

UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia - Sede Bogotá

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas e Industrial

Asignatura:

Ingeniería de Software I

Proyecto - Taller Requerimientos y Clean Code

Natalia Carolina Bautista Sanchez

Daniel Santiago Cocinero Jimenez

Baruj Vladimir Ramirez Escalante

Juan Daniel Ramirez Mojica

Profesor:

Oscar Eduardo Alvarez Rodriguez

2024 - II

Bogotá D.C.

Índice

Selección del Proyecto a Desarrollar	3
Levantamiento de Requerimientos	4
1. Contexto	4
2. Resumen Inicial del Problema	4
3. Beneficios para cada integrante	4
4. Restricciones y Alcance del Proyecto	5
5. Impacto en el Negocio	5
6. Viabilidad Técnica	6
7. Levantamiento de Requerimientos	6
Análisis de Requerimientos y Clasificación con MoSCoW	11
1. Preguntas que surgieron:	11
2. Clasificación MoSCoW:	11
2. Estimación	14
Análisis de Gestión de Software	19
1. Tiempo	19
2. Costo	19
3. Alcance	22
Adjuntos	23

Selección del Proyecto a Desarrollar

Para el desarrollo del proyecto, se decidió trabajar en la propuesta “Club de Lectura Digital”, seleccionado mediante un proceso iterativo de evaluación y priorización.

A partir de una lluvia de ideas inicial, surgieron doce propuestas, las cuales nacieron tanto de las necesidades observadas en nuestras experiencias personales como de nuestros hobbies y del deseo de realizar algo diferente y aprender nuevas herramientas, o incluso de una combinación de ambas.

Para el proceso de consenso dentro del equipo, se adaptó una herramienta llamada “**Selección de cartas**” con el propósito de evaluar y priorizar las ideas. El procedimiento para utilizar esta herramienta es el siguiente:

1. Escriba cada una de las opciones en tarjetas individuales.
2. Cuente el número total de opciones, ya que este será el número máximo de puntuación.
3. Cada miembro del equipo selecciona la opción que menos le convenza, asignándole el número 1, y la opción que más le guste, asignándole el número máximo. Las demás opciones se clasificarán según esta misma regla de puntuación.
4. Pondere los puntajes de todos los miembros del equipo.
5. Realice una breve discusión con el equipo, donde se compartan puntos de vista y se clarifiquen las razones detrás de las preferencias y descartes.
6. Elimine las opciones con los puntajes más bajos.
7. Repita el proceso de manera iterativa hasta que quede una única opción.

De esta manera, cada integrante otorgó puntajes para destacar las opciones más relevantes. En cada iteración, las ideas se discutieron y redujeron progresivamente, concentrándose solo en aquellas con mayor potencial. Este enfoque permitió refinar las alternativas y alcanzar un consenso grupal, culminando en la elección del “Club de Lectura Digital” como la propuesta más adecuada. Puede consultar los resultados de este proceso en el adjunto número **uno** y número **dos**.

Levantamiento de Requerimientos

1. Contexto

El proyecto surge de la propuesta de un integrante del equipo, que observó una oportunidad de negocio en base a uno de sus hobbies, el objetivo de esta propuesta es crear una plataforma digital para fomentar la lectura colectiva y la discusión de libros. Esta iniciativa **busca unir a personas con diversos intereses en un espacio virtual donde puedan compartir experiencias lectoras, debatir ideas y conectarse a través de su amor por los libros**. Para esto se puede usar las plataformas digitales que tienen la capacidad de superar ciertas barreras tradicionales que enfrentan los clubes de lectura, tales como **la dificultad para coordinar horarios, la falta de espacios físicos accesibles para todos los interesados y las limitaciones para conectar a personas con intereses similares**, especialmente en comunidades geográficamente dispersas.

2. Resumen Inicial del Problema

Se enfrenta el desafío de diseñar y desarrollar una plataforma intuitiva y accesible que permita a los usuarios crear, gestionar y participar en clubes de lectura en línea. Además, se requiere un sistema que facilite las discusiones, el seguimiento del progreso de lectura y la toma de decisiones colectivas, como la selección de nuevos libros. La solución debe garantizar una experiencia fluida y atractiva para los usuarios, promoviendo la participación activa y el sentido de comunidad.

