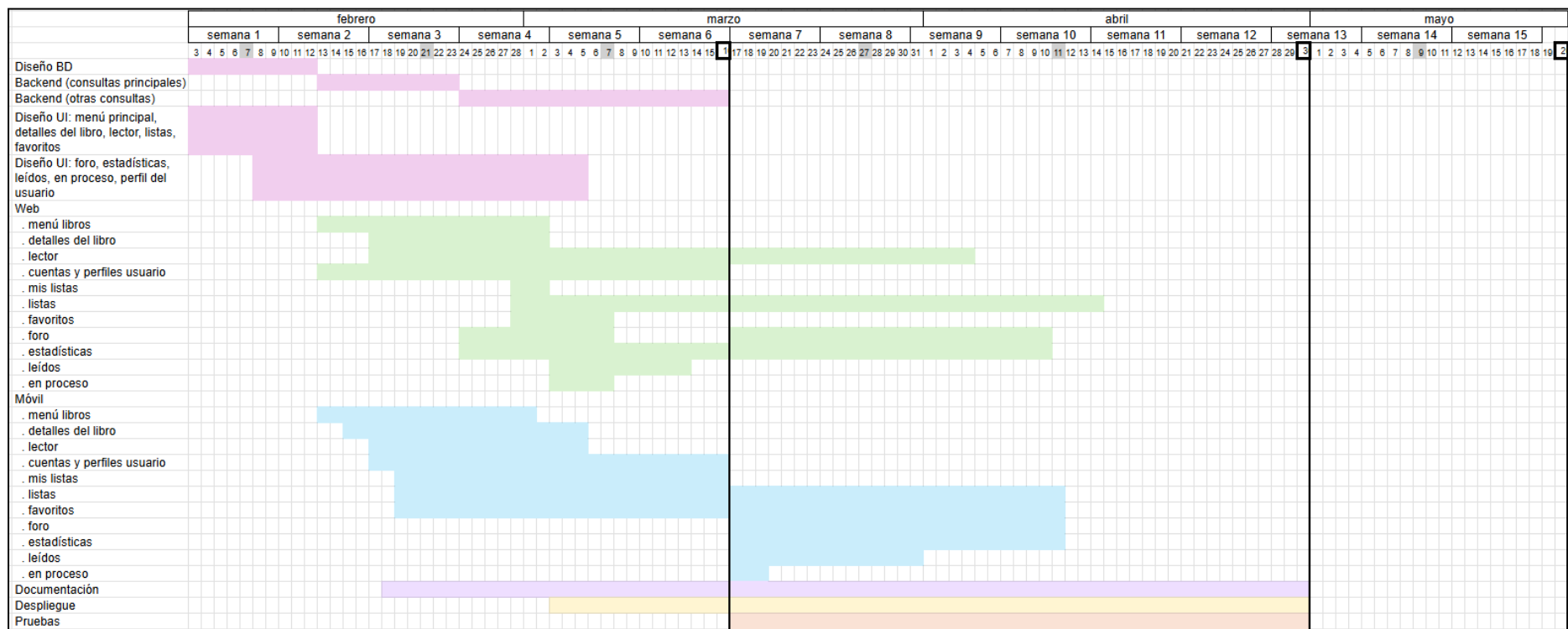
Para controlar la puntualidad, organización y monitorización del proyecto se han otorgado diferentes responsabilidades a los diferentes miembros del equipo:

- **Responsable del calendario y seguimiento del progreso: Alex Giménez Garrido.** Será responsable de la planificación general del proyecto, incluyendo la creación y actualización del **Diagrama de Gantt**, y del seguimiento semanal del avance en relación con las fechas establecidas.
- **Responsable de la división del trabajo: Luna Zhou Chen.** Asignará tareas a cada miembro del equipo, basándose en las habilidades y carga de trabajo de cada persona. Además, gestionará la actualización de las tareas en **GitHub Issues** para garantizar que cada miembro del equipo tenga claro lo que debe hacer.
- **Responsable de la coordinación interna: Irene Pascual Albericio.** Organizará las reuniones semanales del equipo, tanto para revisar el progreso de las tareas como para resolver cualquier duda técnica que surja. Además, será responsable de coordinar los esfuerzos entre los equipos de frontend y backend para asegurar que todas las partes del proyecto se integren correctamente.
- **Responsable de comunicaciones internas: Iván Vayad Díaz.** Gestionará las comunicaciones diarias entre los miembros del equipo utilizando **Whatsapp** y **Discord**. También será responsable de crear y mantener el registro de decisiones y actas de reuniones en **Google Docs**, y de distribuir la información relevante a todos los integrantes del equipo.

#### 3.3.2. Diagrama de Gantt

El **diagrama de Gantt** del proyecto [\[3.1\]](#), muestra cómo han sido planificadas las tareas, así como las dependencias entre tareas y las fechas e hitos importantes del proyecto. Podemos observar una línea que muestra la frontera entre la entrega intermedia y la entrega final. Los días marcados en color gris indican las reuniones obligatorias del equipo. A su vez,

observamos las tareas que tienen dependencia de otras, por lo que han sido planificadas secuencialmente.



[3.1]. Diagrama de Gantt del proyecto

### 3.3.3. División del trabajo

El trabajo a realizar no se va a dividir desde un principio asignando tareas a cada uno, se realizará una división del equipo entre las áreas principales: **backend**, **frontend-móvil** y **frontend-web**, y en cada una de dichas áreas se asignarán *Issues* en GitHub correspondientes a las tareas pendientes de realizar. De esta forma, y al no tener un trabajo preestablecido desde el inicio, los miembros de cada área revisan las *Issues* pendientes de completar y seleccionan una para realizar. Mediante este método nos aseguramos de no sobrecargar a nadie con demasiado trabajo, de forma que todos puedan avanzar a un ritmo que puedan mantener en el tiempo y así cada uno pueda comparar su trabajo y horas dedicadas con el resto para saber si está realizando un buen trabajo, un excesivo trabajo o un trabajo escaso.

Hemos pensado en sobrecargar desde el inicio el área de backend para finalizarla lo antes posible, de forma que los integrantes de esa área se redistribuyan en otras áreas. De esta forma, un integrante de backend será asignado a frontend web y otro a frontend móvil. Los otros 2 miembros se mantendrán en backend, pendientes de consultas pendientes que los integrantes de frontend detecten, pero desarrollarán el proceso de **despliegue** del backend y el de frontend con ayuda de alguno de los integrantes de dichas áreas. Por otra parte, la **documentación** quedará a cargo de uno de los miembros de backend, el desarrollo de **presentaciones y demostraciones** y será desarrollado por los miembros de las áreas más avanzadas y las **pruebas** se realizarán al finalizar el trabajo de cada área de trabajo por los miembros de dicha área. Para la **gestión**, se designaron roles específicos al comienzo del proyecto que quedan reflejados en el [apartado 2](#).

En la tabla de realización de tareas [\[3.2\]](#), mostramos quién, principalmente, ha realizado las tareas generales, aunque todos hayamos realizado aportaciones a cada tarea. Mientras que en la tabla de realización del trabajo [\[3.3\]](#) mostramos quién ha realizado cada tarea planificada en el diagrama de Gantt, tanto para web como para móvil.

| Tarea                           | Personas responsables                                       |
|---------------------------------|-------------------------------------------------------------|
| Diseño base de datos            | Javier Susín y Ariana Porroche                              |
| Diseño UI                       | Diego Simón, Lucía Vázquez, Irene Pascual y Ariana Porroche |
| Backend (principales consultas) | Irene Pascual, Alex Giménez, Javier Susín y Diego Simón     |
| Backend (otras consultas)       | Javier Susín                                                |
| Despliegue                      | Javier Susín, Diego Simón y Alex Giménez                    |
| Documentación                   | Alex Giménez y Ariana Porroche                              |
| Pruebas                         | Iván Vayad y Ariana Porroche                                |

[3.2]. Tabla de realización de tareas generales.

| Tarea frontend                | Personas que la han completado en web     | Personas que la han completado en móvil  |
|-------------------------------|-------------------------------------------|------------------------------------------|
| Menú libros                   | Lucía Vázquez e Iván Vayad                | Alex Giménez y Ariana Porroche           |
| Detalles libro                | Lucía Vázquez                             | Luna Zhou y Ariana Porroche              |
| Lector                        | Javier Susín y Diego Simón                | Ariana Porroche                          |
| Cuentas y perfiles de usuario | Diego Simón y Lucía Vázquez               | Alex Giménez y Ariana Porroche           |
| Mis listas                    | Iván Vayad                                | Luna Zhou                                |
| Favoritos                     | Lucía Vázquez                             | Luna Zhou                                |
| Listas                        | Iván Vayad                                | Luna Zhou                                |
| Foro                          | Lucía Vázquez                             | Luna Zhou y Ariana Porroche              |
| Estadísticas                  | Irene Pascual y Lucía Vázquez             | Luna Zhou                                |
| Leídos                        | Iván Vayad                                | Luna Zhou y Ariana Porroche              |
| En proceso                    | Iván Vayad                                | Luna Zhou y Ariana Porroche              |
| Claro/Oscuro                  | Irene Pascual, Lucía Vázquez e Iván Vayad | Alex Giménez, Ariana Porroche, Luna Zhou |

[3.3]. Tabla de realización del trabajo.

### 3.3.4. Comunicación interna

La comunicación interna del equipo se realizará mediante un grupo de **Whatsapp** en el que se hablarán mediante **mensajes** los detalles urgentes del proyecto, así como dudas respecto a determinados asuntos concretos. Por otra parte, se realizarán **reuniones** vía **Discord**, las cuales se reservarán para situaciones en las que se debe tratar muchos temas o preguntar por opiniones generales del equipo sobre algún tipo de trabajo ya realizado. Para dichas reuniones se rellenarán documentos con las decisiones tomadas, de forma que a todos los integrantes les quede claro el veredicto final. En las situaciones en las que haya una gran diferencia de ideas, trataremos de llegar a un acuerdo común, exponiendo pros y contras de cada opinión y tratando de llegar a un consenso de equipo que sea el mejor para el proyecto, extrayendo la mayor cantidad de ideas positivas de cada opinión.

### 3.3.5. Determinación de tareas a realizar y reparto

El reparto de las tareas a realizar por cada individuo se realizará de forma correspondiente a los **conocimientos de los miembros del grupo**, de forma que a los miembros que conozcan la forma de implementar una funcionalidad se les asigna dicha funcionalidad para agilizar el trabajo de todo el equipo. Las tareas a realizar corresponden a los diferentes requisitos del

proyecto, y se asocian a cada *Issue* de GitHub, de forma que si alguien comienza la implementación de una funcionalidad, marca dicha *Issue* como “In progress” dentro del proyecto correspondiente y se asigna a sí mismo a la *Issue*, si la finaliza la marca como “Done” y pasa a realizar otra tarea. En caso de que alguien termine una tarea y no tenga nada asignado para hacer, revisará el “Project” de GitHub correspondiente al repositorio en el que está trabajando y seleccionará una *Issue* del apartado “Todo” para comenzar a trabajar en ella.

### **3.3.6. Medidas de progreso y monitorización de estado del proyecto**

**Cada dos semanas** mediremos el progreso realizado por cada repositorio, comprobando el **porcentaje de issues completadas del hito**, así como su correspondencia con el diagrama de Gantt y la evolución del progreso global visible en GitHub Projects. Si observamos que el progreso es muy alto o muy bajo, preguntamos a los integrantes de dicho repositorio la causa de dicho avance. Además, cada dos semanas nos **reunimos** todos los miembros del equipo para **mostrar los avances** logrados en dicho tiempo. Si observamos que un sector del equipo ha disminuido su nivel de progreso y creemos que pueden necesitar ayuda, añadimos alguna persona a dicho grupo de trabajo.

### **3.3.7. Gestión del equipo humano**

Para gestionar posibles diferencias de opinión sobre la ejecución y el enfoque del trabajo, el equipo recurrirá al diálogo y la búsqueda de consensos. En caso de disputas, se priorizará alcanzar soluciones positivas con la participación del resto del equipo, asegurando así colaboración y consenso de la mayoría del equipo. En caso de que no se llegue a un acuerdo, consultaremos al responsable de la dirección del equipo para tomar una decisión definitiva.