Los usuarios potenciales del sistema esperan poder compartir sus ideas y participar en discusiones organizadas por temas, así como contar con un sistema que les permita decidir el libro a leer (como lo podría ser un sistema de votación). También esperan una plataforma que permita realizar un seguimiento del progreso de lectura y se genere una experiencia participativa que fomente la interacción entre los miembros y fortalezca el sentido de comunidad.

3. Beneficios para cada integrante

Juan Daniel Ramírez Mojica: A través de este proyecto, espero aprender a utilizar nuevas herramientas que aún no he explorado en profundidad, como la creación de un chat y el uso de bases de datos no relacionales. Del mismo modo, me gustaría poner en práctica los conocimientos que he adquirido en distintas clases en un proyecto más aproximado a la realidad, especialmente en temas recientes como clean code, patrones de diseño y testing, ya que son conceptos relativamente nuevos para mí.

Natalia Carolina Bautista Sanchez: En este proyecto busco aprender a hacer un buen desarrollo frontend, aprendiendo a hacer interfaces que sean estéticas y al mismo tiempo fáciles de utilizar para los usuarios, dándoles una experiencia agradable e intuitiva. Quisiera aprender en el futuro acerca de desarrollo UI/UX, este proyecto me proporciona unas buenas bases para comenzar a aprender del tema.

Daniel Santiago Cocinero Jimenez: Mi objetivo con este proyecto es aprender a utilizar Spring Boot, entender cómo implementar una base de datos no relacional y como es el desarrollo web, puesto que nunca he hecho un backend y me gustaria entender como se conecta con el front.

Baruj Vladimir Ramirez Escalante: Con este proyecto, busco implementar técnicas para el diseño y desarrollo de software, principalmente el desarrollo con IA, pues parece ser que de aquí a futuro se verá cada vez más presente en el desarrollo de proyectos.

4. Restricciones y Alcance del Proyecto

- **Tiempo:** El plazo con el que se cuenta es de alrededor de un mes, ya que se debe entregar el día 6 de marzo del 2025.
- **Presupuesto: \$15.332.744.** Esto se verá reflejado más adelante.
- **Recursos:** El equipo de desarrollo está compuesto por 4 desarrolladores full-stack.
- **Alcance:** Dado que el tiempo y los recursos son limitados, se priorizará la creación de un Producto Mínimo Viable (MVP), enfocándose únicamente en las funcionalidades esenciales que permitan lanzar una versión básica pero funcional.

5. Impacto en el Negocio

Este proyecto tiene el potencial de generar un impacto en diversos aspectos: En términos de beneficios, se espera incrementar el “engagement” de los usuarios, al ofrecer una experiencia interactiva y en cierto modo personalizada que fomente una participación activa.

Siguiendo con el punto anterior, desde un punto de vista más económico, se ve potencial adquiriendo posible financiación por parte de entidades públicas como el ministerio de cultura, con el que se fomenten hábitos de lectura; del mismo modo que se podrían encontrar oportunidades comerciales, como alianzas con editoriales.

Sin embargo, también se identifican riesgos asociados al comportamiento del mercado, la adopción por parte de los usuarios y la competencia de plataformas similares (cómo podría ser goodreads, aunque se podría realizar una alianza), estos aspectos se deben manejar de manera cuidadosa para garantizar el éxito de este proyecto.

6. Viabilidad Técnica

Este proyecto se considera técnicamente viable debido a la experiencia y habilidades del equipo de desarrollo, compuesto por cuatro desarrolladores full-stack. El equipo cuenta con conocimientos sólidos en Java, JavaScript, HTML, CSS, bases de datos relacionales y desarrollo backend, lo que garantiza una base técnica adecuada para el proyecto. Sin embargo, aspectos como la integración de bases de datos no relacionales y la implementación de ciertas funcionalidades específicas requerirán tiempo adicional de investigación y pruebas.

En términos de tecnología, se dispone de herramientas modernas que pueden facilitar el desarrollo, como Spring Boot para la construcción del backend, bases de datos relacionales (SQL) y no relacionales (MongoDB), así como APIs externas para ampliar la funcionalidad del sistema.

A pesar de esto, se deben considerar posibles obstáculos, como la curva de aprendizaje de nuevas tecnologías y los desafíos asociados con la integración de distintos servicios y herramientas dentro del proyecto.

7. Levantamiento de Requerimientos

Para llevar a cabo este proyecto, se planteó inicialmente realizar una entrevista como método de recopilación de información. Los miembros del grupo se reunieron a través de una videollamada por Google Meet, durante la cual discutieron las ideas que cada uno tenía sobre la aplicación. Durante la reunión, se logró un consenso general acerca de las características y funcionalidades principales de la aplicación, al tiempo que se generaron los mock ups iniciales de la aplicación, mientras la llamada era grabada para fines de documentación.

Posteriormente, la grabación fue procesada mediante el modelo de transcripción Whisper, lo que permitió generar un texto transcrito de la reunión. Esta transcripción se utilizó como entrada en ChatGPT, con el objetivo de extraer una lista de requerimientos basada en el consenso alcanzado por el grupo.

La transcripción de la videollamada como los mockups iniciales, se añaden como anexo **tres y cuatro respectivamente**.

Cómo resultado, se obtuvieron los siguientes requerimientos:

Requerimientos Funcionales

01. Registro e inicio de sesión: La plataforma debe permitir la gestión de cuentas de usuarios, permitiendo tanto la creación como la actualización de registros de clientes

Proceso: Autenticación de usuarios.

Funcionalidades:

- a. Registro de nuevos usuarios mediante formulario (nombre, correo electrónico, contraseña).
- b. Inicio de sesión para usuarios registrados.
- c. Recuperación de contraseña mediante correo electrónico.
- d. Almacenamiento seguro de contraseñas usando algoritmos de cifrado (ej: bcrypt).

02. Exploración de clubes: La plataforma debe permitir a los usuarios explorar y buscar clubes que les interesen y de los que quieran hacer parte.

Proceso: Descubrimiento y búsqueda de clubes.

Funcionalidades:

- a. Listado de clubes destacados o populares en la página principal.
- b. Barra de búsqueda para encontrar clubes por:
 - i. Nombre.
 - ii. Género literario.
 - iii. Hashtags.
 - iv. Idioma.

- c. Sugerencias de clubes basadas en las preferencias del usuario.
- d. Visualización de detalles del club antes de unirse (nombre, descripción, libro actual, imagen).
- e. Opción para unirse a clubes:
 - i. Públicos.
 - ii. Privados (requieren contraseña para unirse).

03. Gestión de clubes: La plataforma debe permitir que los usuarios puedan crear y gestionar clubes de acuerdo a las características que deseen.

Proceso: Creación y administración de clubes.

Funcionalidades:

- a. Crear un nuevo club ingresando:
 - i. Nombre del club.
 - ii. Imagen representativa.
 - iii. Descripción breve.
 - iv. Estado del club (público o privado).
- b. Configuración del libro actual a leer en el club.
- c. Visualización del historial de libros leídos.
- d. Edición de información del club (solo por el administrador del club).
- e. Eliminación del club (solo por el administrador del club).

04. Participación en clubes: La plataforma debe permitir la interacción entre miembros de un club. Los usuarios podrán debatir en secciones específicas, crear temas de discusión y votar por el próximo libro.

Proceso: Interacción entre miembros dentro de un club.

Funcionalidades:

- a. Visualización de información del club:
 - i. Nombre.
 - ii. Imagen.

- iii. Libro actual.
- iv. Descripción.
- v. Avisos importantes.
- b. Espacio de debates con:
 - i. Secciones para discutir temas específicos.
 - ii. Posibilidad de crear temas de discusión (requiere aprobación del administrador).
 - iii. Comentarios y respuestas dentro de los temas.
- c. Sistema de votación para decidir el próximo libro a leer.
 - i. Indicador de progreso de lectura:
 - ii. Los usuarios pueden marcar si han terminado el libro que se está leyendo actualmente.
- d. Visualización del porcentaje de usuarios que han completado el libro.

05. Notificaciones: La plataforma debe informar a los usuarios sobre cambios en el libro, nuevos temas de discusión y resultados de votaciones.

Proceso: Información al usuario sobre actualizaciones relevantes.

Funcionalidades:

- a. Notificaciones sobre:
 - i. Cambios en el libro actual.
 - ii. Nuevos temas de discusión.
 - iii. Resultados de las votaciones.
- b. Configuración de preferencias de notificaciones (activadas/desactivadas por usuario).
- c. Envío de notificaciones por correo electrónico.