## 4. Análisis y diseño del sistema

### 4.1. Análisis de requisitos

#### 4.1.1. Stakeholders

---

**Usuario logueado:** usuario que ha iniciado sesión en el sistema.

**Usuario no logueado:** usuario que utiliza el sistema sin iniciar sesión.

**Usuario:** engloba a los usuarios logueados y usuarios no logueados.

#### 4.1.2. Diccionario de datos

---

- (1) **Modo claro:** interfaz con colores claros o pálidos.
- (2) **Modo oscuro:** interfaz con colores oscuros o grisáceos.
- (3) **Libro:** documento de texto disponible para su visualización y lectura (título, autor, sinopsis, número de palabras, número de páginas, número de horas estimadas de lectura).
- (4) **Página destacada:** página de un libro que se guarda como relevante.
- (5) **Listado de libros:** conjunto de libros del sistema que pueden ser consultados, filtrados y leídos por los usuarios.
- (6) **Lista:** conjunto de libros guardados (nombre, pública o privada, foto de la portada, listado de libros)
- (7) **Libro leído:** libro cuya lectura ha finalizado.
- (8) **Libro en proceso de lectura:** libro cuya lectura ha comenzado pero no ha finalizado.
- (9) **Reseña:** opinión o valoración del usuario sobre un libro (comentario, valoración 1-5).
- (10) **Foro:** espacio del sistema donde los usuarios pueden interaccionar publicando preguntas y respuestas relacionadas con los libros o temas de interés.
- (11) **Estadísticas generales:** datos estadísticos sobre todos los usuarios logueados del sistema: top 3 usuarios con más libros leídos en un mes, top 3 usuarios con más libros leídos en un año y los libros más populares en el mes y año seleccionados.
- (12) **Estadísticas propias:** datos estadísticos de un usuario logueado: número de libros en proceso de lectura, número de libros leídos en el mes actual, número de libros leídos en total, temáticas más frecuentes, libros que más le han gustado al usuario y libros recomendados por el sistema.
- (13) **Favoritos:** lista(6) de libros(3) favoritos preestablecida, que no puede ser eliminada por el usuario, pero funcionará como cualquier otra lista(6).
- (14) **Categorías:** Aventura, Biografía, Ciencia Ficción, Cómic, Drama, Erótico, Fantasía, Filosofía, Histórico, Humor, Infantil, Juvenil, Literatura Clásica, Misterio, Poesía, Romance, Terror y Thriller.
- (15) **Chatbot:** ChatBot con inteligencia artificial que hace recomendaciones de libros según tus gustos y responde a tus preguntas.

### 4.1.3. Requisitos funcionales

---

#### 4.1.3.1. RFs del usuario

---

|        |                                                                                      |
|--------|--------------------------------------------------------------------------------------|
| RF_U1  | El sistema permitirá elegir entre modo claro(1) y modo oscuro(2).                    |
| RF_U2  | El sistema permitirá leer un libro(3). [RNF_U1, RNF_U2]                              |
| RF_U3  | El sistema permitirá hacer zoom en la página de un libro(3). [RNF_U2, RNF_U3]        |
| RF_U4  | El sistema permitirá pasar a la página anterior y siguiente de un libro(3).          |
| RF_U5  | El sistema permitirá visualizar el listado de libros(5).                             |
| RF_U6  | El sistema permitirá la búsqueda de libros(3) en el listado de libros(5) por título. |
| RF_U7  | El sistema permitirá filtrar el listado de libros(5) por categorías(14). [RNF_U5]    |
| RF_U8  | El sistema permitirá ver la información de un libro(3).                              |
| RF_U9  | El sistema permitirá consultar otros libros(3) del mismo autor.                      |
| RF_U10 | El sistema permitirá consultar la valoración media de un libro(3). [RNF_U4]          |
| RF_U11 | El sistema permitirá consultar las reseñas(9) de un libro(3).                        |
| RF_U12 | El sistema permitirá ordenar las reseñas(9) de un libro(3). [RNF_U8]                 |
| RF_U13 | El sistema permitirá consultar las listas(6). [RNF_U6]                               |
| RF_U14 | El sistema permitirá consultar las preguntas del foro(10).                           |
| RF_U15 | El sistema permitirá consultar las respuestas a las preguntas del foro(10).          |
| RF_U16 | El sistema permitirá ordenar las preguntas del foro(10). [RNF_U7]                    |
| RF_U17 | El sistema permitirá consultar las estadísticas generales(11).                       |

[4.1]. Tabla de requisitos funcionales del usuario.

#### 4.1.3.2. RFs del usuario logueado

---

|        |                                                                              |
|--------|------------------------------------------------------------------------------|
| RF_UL1 | El sistema permitirá cerrar sesión.                                          |
| RF_UL2 | El sistema permitirá cambiar la contraseña.                                  |
| RF_UL3 | El sistema permitirá cambiar el nombre de usuario.                           |
| RF_UL4 | El sistema permitirá cambiar la foto de perfil.                              |
| RF_UL5 | El sistema permitirá guardar páginas destacadas(4) de un libro(3). [RNF_UL1] |
| RF_UL6 | El sistema permitirá eliminar páginas destacadas(4) de un libro(3).          |

|         |                                                                                        |
|---------|----------------------------------------------------------------------------------------|
| RF_UL7  | El sistema permitirá consultar las páginas destacadas(4) de un libro(3).               |
| RF_UL8  | El sistema permitirá crear una lista(6).                                               |
| RF_UL9  | El sistema permitirá eliminar una lista(6).                                            |
| RF_UL10 | El sistema permitirá editar una lista(6).                                              |
| RF_UL11 | El sistema permitirá guardar un libro(3) en una lista(6).                              |
| RF_UL12 | El sistema permitirá eliminar un libro(3) de una lista(6).                             |
| RF_UL13 | El sistema permitirá consultar sus propias listas (6).                                 |
| RF_UL14 | El sistema permitirá guardar un libro(3) en favoritos(13).                             |
| RF_UL15 | El sistema permitirá eliminar un libro(3) de favoritos(13).                            |
| RF_UL16 | El sistema permitirá consultar los libros(3) en favoritos(13).                         |
| RF_UL17 | El sistema permitirá consultar los libros leídos(7). [RNF_UL2]                         |
| RF_UL18 | El sistema permitirá consultar los libros en proceso de lectura(8). [RNF_UL3, RNF_UL4] |
| RF_UL19 | El sistema permitirá al usuario añadir una reseña(9).                                  |
| RF_UL20 | El sistema permitirá añadir una pregunta al foro(10).                                  |
| RF_UL21 | El sistema permitirá eliminar una pregunta del foro(10). [RNF_UL6]                     |
| RF_UL22 | El sistema permitirá añadir una respuesta a una pregunta del foro(10).                 |
| RF_UL23 | El sistema permitirá eliminar una respuesta del foro(10). [RNF_UL7]                    |
| RF_UL24 | El sistema permitirá filtrar las preguntas del foro(10). [RNF_UL5]                     |
| RF_UL25 | El sistema permitirá consultar las estadísticas propias(12).                           |
| RF_UL26 | El sistema permitirá hacer preguntas en el Chatbot(15).                                |
| RF_UL27 | El sistema permitirá obtener una respuesta a una pregunta en el Chatbot(15).           |

[4.2]. Tabla de requisitos funcionales del usuario logueado.

#### 4.1.3.3. RFs del usuario no logueado

|         |                                                           |
|---------|-----------------------------------------------------------|
| RF_UNL1 | El sistema permitirá registrarse. [RNF_UNL1]              |
| RF_UNL2 | El sistema permitirá iniciar sesión. [RNF_UNL2, RNF_UNL3] |

[4.3]. Tabla de requisitos funcionales del usuario no logueado.

#### 4.1.4. Requisitos no funcionales

---

##### 4.1.4.1. RNFs del usuario

---

|        |                                                                                                                                                                            |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RNF_U1 | El formato de visualización de los libros será en PDF.                                                                                                                     |
| RNF_U2 | Las páginas del PDF del libro se mostrarán a escala 1.0 por defecto.                                                                                                       |
| RNF_U3 | El zoom aumentará o disminuirá el tamaño de la página del PDF del libro entre 1.0 y 2.0 puntos, aumentando de 0.1 en 0.1.                                                  |
| RNF_U4 | El sistema calculará la valoración media de un libro(3) como la media aritmética de todas las valoraciones del 1 al 5 de las reseñas(9) del libro.                         |
| RNF_U5 | El sistema permitirá la asociación de múltiples categorías(14) a un libro.                                                                                                 |
| RNF_U6 | El sistema sólo permitirá consultar las listas públicas(6) de cualquier usuario. Las listas privadas(6) sólo podrán ser consultadas por los propietarios de dichas listas. |
| RNF_U7 | El sistema permitirá ordenar las preguntas del foro(10) por: más recientes, más antiguas, con más respuestas y con menos respuestas.                                       |
| RNF_U8 | El sistema permitirá ordenar las reseñas de un libro por: más recientes, más antiguas, valoraciones más altas y valoraciones más bajas.                                    |

[4.4]. Tabla de requisitos no funcionales del usuario.

##### 4.1.4.2. RNFs del usuario logueado

---

|         |                                                                                                              |
|---------|--------------------------------------------------------------------------------------------------------------|
| RNF_UL1 | El sistema guardará el número de página cuando el usuario se guarde una página destacada(4).                 |
| RNF_UL2 | El sistema guardará los libros leídos(7).                                                                    |
| RNF_UL3 | El sistema guardará los libros en proceso de lectura(8).                                                     |
| RNF_UL4 | El sistema guarda y sincroniza el progreso de lectura de un usuario logueado entre dispositivos web y móvil. |
| RNF_UL5 | El filtrado de las preguntas del foro(10) será por las preguntas añadidas por el propio usuario logueado.    |
| RNF_UL6 | El sistema permitirá eliminar sólo las preguntas publicadas por el propio usuario en el foro(10).            |
| RNF_UL7 | El sistema permitirá eliminar sólo las respuestas publicadas por el propio usuario en el foro(10).           |

[4.5]. Tabla de requisitos no funcionales del usuario logueado.

#### 4.1.4.3. RNFs del usuario no logueado

---

|          |                                                                                           |
|----------|-------------------------------------------------------------------------------------------|
| RNF_UNL1 | El registro se realizará mediante Google o correo y contraseña.                           |
| RNF_UNL2 | El inicio de sesión se realizará mediante Google o correo y contraseña.                   |
| RNF_UNL3 | El sistema comprobará que el usuario tenga una cuenta registrada antes de iniciar sesión. |

[4.6]. Tabla de requisitos no funcionales del usuario no logueado.

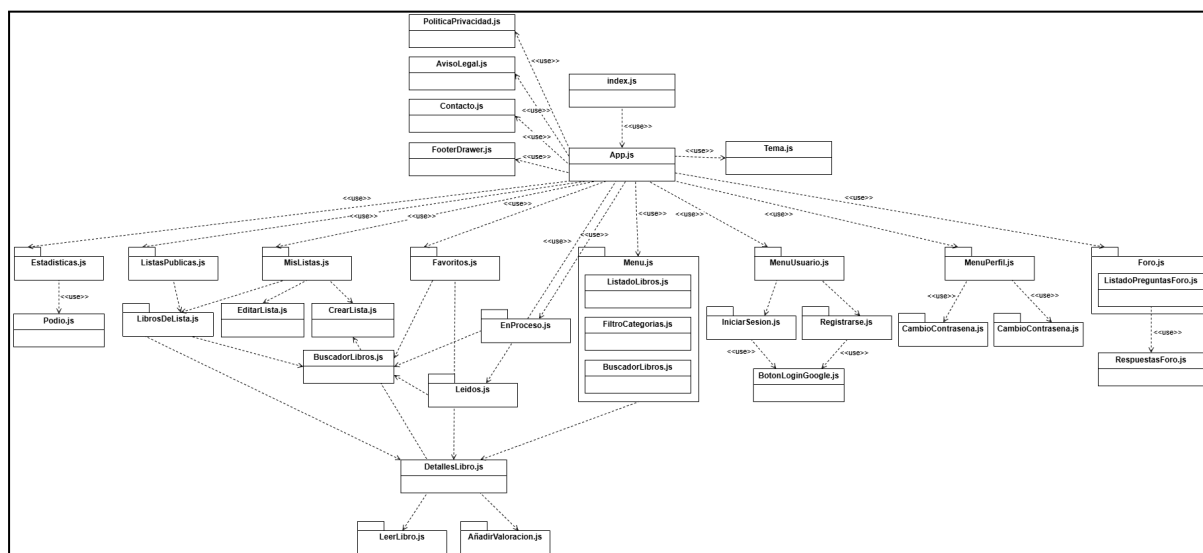
## 4.2. Diseño del sistema

### 4.2.1. Vistas de módulos

#### 4.2.1.1. Vista de módulos del frontend móvil

##### DESCRIPCIÓN DE LOS MÓDULOS

La vista de módulos del frontend móvil [4.7] comienza con el módulo *index.js* que registra el componente *App* del módulo *App.js*. Este componente configura la navegación principal de la app, definiendo el menú lateral y las pilas de navegación. Luego ya están los distintos módulos de la aplicación: *Menú*, *MenuUsuario*, *MenuPerfil*, *Foro*, *Favoritos*, *Leidos*, *EnProceso*, *MisListas*, *ListasPublicas* y *Estadisticas*, que se componen de sus propios módulos y/o utilizan otros. El módulo *DetallesLibro* es invocado desde varios módulos, y a su vez invoca a *LeerLibro*, que es el módulo del lector de libros.