06. Gestión de roles: La plataforma debe permitir la asignación de roles con diferentes permisos. Los administradores tendrán control sobre la gestión del club, mientras que los miembros regulares podrán participar en la visualización, comentarios y votaciones.

Proceso: Asignación y manejo de permisos dentro de un club.

Funcionalidades:

- a. Rol de administrador:
 - i. Crear y editar temas de discusión.
 - ii. Aprobar temas propuestos por miembros.
 - iii. Editar información del club.
 - iv. Moderar miembros (expulsar o aceptar nuevos ingresos).
- b. Roles básicos para miembros regulares (visualizar, comentar, votar).

07. Perfil de usuario: La plataforma debe permitir que el usuario gestione la información de su perfil que desea compartir a los demás.

Proceso: Gestión de información del usuario.

Funcionalidades:

- a. Visualización y edición del perfil personal:
 - i. Nombre.
 - ii. Imagen de perfil.
 - iii. Descripción breve ("sobre mí").
 - iv. Lista de clubes a los que pertenece.
- b. Opcional: Mostrar un historial de libros leídos o géneros favoritos.

Requerimientos No Funcionales

01. Seguridad y privacidad: La plataforma debe garantizar la seguridad de los datos mediante el cifrado de contraseñas y datos sensibles. Los clubes podrán configurarse como privados, con contraseñas seguras, y se deberán aplicar restricciones de acceso a funcionalidades avanzadas según el rol del usuario.

02. Diseño Intuitivo: La plataforma debe contar con un diseño intuitivo que permita al usuario utilizar de manera fácil y eficiente; sin confundirlo ni sobrecargarlo con opciones innecesarias. Además, debe tener un diseño claro y accesible.

03. Rendimiento: La plataforma debe ser capaz de manejar múltiples solicitudes de usuarios de manera eficiente, garantizando tiempos de respuesta rápidos y un rendimiento óptimo.

Análisis de Requerimientos y Clasificación con MoSCoW

1. Preguntas que surgieron:

- ¿Cuál es el número máximo de miembros que puede tener un club?
- ¿Cómo se van a realizar las notificaciones: solo por correo electrónico o también en la página?
- ¿Qué criterios se utilizará para mostrar el listado de clubes destacados o populares en la página principal?
 - ¿Cómo se mide la popularidad de un club? ¿Con el número de miembros?
 - ¿Habrá un sistema de rotación de estos clubes?
- ¿Cómo se definirá el acceso a los clubes privados: solo por contraseña o también usando el modo de invitación por link?
- ¿El sistema de votación para decidir el próximo libro será anónimo o visible para todos los miembros del club?
- ¿Qué tanto podrán los usuarios personalizar su perfil? ¿Qué información podrán añadir?
- ¿Existirá solo un administrador por club o podrían haber varios?
- ¿Cómo podría abandonar un miembro un club si este es el administrador y no existe más de uno? ¿Debe asignar un nuevo administrador antes de abandonar el club?
- ¿Cómo se gestionará el espacio de debate para evitar contenido inapropiado o fuera de contexto?

2. Clasificación MoSCoW:

Recordando, el método MoSCoW clasifica los requerimientos en:

1. Must Have (Imprescindibles): Necesarios para que el sistema funcione.
2. Should Have (Importantes): Aportan valor significativo pero no son críticos.
3. Could Have (Deseables): Mejoran la experiencia pero no son esenciales.

4. Won't Have (No incluir ahora): No se implementarán en esta versión.

Requerimientos Clasificados

a. Must Have:

- i. Registro de nuevos usuarios.
- ii. Inicio de sesión para usuarios registrados.
- iii. Listado de clubes populares en la página principal.
- iv. Permitir a los usuarios unirse a clubes.
- v. Permitir a los usuarios abandonar clubes.
- vi. Configuración del libro actual a leer en el club.
- vii. Espacios de debate.
- viii. División de roles, usuario y administrador.

b. Should Have:

- ix. Almacenamiento seguro de contraseñas (Cifrado).
- x. Visualización de detalles del club (antes de unirse).
- xi. Permitir a los usuarios crear un nuevo club.
- xii. Eliminación del club (solo por el administrador del club).
- xiii. Edición de información del club (solo por el administrador del club).

c. Could Have:

- xiv. Recuperación de contraseña mediante correo electrónico.
- xv. Barra de búsqueda para encontrar clubes.
- xvi. Visualización detallada de la información del club (después de unirse)
- xvii. Sistema de votación para decidir el próximo libro a leer.
- xviii. Notificaciones.

- xix. Crear y editar temas de discusión (administrador del club).
- xx. Visualización y edición del perfil personal.
- xxi. Configuración de clubes privados con contraseñas.

d. Won't Have:

- xxii. Sugerencias de clubes basadas en las preferencias del usuario.
- xxiii. Indicador de progreso de lectura.
- xxiv. Aprobar temas propuestos por miembros (administrador del club).
- xxv. Visualización del historial de libros leídos en el club.
- xxvi. Mostrar un historial de libros leídos o géneros favoritos en el perfil.

2. Estimación

Para el proceso de estimación se utilizará la metodología “Planning Poker” basada en la sucesión de Fibonacci, haciendo uso de la herramienta <https://planningpokeronline.com/>.

Tras haber analizado los resultados obtenidos en el “Planning Poker” y haber comunicado las razones por las cuales habíamos dado cada una de las votaciones, este fue el consenso al que se llegó:

Must Have:

1. Registro de nuevos usuarios.
 - a. **Tiempo estimado:** 1 día.
 - b. **Complejidad:** Baja. (Es un proceso bastante usual)
2. Inicio de sesión para usuarios registrados.
 - a. **Tiempo estimado:** 1 día.
 - b. **Complejidad:** Baja. (También es un proceso bastante usual)
3. Listado de clubes populares en la página principal.
 - a. **Tiempo estimado:** 1.7 días.
 - b. **Complejidad:** Media. (Se requiere la adición de los clubes a la base de datos y una IU más definida).
4. Permitir a los usuarios unirse a clubes.
 - a. **Tiempo estimado:** 1 día.
 - b. **Complejidad:** Baja. (No requiere demasiada lógica en la base de datos)
5. Permitir a los usuarios abandonar clubes.
 - a. **Tiempo estimado:** 1 día.
 - b. **Complejidad:** Baja. (Tampoco requiere demasiada lógica en la base de datos).
6. Configuración del libro actual a leer en el club.
 - a. **Tiempo estimado:** 1 día.
 - b. **Complejidad:** Baja. (Es una modificación pequeña de la UI del club).

7. Espacios de debate.
 - a. **Tiempo estimado:** 2.7 días.
 - b. **Complejidad:** Alta. (Debido a que requiere implementar un sistema de chat tanto en front como en back)
8. División de roles, usuario y administrador.
 - a. **Tiempo estimado:** 2 días.
 - b. **Complejidad:** Media. (Se requiere bastante lógica de permisos, y UI con opciones diferentes).

Should Have:

9. Almacenamiento seguro de contraseñas (Cifrado).
 - a. **Tiempo estimado:** 1.3 días.
 - b. **Complejidad:** Baja. (Debido a que tenemos un integrante con experiencia en el área).
10. Visualización de detalles del club (antes de unirse).
 - a. **Tiempo estimado:** 1 día.
 - b. **Complejidad:** Baja. (No requiere demasiada lógica en la consulta a la base de datos).
11. Permitir a los usuarios crear un nuevo club.
 - a. **Tiempo estimado:** 2 días.
 - b. **Complejidad:** Media. (Debido a que se debe implementar un nuevo rol al usuario, y hacer una modificaciones en la base de datos).
12. Eliminación del club (solo por el administrador del club).
 - a. **Tiempo estimado:** 1 día.
 - b. **Complejidad:** Baja. (No requiere demasiada lógica en la base de datos).
13. Edición de información del club (solo por el administrador del club).
 - a. **Tiempo estimado:** 1 día.
 - b. **Complejidad:** Baja. (No es una modificación muy grande en la base de datos).

Could Have:

14. Recuperación de contraseña mediante correo electrónico.

- a. **Tiempo estimado:** 2.7 días.
- b. **Complejidad:** Alta. (Debido a que requiere buscar un sistema para el soporte de emails).

15. Barra de búsqueda para encontrar clubes.

- a. **Tiempo estimado:** 1.7 días.
- b. **Complejidad:** Media. (Se debe crear varias consultas diferentes en la base de datos.)