[4.7]. Vista de módulos del cliente móvil.

##### DECISIONES DE TECNOLOGÍAS

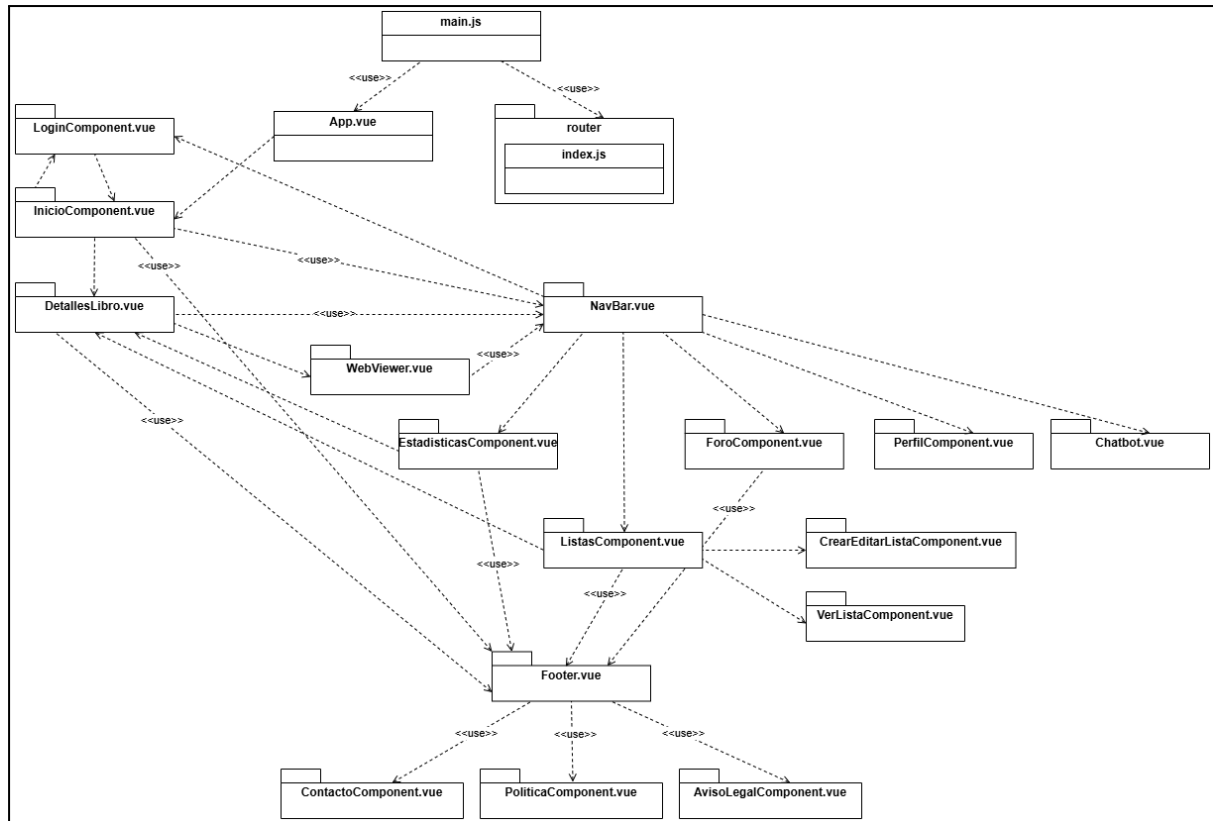
Para el **front-end móvil**, valoramos utilizar tecnologías como **Flutter**, **React Native** y **Java con Android Studio**. Descartamos Flutter ya que ningún miembro del equipo había trabajado con esta tecnología y queríamos reducir al máximo la curva de aprendizaje. También desechamos la opción de usar Java con Android Studio debido al desarrollo más complejo y específico para Android. Finalmente, elegimos **React Native** ya que parte del equipo había trabajado con Next.js y React previamente, y las tecnologías son similares, por lo que podríamos reducir la curva de aprendizaje.

#### 4.2.1.2. Vista de módulos del frontend web

##### DESCRIPCIÓN DE LOS MÓDULOS

La vista de módulos del frontend web [4.8] comienza con el módulo *main.js* que crea el componente *App* del módulo *App.vue*. Este invoca al componente *index.js* que establece las rutas entre las componentes y el *path*. La mayoría de pantallas incluyen la barra de

navegación *NavBar.vue* y el *Footer.vue*, el cual se compone del *AvisoLegalComponent.vue*, *PoliticaComponent.vue* y *ContactoComponent.vue*. Cuando entramos a la aplicación, visualizamos *LoginComponent.vue*, mediante el cual iremos a *InicioComponent.vue*. Desde la barra de navegación *NavBar.js* podemos ir a cualquiera de los módulos *EstadisticasComponent.vue*, *ListasComponent.vue*, *PerfilComponent.vue*, *ForoComponent.vue* y *Chatbot.vue*. El módulo *DetallesLibro.vue* puede ser invocado desde varios módulos, y este es el que utiliza el módulo del lector de libros *WebViewer.vue*.



[4.8]. Vista de módulos del cliente web.

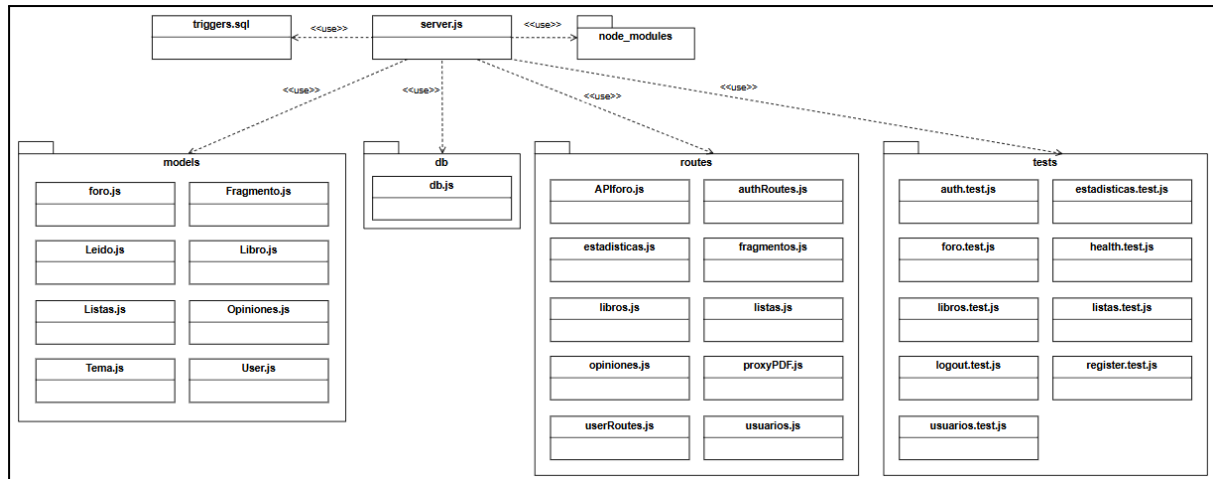
## DECISIONES DE TECNOLOGÍAS

Para el front-end web, consideramos utilizar **Vue.js**, **Next.js** o directamente con HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) y JavaScript **sin un framework**. Descartamos Next.js ya que se basa en React, pero ya utilizamos React Native para la aplicación móvil y no podíamos repetir tecnologías. Optamos por **Vue.js**, ya que proporciona una estructura más organizada, facilita la reutilización de componentes y permite una mayor escalabilidad. Además, su integración con **Bootstrap** nos ayuda a mantener un diseño responsive sin necesidad de escribir estilos desde cero. Descartamos la opción de desarrollo sin framework porque aumentaría la complejidad del mantenimiento y haría más difícil la implementación de funcionalidades dinámicas.

### 4.2.1.3. Vista de módulos del backend

## DESCRIPCIÓN DE LOS MÓDULOS

La vista de módulos del backend [4.9] tiene como punto de entrada el módulo `server.js`, que arranca el servidor y gestiona toda la configuración del backend. El módulo `db.js` configura la conexión con la base de datos PostgreSQL y crea un pool de conexiones. Los modelos incluidos en `models` definen la estructura de las tablas de la base de datos y proporcionan métodos para interactuar con ellas. Los módulos incluidos en `routes` definen las rutas de la API, gestionando las solicitudes HTTP y conectando con la lógica del servidor y la base de datos.



[4.9]. Vista de módulos del backend

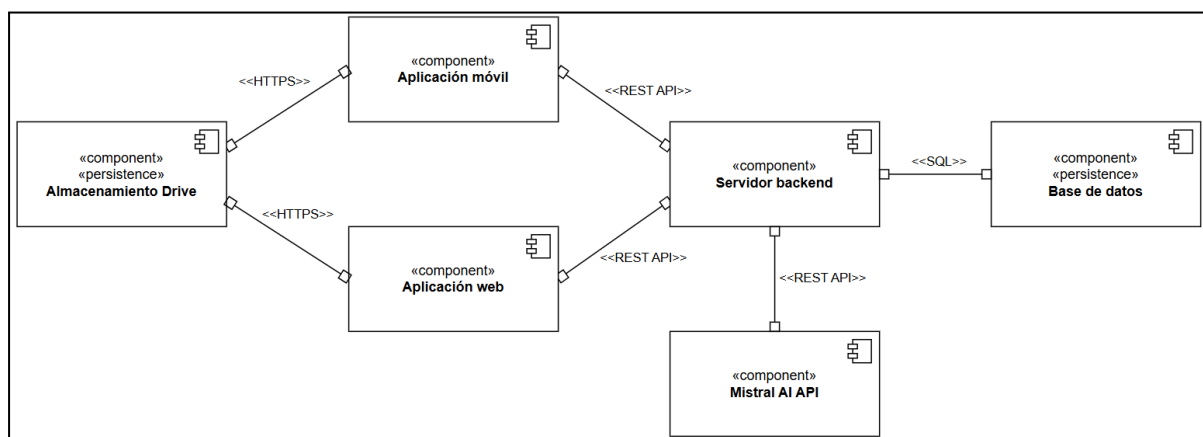
## DECISIONES DE TECNOLOGÍAS

Respecto al **backend**, decidimos utilizar un lenguaje basado en **JavaScript** para mantener coherencia con el frontend y facilitar la integración. Se evaluaron opciones como **Node.js** y **Deno**, pero finalmente escogimos **Node.js** debido a su gran comunidad de soporte y a que la mayoría del equipo ya lo había utilizado previamente, lo que reduce la curva de aprendizaje. Para la implementación de la API, optamos por **Express**, ya que permite gestionar peticiones REST (*Representational State Transfer*) de manera sencilla.

### 4.2.2. Vista de componentes y conectores

#### DESCRIPCIÓN DE LOS COMPONENTES

En la vista de componentes y conectores [4.10] tenemos 2 componentes principales: la *Aplicación móvil* y la *Aplicación web*. Estas contienen tanto la interfaz de usuario como la lógica de la aplicación y se comunican con el *Servidor backend* mediante una *API REST*. Se encarga de gestionar las peticiones y distribuir datos a los clientes. Se conecta con la *Base de datos* mediante *SQL*. La *Base de datos* almacena la información principal de la aplicación, a excepción de los libros y algunas imágenes. Los libros, las imágenes del perfil y de las listas se guardan en Google Drive, lo cual está representado por *Almacenamiento Drive*. Tanto la aplicación móvil como la web acceden a estos recursos mediante *HTTPS*, para establecer una conexión segura.



[4.10]. Vista de componentes y conectores.