16. Visualización detallada de la información del club (después de unirse)

- a. **Tiempo estimado:** 1 día.
- b. **Complejidad:** Baja. (No requiere demasiada lógica en la consulta a la base de datos).

17. Sistema de votación para decidir el próximo libro a leer.

- a. **Tiempo estimado:** 2 días.
- b. **Complejidad:** Media. (Es un elemento con el que varios usuarios interaccionan, y es importante en la experiencia de estos).

18. Notificaciones.

- a. **Tiempo estimado:** 2.7 días.
- b. **Complejidad:** Alta. (Debido a que requiere notificaciones por email y no se tienen conocimientos técnicos sobre esto).

19. Crear y editar temas de discusión (administrador del club).

- a. **Tiempo estimado:** 2.7 días.
- b. **Complejidad:** Alta. (Debido a que requiere implementar un sistema de chat más dinámico).

20. Visualización y edición del perfil personal.

- a. **Tiempo estimado:** 1.3 días.

- b. **Complejidad:** Baja. (No requiere demasiada lógica en la base de datos y en las consultas a esta).

21. Configuración de clubes privados con contraseñas.

- a. **Tiempo estimado:** 1.3 días.
- b. **Complejidad:** Baja. (No requiere mucha lógica de programación, ni en la base de datos).

Won't Have:

22. Sugerencias de clubes basadas en las preferencias del usuario.

- a. **Tiempo estimado:** 3.7 días.
- b. **Complejidad:** Alta. (Requiere algoritmos de recomendación, lógica difícil).

23. Indicador de progreso de lectura.

- a. **Tiempo estimado:** 1 día.
- b. **Complejidad:** Baja. (Debido a que solo es hacer una modificación en la base de datos y UI).

24. Aprobar temas propuestos por miembros (administrador del club).

- a. **Tiempo estimado:** 1.3 días.
- b. **Complejidad:** Baja. (No es complejo de implementar en la base de datos, un poco más complejo en UI).

25. Visualización del historial de libros leídos en el club.

- a. **Tiempo estimado:** 1.7 días.
- b. **Complejidad:** Media. (Requiere implementar trazabilidad y consultas en la base de datos).

26. Mostrar un historial de libros leídos o géneros favoritos en el perfil.

- a. **Tiempo estimado:** 2 días.
- b. **Complejidad:** Media. (Parecido a un campo sobre mí, no requiere demasiada complejidad de implementación).

A continuación, se presentan dos gráficos que ayudan a visualizar el análisis realizado.

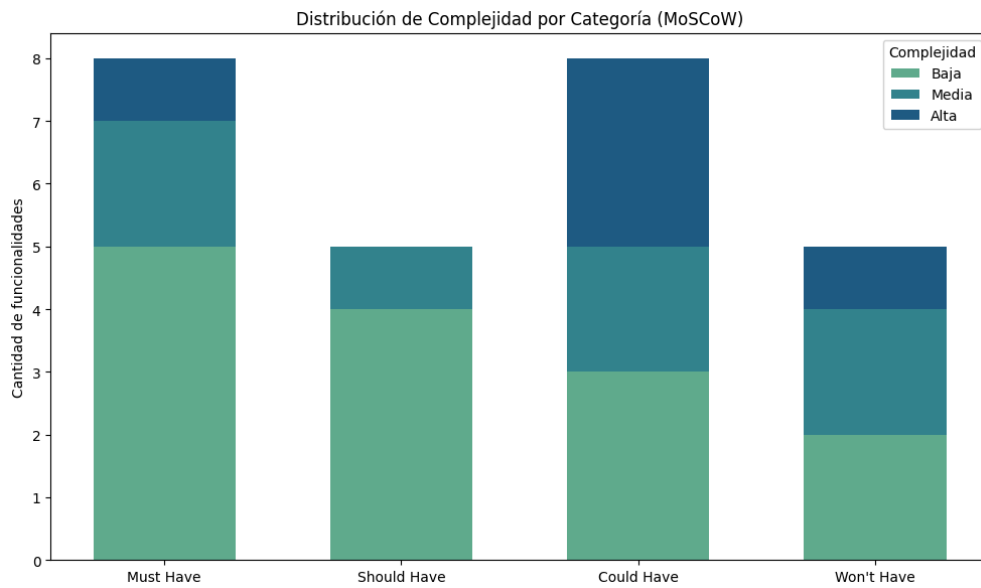


Figura 1: Gráfico de barras que muestra la 'Distribución de Complejidad por Categoría' según el método MoSCoW, lo que permite observar la asignación de complejidad técnica a cada funcionalidad.