## DECISIONES DE TECNOLOGÍAS

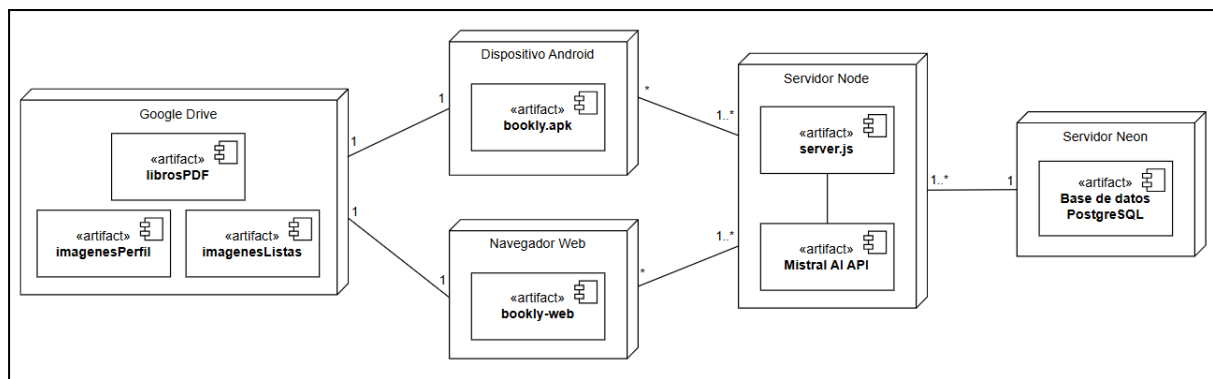
La **API** seguirá un diseño **RESTful**, garantizando una clara separación entre cliente y servidor, el uso de métodos HTTP (*HyperText Transfer Protocol*) estándar y respuestas en formato JSON (*JavaScript Object Notation*). La mayoría de las operaciones serán **asíncronas**, lo que permitirá gestionar múltiples solicitudes sin bloquear el servidor. Para el almacenamiento de datos, se utilizará **PostgreSQL**, un sistema SQL (*Structured Query Language*) robusto y ampliamente conocido por el equipo, lo que facilitará la gestión de datos relacionales.

Para la implementación de **ChatBot**, usamos la inteligencia artificial **Mistral LeChat**, con la versión open-mixtral-8x22b, la cual se ha implementado mediante el uso de una API [4]. Hemos hecho uso de la API de tal forma que conseguimos que sea transparente al usuario, es decir, en ningún momento saben que es una IA y se siente como un chatbot real. Además gracias al prompt que hemos introducido para su funcionamiento conseguimos una experiencia personalizada del usuario. Consideramos crear de 0 un chatbot entrenándolo con datos, mediante **Dialogflow**, o usar la API de ChatGPT, pero se descartaron por malos resultados, y por la necesidad de tarjeta de crédito respectivamente. Finalmente se usó Mistral ya que es la IA más veloz y nos permite hacer uso de una API de forma gratuita.

### 4.2.3. Vista de despliegue

#### DESCRIPCIÓN DE LOS NODOS

En la vista de despliegue [4.11] asignamos los componentes software al hardware donde se ejecutarán. Para la aplicación móvil, tenemos un nodo *Dispositivo Android*, donde se ejecuta *bookly.apk*. Para la aplicación web, en este caso el nodo *Navegador Web* es el que ejecuta *bookly-web*. Ambos nodos se conectan con el *Servidor Node* del backend, que ejecuta *server.js*. Ya por último, el servidor del backend se conecta con el *Servidor Neon*, que contiene la *Base de datos PostgreSQL*. Cabe destacar que tenemos un nodo *Google Drive*, que almacena los *librosPDF*, las *imagenesPerfil* y las *imagenesListas*, al cual se conectan ambas aplicaciones.



[4.11]. Vista de despliegue.

#### DECISIONES DE TECNOLOGÍAS

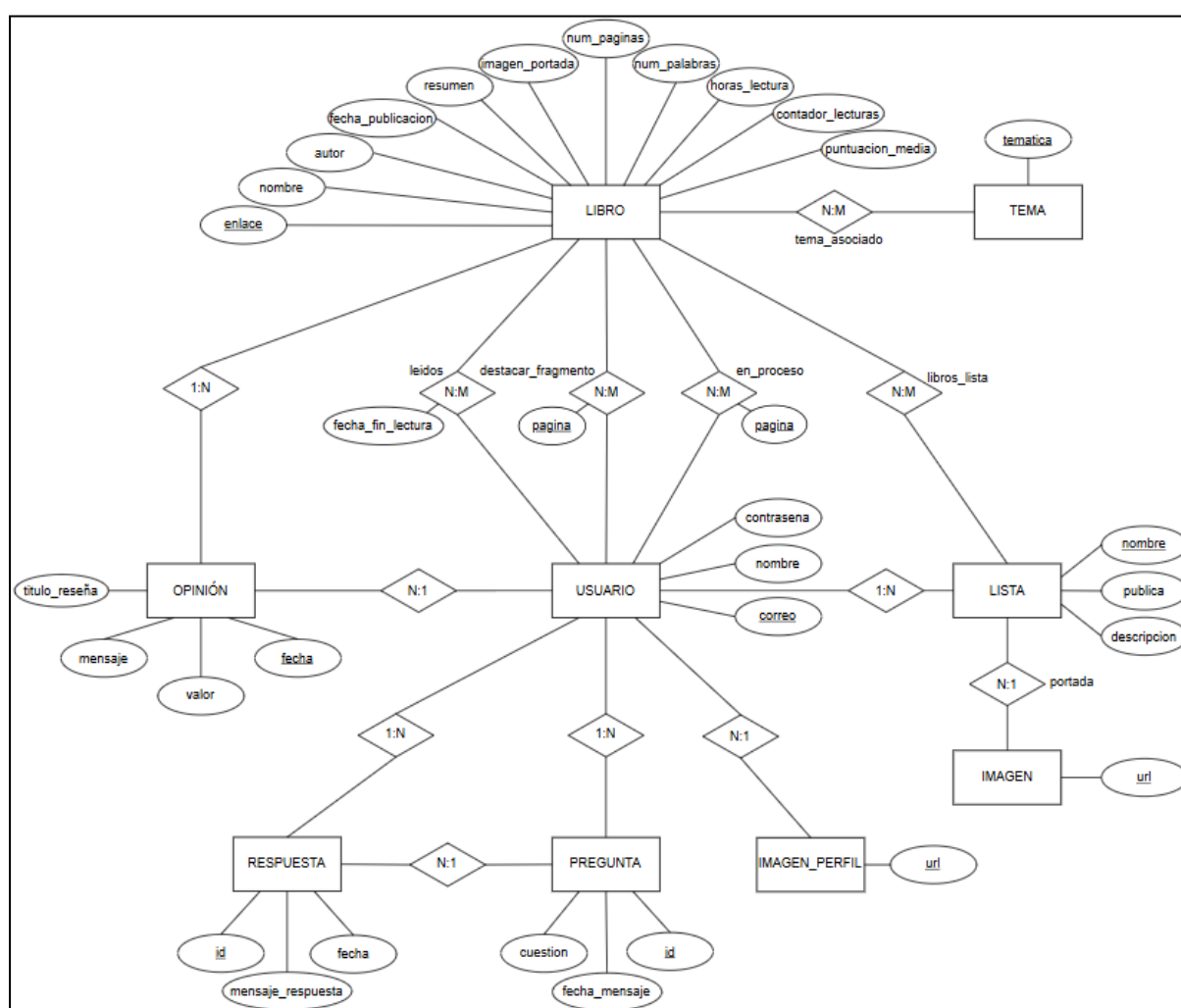
Para el **despliegue**, consideramos opciones como **Vercel**, **Render** y **Netlify**. Optamos por **Render**, ya que permite desplegar tanto el backend como la base de datos en la misma plataforma, simplificando la configuración y la administración del proyecto. Descartamos Vercel porque está más orientado a aplicaciones front-end estáticas, y Netlify, ya que su enfoque principal es el despliegue de sitios web, sin tantas facilidades para el backend y la base de datos. Además algunos miembros del equipo ya conocen Render de proyectos anteriores, lo que facilitará el despliegue.

En cuanto a la **base de datos**, consideramos opciones como **Render** y **MongoDB Atlas**, pero finalmente elegimos **Neon** por su mejor rendimiento en **cargas dinámicas** y su arquitectura optimizada que separa almacenamiento y cómputo, permitiendo **autoescalado** según la demanda, además de que parte del equipo ya había trabajado con Neon.

## 4.2.4. Diseño de la base de datos

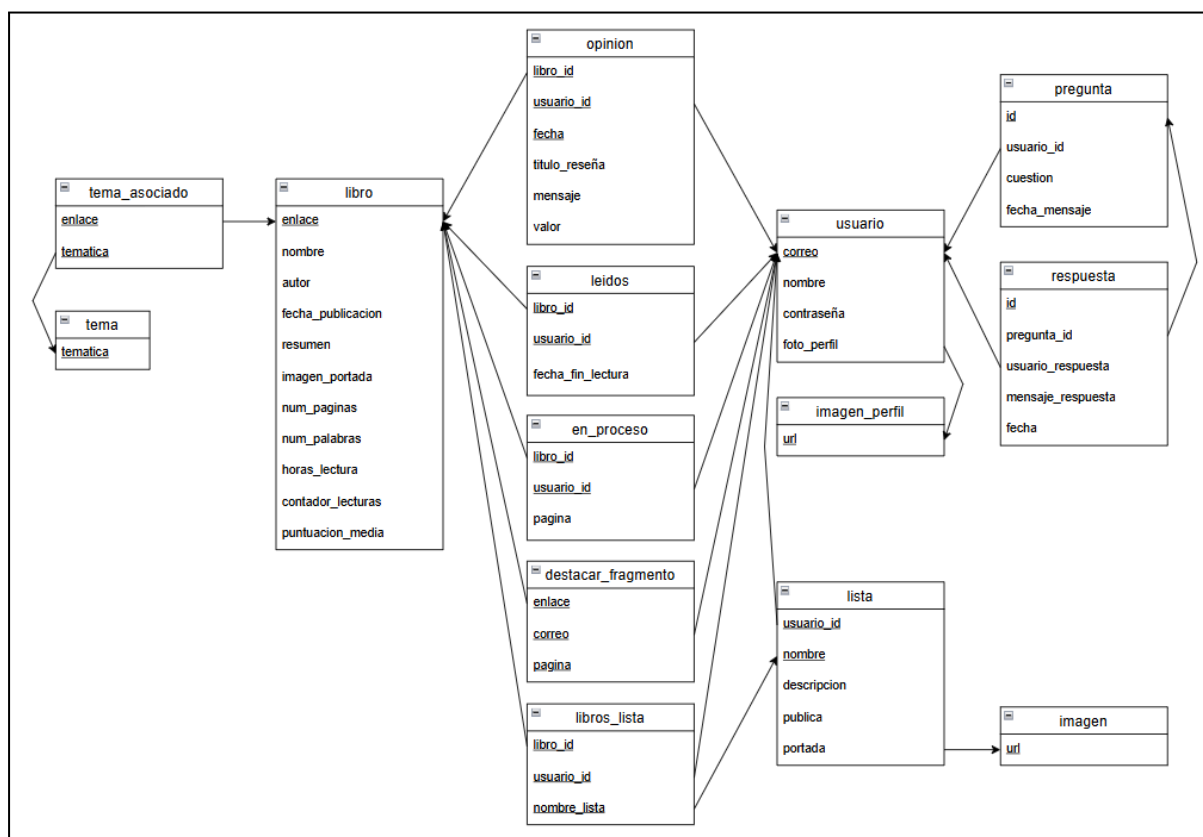
### DESCRIPCIÓN DE LA BASE DE DATOS

El modelo entidad-relación de la base de datos [4.12] muestra cómo se estructura la información de nuestra aplicación de lectura de libros. Los *Libros* guardan información sobre el autor, el contenido, la portada, etc, y están asociados a un *Tema* o a varios. Los *Usuarios* tienen un nombre, un correo y una contraseña que los identifica. Estos pueden poner una *Opinión*, incluyendo un mensaje y una valoración del 1 al 5. También pueden leer libros, guardar páginas destacadas y mantener el progreso de lectura de los libros. Además, pueden crear *Listas* de libros públicas o privadas, e interactuar así con otros usuarios. Con respecto a la interacción, existe un *Foro* donde los usuarios pueden preguntar inquietudes, hablar con otros usuarios o sugerir mejoras, y responder a dichas preguntas.



[4.12]. Modelo entidad-relación de la base de datos.

El modelo relacional [4.13] representa en forma de tablas las entidades descritas anteriormente y sus relaciones. Estas tablas son las que usaremos para crear las tablas de la base de datos.



[4.13]. Modelo relacional de la base de datos.

## DECISIONES DE TECNOLOGÍAS

Hemos elegido un **modelo relacional** en lugar de una base de datos NoSQL como **MongoDB** debido a la naturaleza estructurada y relacional de nuestros datos. Las bases de datos relacionales permiten manejar eficientemente estas relaciones mediante claves foráneas y joins. Además, ofrecen integridad referencial y la capacidad de realizar consultas complejas de manera eficiente. También garantizan la consistencia y transacciones ACID, esenciales para operaciones críticas como las opiniones de los usuarios y la gestión de listas de libros. MongoDB, no es la mejor opción para este tipo de sistema.