Distribución del tiempo dedicado por categoría (MoSCoW)

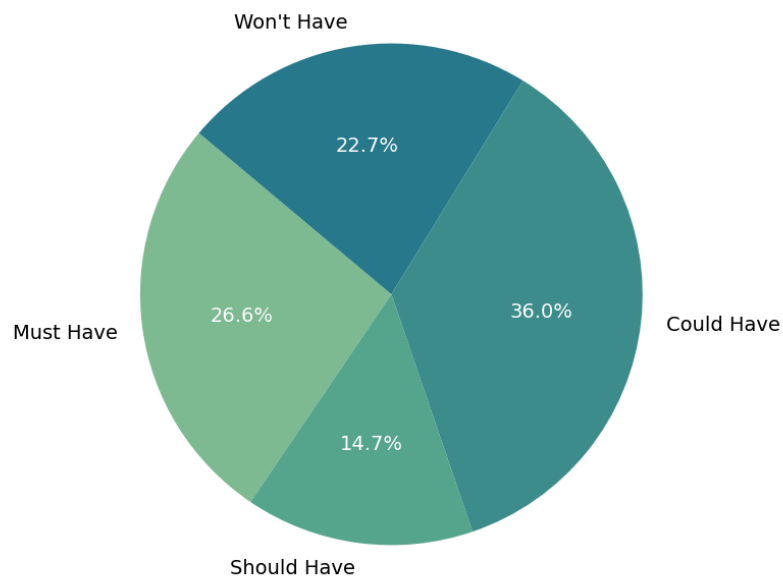


Figura 2: Gráfico circular titulado 'Distribución del Tiempo Dedicado por Categoría (MoSCoW)', que refleja cómo se ha distribuido el esfuerzo temporal entre las diferentes funcionalidades del proyecto.

Análisis de Gestión de Software

1. Tiempo

El desarrollo del proyecto se dividirá en las siguientes etapas:

- **Diseño (2 semanas)**
- **Desarrollo (1 semana)**
- **Testing (1 semanas)**
- **Documentación (1 semana)**

Total estimado: 4 semanas

Para ordenar el tiempo del desarrollo se usó un diagrama de Gantt donde se separó las tareas por integrantes y se definieron intervalos de tiempo para cada tarea.

[5. Diagrama De Gantt.pdf - Google Drive](#)

2. Costo

Para esta estimación, se consideraron los siguientes aspectos:

- **Jornada laboral:** Los empleados trabajarán a tiempo completo. Dado que el proyecto se finalizará antes del 15 de julio de 2025, se tomará en cuenta lo estipulado por la Ley 2101 de 2021, que establece una jornada laboral máxima legal de 46 horas semanales.
- **Modalidad de trabajo:** Dado que el trabajo será remoto, no se contempla el auxilio de transporte.
- **Salario base:**
 - Se utilizará el salario mínimo establecido para el año 2025, conforme al artículo 1 del Decreto 1572 de 2024, que establece un salario mínimo de **\$1.423.500 pesos mensuales**, con vigencia a partir del 1 de enero de 2025. A continuación, se presenta el decreto completo: [Decreto 1572 de 2024 - Gestor Normativo - Función Pública](#).
 - De acuerdo con la página "talent.com", el salario medio para un desarrollador junior es de **\$2.500.000** (dos millones quinientos mil pesos). Por lo tanto, se considerará que a cada desarrollador se le asignan 2 salarios mínimos legales vigentes, lo que resulta en un salario de **\$2.847.000** (dos millones ochocientos cuarenta y siete mil pesos) por cada desarrollador. A

continuación encuentra la página: [Salario medio para Desarrollador Junior en Colombia 2025](#)

- **Cálculo de costos laborales:** El cálculo del costo de los empleados se realizará utilizando la calculadora laboral oficial presentada por el Ministerio del Trabajo. A continuación, se presenta el decreto completo: [MinTrabajo Calculadora Laboral](#)
 - Se obtiene que para el empleador, el costo de contratar a un desarrollador por un mes, es de **\$3.739.686**.

PROVISIÓN MENSUAL EN NÓMINA		
Provisiones Mensuales de Nómina: Las prestaciones sociales se deben provisionar cada vez que se liquida la nómina, estas tienen como finalidad causar mensualmente los gastos correspondientes a las prestaciones sociales de los trabajadores. Al realizar la liquidación definitiva de contrato de trabajo se pueden presentar tres situaciones diferentes: i) Que el valor acumulado de las provisiones por concepto de prestaciones sea igual al valor de la liquidación. En este caso se debe realizar ajuste de provisión. ii) Que el valor acumulado por concepto de provisiones sea inferior. En este caso se debe realizar ajuste de provisión. iii) Que el valor acumulado por concepto de provisiones sea igual al valor determinado en la liquidación del contrato de trabajo.		
Salario	2.847.000	?
Transporte	0	?
PRESTACIONES SOCIALES		
Cesantías	237.250	?
Intereses sobre cesantías	28.470	?
Primas	237.250	?
Vacaciones	118.625	?
APORTES A LA SEGURIDAD SOCIAL		
Pensiones (AFP)	0	?
Salud (EPS)	0	?
Riesgos Laborales (ARL)	14.861	?
PARAFISCALES		
Caja de compensación familiar	113.880	?
ICBF	85.410	?
SENA	56.940	?
TOTAL	3.739.686	?

Dado a que se cuenta con cuatro desarrolladores, se obtiene que el costo total por los 4 desarrolladores es de: **\$14.958.744**.

Para el presupuesto de despliegue se tomará en cuenta el backend en Java con Spring Boot, el frontend en React.js, y bases de datos MongoDB y MySQL. Esta estimación asume un uso moderado de recursos y tráfico.

Se hace el cálculo para el despliegue en aws.

Servicios y Recursos Estimados:

Instancia de Computación (Servidor Backend):

- **Tipo:** Instancia t3.medium (2 vCPU, 4 GB RAM)
- **Uso:** 730 horas/mes (24/7)
- **Costo estimado:** \$37 USD/mes

Instancia de Computación (Servidor Frontend):

- **Tipo:** Instancia t3.small (2 vCPU, 2 GB RAM)
- **Uso:** 730 horas/mes (24/7)
- **Costo estimado:** \$18 USD/mes

Base de Datos (MongoDB o MySQL):

- **Servicio:** Amazon RDS db.t3.micro (1 vCPU, 1 GB RAM)
- **Almacenamiento:** 20 GB
- **Costo estimado:** \$15 USD/mes

Almacenamiento de Objetos (para imágenes y otros archivos estáticos):

- **Servicio:** Amazon S3
- **Almacenamiento:** 50 GB
- **Transferencia de datos:** 50 GB
- **Costo estimado:** \$10 USD/mes

Transferencia de Datos Saliente:

Asumiendo 100 GB de transferencia de datos saliente

- **Costo estimado:** \$9 USD/mes

Costo Total Estimado: Aproximadamente \$89 USD/mes

A conversión de hoy (31/01/2025): **\$374.000** pesos colombianos

Lo que nos da un total de presupuesto para el proyecto de **\$15.332.744** pesos colombianos.

3. Alcance

Dado el tiempo limitado para el desarrollo del proyecto, es fundamental enfocarnos en entregar un producto funcional y estable dentro del plazo establecido. Las funcionalidades clasificadas como **Must Have** representan el núcleo esencial de la plataforma, permitiendo a los usuarios registrarse, unirse y participar en clubes de lectura, y a los administradores gestionar los clubes de manera básica.

Incluir funcionalidades adicionales de las categorías **Should Have** y **Could Have** implicaría un aumento en la complejidad del desarrollo, lo que podría comprometer la calidad, estabilidad y entrega del producto dentro del tiempo estipulado. Al centrarnos en el **Must Have**, garantizamos que la aplicación cumpla con su propósito principal y tenga una base sólida para futuras mejoras en iteraciones posteriores.

Adjuntos

- [1] Propuestas Realizadas:  1. Lluvia de Ideas Propuestas.pdf .
- [2] Selección de Propuestas:  2. Selección Propuestas .
- [3] Transcripción de la Videollamada: [3. Transcripción.txt](#).
- [4] Mockups Iniciales:  4. Mockups Iniciales.png .
- [5] Diagrama de Gantt:  5. Diagrama De Gantt.pdf