## 5. Memoria del proyecto

### 5.1. Inicio del proyecto

El inicio del proyecto se desarrolló según lo planificado en la [sección 3.1.1](#). Se realizaron las configuraciones necesarias de los proveedores de servicios, como se había previsto. A continuación, detallamos las acciones realizadas:

- **Google Drive:** Creamos una carpeta compartida en Google Drive para almacenar todos los libros, imágenes y documentación del proyecto. Organizamos todo el contenido en carpetas adecuadamente para mantener el acceso y la gestión de los documentos de manera eficiente.
- **Neon:** Todos los miembros nos registramos en Neón, creamos una base de datos y se concedió acceso por invitación a todos los miembros del grupo.
- **Figma:** Todos los miembros nos registramos en Figma, creamos un proyecto para la interfaz de web y otro para la interfaz móvil. Invitamos a todos los miembros a ambos proyectos para poder realizar aportaciones y coger ideas de uno y de otro.
- **Render:** Los encargados del despliegue se registraron en Render, realizaron la configuración del servidor y se realizaron los cambios necesarios en las rutas del proyecto para poder utilizar el servidor, y acceder mediante el mismo a la base de datos.

En cuanto a las pruebas de la aplicación móvil, se utilizó una variedad de dispositivos móviles propios para realizar las primeras pruebas. Estas pruebas se realizaron bajo la premisa de que la aplicación debe ser compatible con Android 5.0 o superior. Para las pruebas sencillas como el correcto funcionamiento de un requisito insertado, utilizamos el **emulador** proporcionado por Android Studio junto a Expo, entre los que se encuentran "Medium Phone API 35" y "Small Phone API 35", los cuales utilizan API 35 correspondiente a la versión **Android 12**. Mientras que para pruebas reales en **dispositivos físicos** utilizamos "Redmi Note 9S" con "MIUI Global 13.0.2" y **Android 12**, "Samsung Galaxy A34 5G" y "Samsung A52S 5G" con "One UI 6.1" y **Android 14**.

La **formación** de los miembros del equipo fue exitosa, la documentación creada por los propios miembros del equipo para comprender el uso de GitHub e implementarlo en el trabajo cumplió su papel con creces, así como la formación en entornos desconocidos como React Native o Vue.js, formación llevada a cabo con tutoriales externos que ayudó al aprendizaje de dicha tecnología.

## 5.2. Ejecución y control del proyecto

La ejecución y control del proyecto han seguido los planes establecidos en las **secciones 3.2 y 3.3**, de forma que se ha seguido un desarrollo del proyecto ordenado, así como una buena gestión del equipo de trabajo y del reparto de tareas.

### 5.2.1. Funcionamiento del reparto de trabajo

El reparto de trabajo siguió la planificación establecida, con cada miembro encargado de partes específicas del proyecto según sus conocimientos y responsabilidades. Se **asignaron tareas** a través de GitHub *Issues*, donde cada desarrollador seleccionaba una tarea y la marcaba como "In Progress" hasta su finalización, marcándola como "Done". Además, si una parte del proyecto quedaba muy avanzada respecto a las demás, como es el caso de backend, se realiza una **reestructuración** del grupo de trabajo, reduciendo la cantidad de personal dedicada a dicha sección y redistribuyéndolos en el resto de secciones, prestando especial atención a las más atrasadas y pensando en las nuevas áreas de trabajo como documentación, despliegue y presentación, siguiendo los pasos descritos en la **sección 3.3.3**. Este sistema de reparto nos permitió adaptarnos a los ritmos variables de cada área sin comprometer la entrega.

### 5.2.2. Funcionamiento de la comunicación interna

La comunicación interna ha sido fluida durante todo el proyecto, siguiendo la idea principal de la **sección 3.3.4** a través de **WhatsApp** para temas **urgentes, temas de discusión rápida o dudas** y **Discord** para **reuniones y discusiones**, al inicio del proyecto, como es el caso de la discusión inicial sobre cómo afrontar el trabajo desde el comienzo del mismo, las tecnologías a utilizar, para las cuales se tuvo que hacer un resumen de las mejores opciones posibles y en cuales teníamos experiencia. Además, **documentamos** todas las decisiones clave en Google Docs para que el acuerdo alcanzado quede por escrito. También se han realizado reuniones presenciales periódicas cada 2 semanas, en las que se ha hablado sobre los avances realizados entre reuniones en cada área, actualizando así al resto del equipo sobre el estado del trabajo, encargando pequeñas tareas al resto del equipo y realizando fracciones del trabajo en pequeños grupos.

### 5.2.3. Funcionamiento de la medición del progreso y el conocimiento del avance

Como se describió en la **sección 3.3.6** el progreso del proyecto se midió con éxito mediante **GitHub**, mediante el uso de **Issues y Projects**, donde hemos podido observar el trabajo realizado, el trabajo que se está realizando y el trabajo pendiente de realizar en cada momento del proyecto en las áreas de backend y frontend. Para llevar un seguimiento del progreso, además de utilizar GitHub como hemos mencionado previamente, realizamos **reuniones y revisiones bisemanales** para dar a conocer a los demás miembros el trabajo realizado y el que está en proceso, así como una supervisión del avance, de forma que los

demás miembros tengan una prueba verídica del avance realizado y puedan dar una retroalimentación del mismo para saber si se deben realizar cambios en el trabajo o en la forma de trabajo.

En otros aspectos en los que no se utiliza GitHub *Issues*, como para la documentación, el despliegue, o la presentación se ha utilizado Whatsapp para comunicar los avances realizados, así como las reuniones presenciales para mostrar todo el avance realizado con pruebas del mismo.

#### **5.2.4. Ajustes al calendario inicial**

Tuvimos que realizar algunos **ajustes al calendario inicial**, en parte debido a que el sector de backend avanzó mucho desde el inicio, incluso más de lo esperado, lo que significó que pudimos liberar a 2 personas del sector de backend desde muy temprano y redirigir a cada uno a las áreas de web y móvil y comenzar con nuevas áreas del proyecto como el despliegue y la documentación, tal y como se especifica en las tablas [\[2.1\]](#) y [\[2.2\]](#) de la [sección 2](#) de organización del proyecto. Los 2 miembros que se quedaron en el backend, también se centraron en descubrir formas de implementar la lectura de PDFs de la aplicación web, mientras resolvían consultas que iban surgiendo y realizaban el despliegue.

A partir de este punto, nos centramos en avanzar lo máximo posible en todos los aspectos y tratando de alcanzar los objetivos de la entrega intermedia lo antes posible, tratando de maximizar la eficiencia del periodo en el que disponíamos de más tiempo de trabajo, siguiendo las estrategias de prevención de riesgos. Lo siguiente que se decidió completar fue el despliegue total de la aplicación, tanto del backend como del frontend web y móvil (APK). Y finalmente desarrollar las pruebas del sistema así como la presentación y demostración del mismo.

#### **5.2.5. Adecuación de las herramientas y tecnologías empleadas**

Las herramientas han sido una correcta elección, ya que no hemos realizado ningún cambio en las mismas y han funcionado como se esperaba, aunque eso no quiere decir que no hemos tenido problemas. Uno de los problemas principales ha sido a la hora de imprimir los PDFs de los libros, ya que el lector de PDFs de Google Drive nos daba bastantes problemas, por lo que se decidió investigar sobre otras opciones, como son DropBox o OneDrive, pero tenían problemas similares o incluso peores. Otra herramienta que dió algunos problemas fue Render al tratar de conectar la base de datos con el sistema en web y móvil para realizar el despliegue en servidor de la base de datos y del sistema para web y para Android únicamente la base de datos, aunque no fue nada que no se pudiera resolver, por lo que no se pensó en sustituir esta herramienta.

Por otra parte, las tecnologías utilizadas respondieron bien a nuestras necesidades, aunque hubo problemas que tuvimos que solucionar. En el caso de React Native los problemas principales vinieron a raíz de que era la primera vez para todos que utilizamos esta tecnología, lo que incrementó la curva de trabajo para el área de móvil, y a pesar de ser una tecnología no muy compleja de utilizar, se debió aprender correctamente el funcionamiento

de la misma. Vue por su parte, no dió tantos problemas, ya que se basa en HTML y CSS, tecnologías que ya hemos utilizado, aunque la mayor complejidad fue adaptarse a su estructura y comprenderla. Para ambas tecnologías hay que destacar la lectura de PDFs, lo que ha llevado una gran cantidad de horas de investigación mediante prueba y error sobre cómo implementar a nuestro gusto, trabajando con diferentes bibliotecas hasta encontrar la adecuada. El resto de tecnologías no dieron pie a problemas.

El uso de Mistral para Chatbot fue esencial, ya que como hemos comentado, se probó a entrenar una IA mediante datos de entrenamiento, pero los resultados fueron muy pobres. Pero Mistral nos dió resultados desde el minuto 0, así que destacamos muy positivamente su elección.

### **5.2.6. Funcionamiento de procesos de control del código, construcción y despliegue**

El **control de las versiones del código** ha sido bastante sencillo, ya que al utilizar GitHub y su funcionalidad de ramas, hemos podido gestionar cada uno su propio trabajo en paralelo al resto sin molestarnos unos con otros. La decisión principal ha sido la de mantener una rama principal main, la cual debe estar siempre activa, de forma que el contenido de dicha rama esté lista para ser ejecutada en cualquier momento. En caso de observar errores, se crea una nueva rama y se realiza un merge cuando se corrigen errores del main. Esta es la misma funcionalidad que para añadir funcionalidades, creando una rama nueva, realizando el trabajo ahí y realizando un merge con la rama main cuando se finalice la funcionalidad y se compruebe que no tiene errores. Para **las versiones de construcción** hemos documentado las dependencias a añadir, poniéndolas en común de esta forma y realizando las instalaciones necesarias mediante "npm", además de haber añadido la construcción automática por medio de GitHub Actions. Las **versiones de despliegue** se han gestionado de una forma más simple que el código, ya que al haber uno o dos encargados para el despliegue hemos decidido que trabajen juntos, realizando llamadas por discord o presenciales, de forma que haya 2 personas pendientes del mismo proceso para prevenir errores y sumar ideas entre los 2.

### **5.2.7. Pruebas del software**

Durante el desarrollo del software se han realizado pequeñas pruebas, de forma que al implementar funcionalidades y pantallas cada uno era responsable de verificar que la funcionalidad se implementa de forma correcta, y no perjudica ningún otro apartado del sistema.

Por otra parte, y ya ubicados en la parte final del proyecto, hemos realizado diferentes tipos de pruebas, siguiendo las estipuladas en el [apartado 3.2.9](#), como es el caso de la revisión por pares, de forma que personal del resto de equipos de trabajo verifican la correcta implementación de funcionalidades, y el reflejo del diseño inicial, hemos seguido un checklist de cobertura funcional, así como de pantallas, navegación y la similitud entre web y móvil. También hemos realizado una revisión general de documentos, como es el caso de

este mismo documento, o de diagramas, y de la correcta descripción y comprensión de los requisitos. Por último se han realizado pruebas manuales, en las que se han seguido escenarios de uso real, de forma que revisemos posibles problemas de accesibilidad y revisar el correcto funcionamiento en dispositivos de diferente tamaño.

## 5.3. Cierre del proyecto

### 5.3.1. Comparación estimaciones iniciales con resultados finales

Al comienzo del proyecto realizamos unas estimaciones sobre los posibles esfuerzos y tamaño del proyecto, y estas son las comparaciones con los resultados finales. En primer lugar se muestra una tabla con las horas de inversión esperadas y otra con las horas invertidas reales, y finalmente se muestra una tabla con las líneas de código esperadas por módulo y otra tabla con las líneas de código finales (calculadas a partir del comando “git fame” en los repositorios de GitHub de la organización).

| Diseño | Backend | Frontend-web | Frontend-móvil | Documentación | Pruebas | TOTAL |
|--------|---------|--------------|----------------|---------------|---------|-------|
| 50     | 125     | 200          | 225            | 50            | 50      | 800   |

[5.1]. Tabla de horas esperadas por área.

| Miembro      | Diseño | Backend | Web | Móvil | Documentación | Pruebas | TOTAL |
|--------------|--------|---------|-----|-------|---------------|---------|-------|
| Luna         | 0      | 0       | 0   | 91,5  | 2,5           | 0       | 94    |
| Lucía        | 4      | 0       | 83  | 0     | 5,5           | 1       | 93,5  |
| Irene        | 4,5    | 22      | 51  | 0     | 10,5          | 2       | 90    |
| Iván         | 1      | 1       | 81  | 0     | 6             | 1       | 90    |
| Alex         | 2      | 14      | 0   | 18    | 65            | 0       | 99    |
| Javier       | 5      | 66      | 20  | 0     | 4             | 0       | 95    |
| Diego        | 2      | 41      | 38  | 0     | 8             | 0       | 89    |
| Ariana       | 7      | 0       | 0   | 66,5  | 20,5          | 6       | 100   |
| <b>TOTAL</b> | 25,5   | 144     | 273 | 176   | 122           | 10      | 750,5 |

[5.2]. Tabla de horas reales.

| Backend | Frontend-web | Frontend-movil | TOTAL  |
|---------|--------------|----------------|--------|
| 5.000   | 5.000        | 5.000          | 15.000 |

[5.3]. Tabla de líneas de código esperadas por área.

| Backend | Frontend-web | Frontend-movil | TOTAL  |
|---------|--------------|----------------|--------|
| 6.574   | 8.030        | 8.458          | 23.062 |

[5.4]. Tabla de líneas de código reales.

Como podemos observar, el tiempo dedicado real ha sido bastante cercano al tiempo esperado, lo que significa que hemos trabajado dedicando un tiempo correcto cada uno de nosotros. Por otra parte, la estimación de líneas de código se ha alejado bastante del número real de líneas, esto se debe principalmente a la falta de experiencia de algunas de las tecnologías utilizadas y a la poca experiencia en proyectos tan grandes y con tantos miembros.

### 5.3.2. Riesgos presentes

En la [Sección 3.1.2](#) se identificaron posibles riesgos que el equipo podía encontrarse a lo largo del proyecto y sus posibles estrategias de mitigación. Algunos de estos riesgos **se han presentado** durante el proceso, como es el caso de que los integrantes tuvieran **mucha carga de trabajo**, para lo cual seguimos la estrategia de adelantar la mayor cantidad de trabajo para la época en la que se dispusiese de mayor tiempo libre y distribuir el trabajo, lo cual funcionó de manera excelente. También nos encontramos con los riesgos de **no haber trabajado con algunos compañeros** antes, así como trabajar en un **grupo tan grande**, para lo cual seguimos al pie de la letra las estrategias de mitigación, organizando reuniones y dando a conocer al resto nuestros conocimientos y dividiendo en tareas el trabajo, obteniendo también éxito en la mitigación de estos riesgos. Por otra parte, hemos afrontado los riesgos de **no tener conocimiento del cliente** y de **utilizar tecnologías no conocidas**, de forma que al igual que en los casos anteriores las hemos podido solucionar siguiendo las estrategias de mitigación fijadas, realizando entrevistas con el cliente y realizando formaciones iniciales para los miembros del equipo que no conocían dichas tecnologías.

Cabe destacar que estos riesgos han sido los principales mencionados en la [Sección 3.1.2](#), el resto de riesgos presentados en el [Apéndice 1](#) también han sido enfrentados y mitigados con éxito siguiendo las estrategias fijadas (ver [Al.2](#)). Fuera de estos riesgos mencionados no hemos detectado ningún otro al que hayamos tenido que enfrentarnos ni mitigar de ningún modo.

En futuros proyectos, consideraríamos estos mismos riesgos como relevantes y mantendremos las estrategias aplicadas, ya que han demostrado ser efectivas.

### 5.3.3. Lecciones aprendidas sobre herramientas y tecnologías

En la [Sección 3.2](#) mencionamos las diferentes **herramientas** utilizadas en el proyecto, así como el uso que se les va a dar. Algunas de estas herramientas eran novedosas para algunos miembros del equipo, lo que nos ha llevado a aprender cómo usarlas y qué posibilidades ofrecen. **GitHub** ha sido una de las herramientas más importantes del proyecto, ya que gracias a ella hemos podido compartir el código de trabajo y repartir las

tareas fácilmente. Ha sido un aprendizaje muy valioso ya que es una herramienta muy útil, y aunque no sepamos utilizar todas sus funciones, ya conocemos las principales, como son las ramas para trabajar el código sin afectar a los demás, las *Issues* y los *Projects* para distribuir el trabajo, así como los *Insights* para ver la cantidad de código aportado. También hemos aprendido a cómo utilizar **VSCode** en conjunto con GitHub, asociando un directorio a un repositorio por medio de la terminal. Para la inspección del software de móvil utilizamos **Expo Go**, para el cual hemos aprendido cómo configurarlo con un emulador de Android en el ordenador y cómo enlazarlo con el proyecto para probar el código en un smartphone.

Por otra parte, en la [Sección 4.2](#) hablamos sobre las diferentes **tecnologías** utilizadas, de las cuales destacamos el aprendizaje de **Vue.js**, en la que hemos aprendido a cómo utilizar dicho framework como estructura inicial de organización para el proyecto web, también hemos progresado con **React Native**, con el cual hemos aprendido a crear un proyecto Android sin la necesidad del uso de Android Studio, así como su organización y funcionamiento. También nos llevamos nuevos conocimientos de **Javascript**, ya que algunos miembros del equipo lo habían utilizado de forma bastante reducida previamente, y han desarrollado nuevas virtudes con dicho lenguaje. Lo mismo ocurre con **Node, Express y Render**, aunque Express y Render ha sido una novedad para casi todos los miembros del equipo, de forma que se ha ampliado el conocimiento de las APIs y su uso mediante Express y el despliegue del proyecto tanto aplicación como base de datos en Render.

### 5.3.4. Recopilación de esfuerzos

| Área           | Miembro         | Funciones                                                                                                                                                                                                                        |
|----------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Backend        | Javier Susín    | Configuración archivos conexión base de datos, y del servidor para conectar backend con frontend, participación en foros, fragmentos, opiniones, libro, proxy para lector. Administración del despliegue.                        |
|                | Diego Simón     | Configuración archivos conexión base de datos, y del servidor para conectar backend con frontend, gestión de usuarios (inicio/registro), participación en libros, tema.                                                          |
|                | Irene Pascual   | Gestión de listas y favoritos en backend, integración de estadísticas en backend y frontend, post de valoración, triggers automáticos, mejoras dinámicas en el frontend (botones, emoticonos, flechas de desplazamiento...).     |
|                | Alex Giménez    | Temáticas y libros, gestión páginas destacadas, libros leídos, publicar preguntas, cambiar contraseña                                                                                                                            |
| Frontend web   | Irene Pascual   | Página de estadísticas y en proceso de retoques estéticos de página de inicio.                                                                                                                                                   |
|                | Lucía Vázquez   | Aplicar y mantener estado de modo claro/oscuro en pantallas, detalles libros, inicio de sesión con usuario y contraseña, foro, estadísticas generales, mis favoritos, perfil y cambios estéticos para seguir la línea del móvil. |
|                | Iván Vayad      | Visualización, creación, edición y eliminación de listas (en proceso, leído, otros...), inicio, modo claro/oscuro                                                                                                                |
| Frontend móvil | Luna Zhou       | Gestión de listas, libros favoritos, leídos y en proceso, foro, estadísticas generales y del propio usuario logueado.                                                                                                            |
|                | Ariana Porroche | Listado de libros, detalles del libro (sinopsis, otros libros, otros datos de interés, valoraciones), lector, iniciar sesión y registrarse. Libros leídos y en proceso, y foro.                                                  |
|                | Alex Giménez    | Modo claro/oscuro, buscador libros, gestión de perfil de usuario                                                                                                                                                                 |

[5.5]. Tabla de funciones software completadas por miembro.

Aunque en la tabla no quede reflejado, queremos destacar el coste que ha tenido el **aprender a gestionar un PDF**, tanto en web como en móvil, de una forma similar a un libro, añadiendo botones en la pantalla para pasar o retroceder la página, hacer zoom, así como añadir sus respectivas funcionalidades. La labor de investigación y pruebas realizadas para aprender a gestionar estas funcionalidades ha sido realizada por Javier y Diego en el ámbito web y por Ariana en el ámbito móvil.

Además de las funciones software, hemos realizado otras tareas que también han tenido gran coste de realización. En el caso del **despliegue** de backend, se realizó por parte de Diego Simón y de Javier Susín, así como el de web. El despliegue de móvil fue realizado por Javier Susín y Alex Giménez. La **documentación** principal fue realizada por Alex Giménez, mientras que Ariana Porroche realizó el análisis y diseño del sistema y Diego Simón realizó la documentación de backend mediante Swagger. La **presentación y demostración** ha sido llevada a cabo por Iván Vayad, Irene Pascual y Alex Giménez. Y las **pruebas** se han realizado entre todos los miembros, analizando posibles errores antes de actualizar el proyecto y mediante la inserción de acciones en GitHub Actions por parte de Irene Pascual, Lucía Vázquez, Iván Vayad y Ariana Porroche.

## Conclusiones

Como conclusión final, queremos llevar a cabo una reflexión sobre el trabajo realizado y los objetivos establecidos al inicio del mismo, destacar las fortalezas y debilidades del equipo a lo largo del proyecto, describir los principales retos del mismo, y las lecciones aprendidas y por último, los cambios que realizaríamos para realizar futuros proyectos.

El proyecto nos ha permitido cumplir con los **objetivos planteados** en gran medida. A lo largo del desarrollo, hemos aprendido a utilizar GitHub para la gestión de proyectos grupales, comprendiendo sus utilidades y la importancia del control de versiones en un entorno colaborativo. Además, logramos desarrollar un mismo producto para dos sectores distintos, adaptándolo tanto a la versión móvil como a la web. También hemos adquirido experiencia en la coordinación dentro de un equipo grande, afrontando los desafíos que implica, como la comunicación efectiva, el seguimiento del progreso, la aplicación de convenciones de nombrado y estándares de código. Finalmente, la formalización de entregas y la gestión del equipo humano nos han permitido mejorar nuestra organización y calendario.

A lo largo del proyecto, hemos demostrado varias **fortalezas** que han sido clave para su desarrollo exitoso. Destacamos nuestras ganas de aprender y trabajar, lo que nos ha permitido adaptarnos a los retos que surgieron en el camino. Además, aceptamos nuestras responsabilidades de manera comprometida, logrando una distribución equilibrada del trabajo. Contábamos con conocimientos previos sobre algunos entornos, lo que nos ayudó a avanzar con mayor fluidez. También hemos mantenido una buena coordinación y cooperación dentro del equipo, lo que nos permitió gestionar el tiempo de manera eficiente, distribuir el trabajo de forma equitativa y cumplir con los objetivos planteados.

Sin embargo, también hemos enfrentado algunas **dificultades**. En ciertos momentos, la carga de trabajo de otras asignaturas afectó nuestra disponibilidad, haciendo que algunas fases del proyecto fueran más exigentes o que no pudiéramos dedicarle tanto tiempo. Además, tuvimos que trabajar con alguna tecnología que no conocíamos, lo que representó un reto adicional y nos obligó a aprender rápidamente. A pesar de estos obstáculos, logramos superarlos gracias al esfuerzo conjunto y al compromiso de todo el equipo.

Como hemos mencionado en los párrafos anteriores, uno de los principales **retos** que enfrentamos en el proyecto fue aprender a utilizar tecnologías que no conocíamos. Esto nos obligó a investigar, hacer pruebas y apoyarnos entre nosotros para avanzar de manera eficiente. También fue un desafío adaptarnos a trabajar en un grupo tan grande, ya que al principio la organización resultaba caótica. Sin embargo, con el tiempo logramos coordinarnos mejor y establecer una comunicación más efectiva.

Otro **reto** importante fue reproducir los PDF de la manera deseada tanto en la web como en la aplicación móvil, lo que nos llevó a probar diferentes soluciones hasta encontrar la más adecuada. Además, tuvimos que hacer una buena distribución del trabajo y planificar un calendario realista que nos permitiera seguir avanzando en el proyecto, incluso en aquellas épocas en las que no podíamos dedicarle tanto tiempo. Para ello, intentamos avanzar lo máximo posible al inicio, anticipándonos a los periodos de mayor carga académica o menor disponibilidad. Para superar estos desafíos, fuimos ajustando nuestra planificación y metodología de trabajo, asegurándonos de que el esfuerzo de todos estuviera bien organizado y equilibrado.

Durante el proyecto, adquirimos una serie de **conocimientos y habilidades clave**. Aprendimos a trabajar con nuevas tecnologías como React Native y Vue, comprendiendo su estructura y funcionamiento. También mejoramos nuestro manejo de GitHub, no solo para control de versiones, sino también para la gestión de la construcción e integración del software. Otro aprendizaje importante fue la exportación de un proyecto React Native a APK, algo que nos permitió completar el desarrollo en la plataforma móvil. Además, desarrollamos habilidades en organización y gestión de un grupo grande, lo que implicó una mejor distribución del trabajo, coordinación y seguimiento del progreso. Aprendimos a mostrar PDFs tanto en web como en móvil, lo cual fue un desafío técnico relevante. También adquirimos conocimientos sobre análisis y gestión de riesgos en un proyecto, el seguimiento de estándares de desarrollo, la implementación de un control de calidad y la gestión de la planificación mediante diagramas de Gantt.

Entre las **herramientas y metodologías más útiles**, destacamos el uso de GitHub para la integración del código y la colaboración en equipo, y la planificación mediante diagramas de Gantt, que nos ayudó a estructurar y seguir el progreso del proyecto de manera eficiente.

Por último, **de cara a otros proyectos**, si tuviéramos que comenzar uno nuevo aplicando una metodología de gestión basada en procesos, **introduciríamos algunos cambios** relevantes para optimizar el desarrollo. En primer lugar, estableceríamos desde el inicio una estructura más clara de roles y responsabilidades, para evitar una posible confusión inicial y mejorar la organización del equipo. También implementaríamos reuniones de seguimiento más frecuentes y breves, especialmente al principio del proyecto: al inicio podrían ser incluso diarias, para comprender el trabajo de los demás y evitar pasos en falso que nos obliguen a retroceder. A medida que el proyecto avanza, estas reuniones podrían pasar a ser semanales, para compartir avances, identificar el trabajo pendiente y coordinar los siguientes objetivos. Hemos observado que cuando parte del equipo ya conoce las herramientas y otra parte no, puede generarse un desequilibrio en el ritmo de trabajo. Esto puede hacer que algunos miembros se sientan descolgados o perdidos en las fases

iniciales. Por ello, sería útil planificar formaciones breves o tiempos de adaptación desde el comienzo.

Por otro lado, **mantendríamos muchas de las prácticas** que sí han funcionado bien, como el uso intensivo de GitHub para control de versiones y gestión de tareas, la anticipación del trabajo antes de los periodos de mayor carga académica, la división modular del proyecto para que cada miembro pudiera avanzar de forma más independiente y sobretodo la elección de trabajo a realizar mediante las *Issues* de Github, ya que han aportado frescura y libertad al trabajo. En definitiva, repetiríamos las buenas prácticas adquiridas, pero siendo más realistas en cuanto al tiempo y más proactivos en la organización desde el principio.

En definitiva, repetiríamos las buenas prácticas adquiridas, pero siendo más realistas en cuanto al tiempo necesario y más proactivos en la organización desde el inicio.

# Bibliografía y referencias

[1]. Enlace acceso backend en render: <https://backend-dcy8.onrender.com>

[2]. Página sobre el WCAG 2.1 del W3C:  
<https://www.w3.org/WAI/standards-guidelines/wcag/glance/es>

[3]. Guía de diseño para accesibilidad de Android:  
<https://developer.android.com/design/ui/mobile/guides/foundations/accessibility?hl=es-419>

[4]. Guía para el uso de la API de Mistral Le Chat: <https://docs.mistral.ai/api/>

Página de información de Google Drive:  
<https://workspace.google.com/intl/es/products/drive/>

Página de inicio de la base de datos online Neon: <https://neon.tech/>

Página de inicio e Figma: <https://www.figma.com/>

Página de inicio de GitHub: <https://github.com/>

Página de inicio de Render: <https://render.com/>

Página de inicio de Expo Go: <https://expo.dev/go>

# Apéndice I: Análisis de riesgos.

A continuación, presentamos las tablas de análisis de riesgos elaboradas para el proyecto. En ellas detallamos los riesgos identificados, su probabilidad de ocurrencia, impacto, los responsables de cada riesgo, estrategias de mitigación, su aparición o no aparición y si ha sido efectiva la mitigación.

| Código | Descripción                                                                                  | Riesgo S/N | Probab. 1--5 | Impacto 1--5 | Exp. al riesgo Prob X Imp |
|--------|----------------------------------------------------------------------------------------------|------------|--------------|--------------|---------------------------|
| Ri1    | Hay integrantes del equipo que tienen muchos proyectos (y/o asignaturas) a la vez            | S          | 4            | 3            | 12                        |
| Ri2    | Todo, o parte del equipo, nunca ha trabajado junto                                           | S          | 3            | 3            | 9                         |
| Ri3    | Todo, o parte del equipo, sí ha trabajado de manera conjunta                                 | N          |              |              |                           |
| Ri4    | Nunca se ha trabajado en un grupo tan grande                                                 | S          | 4            | 2            | 8                         |
| Ri5    | El conocimiento o desconocimiento del cliente (profesor)                                     | S          | 4            | 3            | 12                        |
| Ri6    | El proyecto cuenta con unas funcionalidades que no están muy claramente definidas            | S          | 2            | 4            | 8                         |
| Ri7    | Nunca se ha construido un sistema tan complejo como el propuesto                             | S          | 4            | 2            | 8                         |
| Ri8    | Se ha elegido una tecnología que se conoce poco o nada                                       | S          | 2            | 2            | 4                         |
| Ri9    | Se ha trabajado, o no, con el entorno de despliegue elegido                                  | S          | 2            | 2            | 4                         |
| Ri10   | La fecha de entrega está muy lejos en el tiempo                                              | N          |              |              |                           |
| Ri11   | El proyecto hay que desarrollarlo en dos fases                                               | S          | 2            | 1            | 2                         |
| Ri12   | El proyecto requiere la entrega de un sistema funcionando y una documentación complementaria | N          |              |              |                           |

[AI.1]. Tabla de riesgos del proyecto.

| Código de riesgo | Descripción                                                                       | Estrategia de mitigación                                                                                                                    | Responsable | ¿Se presentó? (S/N) | ¿Ha resultado efectiva la mitigación? |
|------------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------------------|---------------------------------------|
| Ri1              | Hay integrantes del equipo que tienen muchos proyectos (y/o asignaturas) a la vez | Distribuir el trabajo tratando de aprovechar la época en la que se dispone de mas tiempo para adelantar trabajo.                            | Luna        | S                   | S                                     |
| Ri2              | Todo, o parte del equipo, nunca ha trabajado junto                                | Organizar reuniones, dar a conocer los conocimientos y habilidades de cada uno, y permitir una comunicación abierta.                        | Ariana      | S                   | S                                     |
| Ri4              | Nunca se ha trabajado en un grupo tan grande                                      | Realizar una correcta división de tareas y dejar el trabajo ordenado.                                                                       | Lucía       | S                   | S                                     |
| Ri5              | El conocimiento o desconocimiento del cliente (profesor)                          | Realizar una mayor cantidad de entrevistas y reuniones y dejar las previsiones del proyecto claras desde el inicio.                         | Alex        | S                   | S                                     |
| Ri6              | El proyecto cuenta con unas funcionalidades que no están muy claramente definidas | Dejar las funcionalidades claramente definidas desde el principio con la verificación del cliente.                                          | Ariana      | S                   | S                                     |
| Ri7              | Nunca se ha construido un sistema tan complejo como el propuesto                  | Llevar una planificación y división estricta del trabajo, prestar mayor atención a las dificultades e investigar sobre proyectos similares. | Irene       | S                   | S                                     |
| Ri8              | Se ha elegido una tecnología que se conoce poco o nada                            | Realizar una formación inicial para aquellos que no la conocen mediante información interna o externa.                                      | Diego       | S                   | S                                     |
| Ri9              | Se ha trabajado, o no, con el entorno de despliegue elegido                       | Investigar acerca de ejemplos de como utilizar el mismo y realizar pequeñas pruebas a parte.                                                | Javier      | N                   |                                       |
| Ri11             | El proyecto hay que desarrollarlo en dos fases                                    | Establecer deadlines estrictos de pequeñas partes del proyecto y dejar claro dónde empieza y donde acaba cada fase.                         | Iván        | N                   |                                       |

[AI.2]. Tabla de mitigación de riesgos del proyecto.

# **Anexo I. Resultados de la revisión intermedia**

Tras realizar la revisión intermedia del trabajo, hemos encontrado algunos detalles a tener en cuenta, y hemos obtenido diversas conclusiones. En primer lugar hemos detectado la falta de issues para las tareas de realizar el despliegue del sistema, documentar los procesos realizados, o realizar pruebas. No estamos siguiendo ninguna guía de estilo de comunicación estricta, sino que hemos llegado a un acuerdo entre nosotros. No hemos realizado la documentación de la API web que expone el backend, ni pruebas formales por el momento.

Para corregir estos problemas, hemos decidido aplicar algunas soluciones, como realizar un repaso de las issues faltantes y añadirlas a github, además hemos decidido realizar la documentación de la API del backend mediante Swagger, y comenzaremos a realizar pruebas de forma formal.

<https://docs.google.com/spreadsheets/d/1ZM2hPeq57rXDmTBtl6xek4X8d5rtZF0x/edit?usp=sharing&ouid=110195556502424108042&rtpof=true&sd=true>

# Anexo II. Presentación final

La presentación de BOOKLY ha seguido los principios vistos en los temas teóricos sobre cómo hacer buenas presentaciones. A continuación vamos a explicar qué hemos utilizado:

## 1. Idea central clara y “regla de los 3 segundos”

Todo diseño parte de una sola “gran idea”, fácilmente identificable en menos de 3 segundos. En general, esta idea principal se transmite mediante el título de la diapositiva, con tipografía grande y limpia, en negrita. Se acompaña con alguna ilustración relacionada.

El texto breve y las imágenes funcionan como apoyo al discurso, nunca como sustituto.

## 2. Tipografía y legibilidad accesible

Usamos siempre tipos de letra clara (como Roboto Mono, Lexend, Abel o Arial), con un tamaño generoso y con suficiente contraste respecto al fondo. Todo esto garantiza que el mensaje se lea sin esfuerzo.

## 3. Espacio en blanco y diseño minimalista

El espaciado genera “respiración” visual y dirige el foco. Además no hace tan pesado para el público el seguimiento de la presentación.

## 4. Coherencia cromática y de estilo

Hemos seguido una paleta reducida (tonos amarillos, marrones y neutros), que hemos aplicado coherentemente. Esto refuerza la identidad de la presentación (ya que además los colores elegidos son los mismos colores que determinamos para nuestra aplicación Bookly).

Estos colores están debidamente comprobados los niveles de contraste, y hace tanto la presentación como la aplicación, más accesible.

## 5. Jerarquía visual y patrón de lectura natural

Organizamos el contenido siguiendo el patrón F-Z (título arriba, texto centro-izquierda, imagen derecha), facilitando el “escaneo” por parte de la audiencia.

## 6. Uso de elementos visuales sencillos y relevantes

Iconos o fotografías bien elegidos transmiten mucho más que párrafos de texto. En adición, hace más ligera la presentación y mantiene al público más interesado.

## 7. Storytelling y enfoque en el público objetivo

Incorporamos dos microhistorias, como la de Lucía y Lucas, que logra humanizar desde el principio la presentación al mostrar el “antes y después”. Con esto se logra empatía y se subraya el valor del producto.

Además, el texto breve y emotivo acompañado de imágenes, muestran expresiones reales, y esto refuerza el vínculo con la audiencia.

#### **8. Organización en bloques diferenciados**

El contenido se divide en secciones bien definidas, mediante una slide como título. Esto guía a la audiencia y evita la sobrecarga cognitiva.

Cada bloque aporta un hilo lógico y prepara al público para el siguiente.

#### **9. Comparaciones efectivas y unidades gráficas repetidas**

En comparaciones, o imágenes o logos incorporados, se mantiene siempre la misma escala y estilo: mismos tamaños, formato de texto y espaciado igual.

#### **10. Visibilizar el trabajo en equipo y las lecciones aprendidas**

Mostrar brevemente al equipo, los retos superados y las lecciones aprendidas aporta credibilidad. Demuestra también reflexión y refuerza la confianza en el proyecto.

#### **11. Llamada a la acción inspiradora**

Se reserva la última diapositiva para un mensaje motivador: invita a probar Bookly, crearse una cuenta o compartir la experiencia. Esto deja en el público una impresión emocional y un “siguiente paso” bien definido.

# BOOKLY

## Volver a leer nunca fue tan fácil

Abril 2025 – Presentación Proyecto Final

Equipo: Iván, Diego, Javier, Lucía, Irene, Ariana, Luna y Alex



### Nuestro equipo



- Javier
- Diego
- Irene
- Alex



- Lucía
- Iván
- Irene
- Diego
- Javier



- Luna
- Ariana
- Alex



- Alex
- Ariana

# ÍNDICE

1. INTRODUCCIÓN
2. CÓMO SE ALCANZA ESA VISIÓN
3. EXPLORANDO BOOKLY
4. COMUNIDAD DE BOOKLY
5. EL PLUS INESPERADO
6. REALIZACIÓN
7. REFLEXIONES FINALES

## 1. INTRODUCCIÓN

*"Bookly nace para acompañarte de nuevo al mundo de la lectura."*



## 2. CÓMO SE ALCANZA ESA VISIÓN

*"Conectando emociones, motivaciones y hábitos para volver a disfrutar leyendo."*



### LA HISTORIA DE LUCÍA...



*"Volver a leer es posible.  
Bookly lo hizo para Lucía."*



## LA HISTORIA DE LUCAS...



"Lucas quería leer fácil y gratis.  
Bookly le dio el cómo."



## ¿Y SI LEER FUERA TAN ADICTIVO COMO ESCUCHAR MÚSICA O APRENDER JUGANDO?

Música  
personalizada



"escuchar"

Aprendizaje  
gamificado



"aprender"

Lectura  
personalizada



"leer"

## 3. EXPLORANDO BOOKLY

"Leer, organizar tus  
libros y disfrutar."







## DETALLES DE UN LIBRO



**Página Web:**



**Aplicación Móvil:**

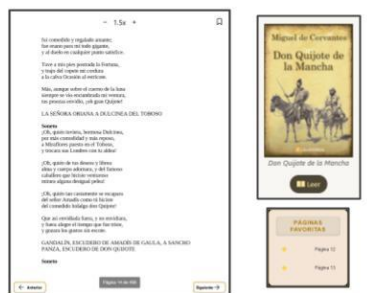




## LECTOR



**Página Web:**



**Aplicación Móvil:**





## LISTAS PERSONALES

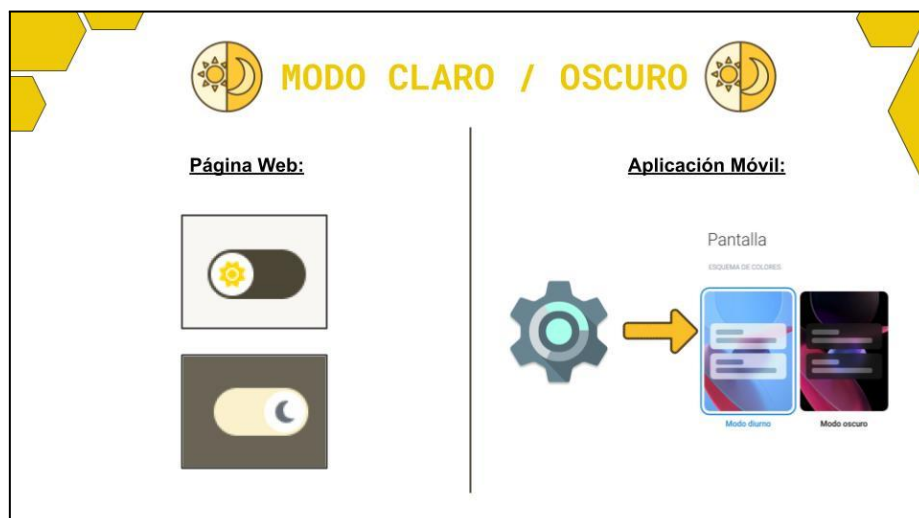


**Página Web:**



**Aplicación Móvil:**





## 4. COMUNIDAD DE BOOKLY

*"Leer no tiene que ser algo solitario"*



FORO

**Página Web:**

**Aplicación Móvil:**

VALORACIONES

**Página Web:**

**Aplicación Móvil:**

ESTADÍSTICAS

**Página Web:**

**Aplicación Móvil:**

## 5. EL PLUS INESPERADO

*"No sabíamos que lo necesitábamos... hasta que empezamos a hablar con él."*



### CHATBOT DE BOOKLY

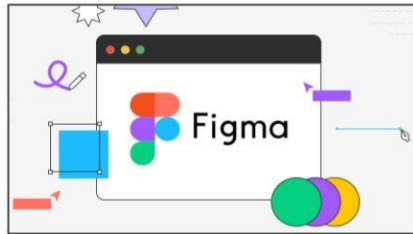


## 6. REALIZACIÓN

*"Las mejores ideas surgen hablando... y se hacen realidad entre bocetos."*



## ¿CÓMO LLEVAMOS A CABO ESTE PROYECTO ENTRE 8 PERSONAS?



## DIFICULTADES SUPERADAS

- Organización
- Nuevas tecnologías
- Despliegues, configuraciones... fallidos



## LECCIONES APRENDIDAS



- Comunicación ante todo
- Resolver juntos problemas
- Avanzar sin perfección
- Todos sumamos siempre



## >>I SIGUIENTES PASOS >>I

- *Notificaciones push*
- *Lectura offline*
- *Clubes*
- *Logros y retos*



## 7. REFLEXIONES FINALES

*"Bookly ya nos hizo  
volver a leer.  
Ahora te toca a ti"*



¿Qué pasa cuando una  
app conecta con lo que  
de verdad te hace falta?



**BOOKLY**

# Glosarios

**Android Studio:** Entorno de desarrollo integrado (IDE) oficial para desarrollar aplicaciones Android.

**API (Application Programming Interface):** Conjunto de reglas y herramientas que permiten que diferentes sistemas o aplicaciones se comuniquen entre sí.

**APK (Android Package Kit):** Formato de archivo utilizado para distribuir e instalar aplicaciones en dispositivos Android.

**Autoescalado:** Técnica en la computación en la nube que ajusta automáticamente los recursos según la demanda.

**Backend:** Parte de una aplicación o sistema que maneja la lógica de negocio, bases de datos y servidores, funcionando en el lado del servidor.

**Bootstrap:** Framework de código abierto para diseñar sitios web responsive y móviles mediante HTML, CSS y JavaScript.

**Commit (GitHub):** Acción de guardar cambios en un repositorio de Git, creando un historial de versiones.

**CSS (Cascading Style Sheets):** Lenguaje de estilos utilizado para definir la apariencia y el diseño de páginas web.

**Dart:** Lenguaje de programación desarrollado por Google, utilizado principalmente en el desarrollo de aplicaciones con Flutter.

**Deno:** Entorno de ejecución para JavaScript y TypeScript, creado como una alternativa a Node.js.

**Diagrama de Gantt:** Herramienta de gestión de proyectos que representa visualmente el cronograma de tareas en un gráfico de barras.

**Express:** Framework para Node.js que facilita la creación de aplicaciones web y APIs.

**Expo Go:** Aplicación que permite ejecutar y probar proyectos desarrollados con React Native sin necesidad de compilarlos.

**Framework:** Conjunto de herramientas y buenas prácticas que facilitan el desarrollo de software proporcionando una estructura base.

**Frontend:** Parte visual e interactiva de una aplicación o sitio web con la que el usuario interactúa directamente.

**Git Flow (GitHub):** Modelo de trabajo en Git que define un proceso estructurado para gestionar el desarrollo de software mediante ramas.

**GitHub:** Plataforma de alojamiento de código basada en Git que permite la colaboración en proyectos de software.

**GitHub Actions:** Es una plataforma de automatización integrada en GitHub que permite crear flujos de trabajo para la integración y entrega continua (CI/CD). Permite ejecutar pruebas, compilar código y desplegar aplicaciones automáticamente en función de eventos como commits o pull requests.

**Google Docs:** Herramienta de procesamiento de texto en línea que permite la edición y colaboración en documentos en la nube.

**Google Drive:** Servicio de almacenamiento en la nube de Google que permite guardar y compartir archivos en línea.

**GUI (Graphical User Interface):** Interfaz gráfica de usuario que facilita la interacción con un sistema mediante elementos visuales.

**HTML (HyperText Markup Language):** Lenguaje de marcado utilizado para estructurar y diseñar páginas web.

**HTTP (HyperText Transfer Protocol):** Protocolo de comunicación utilizado en la web para la transferencia de datos entre servidores y clientes.

**Issue (GitHub):** Funcionalidad en GitHub que permite reportar errores, sugerir mejoras y gestionar tareas dentro de un repositorio.

**Java:** Lenguaje de programación de propósito general, ampliamente utilizado en desarrollo web, móvil y empresarial.

**Javascript:** Lenguaje de programación interpretado y de alto nivel utilizado para desarrollar aplicaciones web interactivas.

**JSON (JavaScript Object Notation):** Formato ligero de intercambio de datos basado en texto, fácil de leer y escribir para humanos y máquinas.

**Merge (GitHub):** Acción de combinar cambios de una rama en otra dentro de un repositorio de Git.

**MongoDB Atlas:** Servicio en la nube para bases de datos NoSQL de MongoDB, que facilita la escalabilidad y administración.

**Neon:** Plataforma de base de datos PostgreSQL en la nube, diseñada para escalabilidad y eficiencia.

**Netlify:** Plataforma para el desarrollo y despliegue de sitios web estáticos y aplicaciones modernas.

**Node.js:** Entorno de ejecución de JavaScript basado en el motor V8 de Chrome, que permite ejecutar código fuera del navegador.

**npm:** Es el gestor de paquetes por defecto para Node.js. Permite instalar, administrar y compartir dependencias y bibliotecas de JavaScript, facilitando el desarrollo de aplicaciones al proporcionar acceso a miles de paquetes disponibles en su registro.

**PDF (Portable Document Format):** Formato de archivo diseñado para presentar documentos de manera independiente del software, hardware o sistema operativo.

**PostgreSQL:** Sistema de gestión de bases de datos relacional de código abierto, conocido por su robustez y escalabilidad.

**Postman:** Herramienta utilizada para desarrollar, probar y documentar APIs.

**React Native:** Framework de desarrollo de aplicaciones móviles basado en React que permite crear apps para iOS y Android con JavaScript.

**Repositorio:** Espacio de almacenamiento en Git o plataformas como GitHub donde se guardan proyectos y su historial de versiones.

**REST (Representational State Transfer):** Estilo de arquitectura para el desarrollo de APIs basado en principios como la comunicación sin estado y el uso de HTTP.

**RESTful:** Término que describe una API que sigue los principios de la arquitectura REST.

**Software:** Conjunto de programas, instrucciones y datos que permiten la operación de dispositivos electrónicos.

**SQL (Structured Query Language):** Lenguaje utilizado para gestionar y consultar bases de datos relacionales.

**Stakeholder:** Persona o grupo con interés en un proyecto o empresa, como clientes, empleados o inversionistas.

**UML (Unified Modeling Language):** Lenguaje de modelado visual utilizado en ingeniería de software para diseñar y documentar sistemas.

**URL (Uniform Resource Locator):** Dirección que identifica un recurso en la web, como una página o un archivo.

**VSCode (Visual Studio Code):** Editor de código fuente desarrollado por Microsoft, popular por su flexibilidad y extensibilidad.

**Vue.js:** Framework progresivo de JavaScript utilizado para construir interfaces de usuario y aplicaciones web de una sola página.

**WCAG (Web Content Accessibility Guidelines):** Conjunto de directrices para hacer que el contenido web sea accesible para personas con discapacidades.

# Índice de figuras

| Nº                   | Figura                                       |
|----------------------|----------------------------------------------|
| <a href="#">Al.1</a> | Tabla de riesgos del proyecto.               |
| <a href="#">Al.2</a> | Tabla de mitigación de riesgos del proyecto. |
| <a href="#">3.1</a>  | Diagrama de Gantt del proyecto               |
| <a href="#">4.7</a>  | Vista de módulos del cliente móvil           |
| <a href="#">4.8</a>  | Vista de módulos del cliente web             |
| <a href="#">4.9</a>  | Vista de de módulos del backend              |
| <a href="#">4.10</a> | Vista de componentes y conectores            |
| <a href="#">4.11</a> | Vista de despliegue                          |
| <a href="#">4.12</a> | Modelo entidad-relación de la base de datos  |
| <a href="#">4.13</a> | Modelo relacional de la base de datos        |

# Índice de tablas

| Nº                  | Tabla                                                      |
|---------------------|------------------------------------------------------------|
| <a href="#">2.1</a> | Tabla de subgrupos de trabajo primera fase                 |
| <a href="#">2.2</a> | Tabla de subgrupos de trabajo segunda fase                 |
| <a href="#">2.3</a> | Tabla de roles y responsabilidades por miembro del equipo  |
| <a href="#">3.2</a> | Tabla de realización de tareas generales.                  |
| <a href="#">3.3</a> | Tabla de realización del trabajo                           |
| <a href="#">4.1</a> | Tabla de requisitos funcionales del usuario                |
| <a href="#">4.2</a> | Tabla de requisitos funcionales del usuario logueado       |
| <a href="#">4.3</a> | Tabla de requisitos funcionales del usuario no logueado    |
| <a href="#">4.4</a> | Tabla de requisitos no funcionales del usuario             |
| <a href="#">4.5</a> | Tabla de requisitos no funcionales del usuario logueado    |
| <a href="#">4.6</a> | Tabla de requisitos no funcionales del usuario no logueado |
| <a href="#">5.1</a> | Tabla de horas esperadas por área                          |
| <a href="#">5.2</a> | Tabla de horas reales                                      |
| <a href="#">5.3</a> | Tabla de líneas de código esperadas por área               |
| <a href="#">5.4</a> | Tabla de líneas de código reales                           |
| <a href="#">5.5</a> | Tabla de funciones software completadas por miembro        |