



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

– **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de
Telecomunicación

Creación de una comunidad literaria en línea: Desarrollo de
una red social de libros

Trabajo Fin de Grado

Grado en Tecnología Digital y Multimedia

AUTOR/A: Lagarda Pérez, Irene

Tutor/a: Arce Vila, Pau

CURSO ACADÉMICO: 2023/2024



Título

Creación de una comunidad literaria en línea: Desarrollo de una red social de libros.

Resumen

Desarrollo de una aplicación web para una red social de libros utilizando el conjunto de tecnologías MERN (MongoDB, Express, React, Node.js). La aplicación permite a los usuarios llevar un registro de los libros leídos, calificarlos y seguir a amigos y/u otros lectores. El objetivo es crear una plataforma interactiva y social para los amantes de la lectura, donde puedan compartir recomendaciones y descubrir nuevos libros basados en las calificaciones de otros usuarios.

Palabras clave

- Comunidad literaria.
- Red social de libros.
- Aplicación web.
- Tecnologías MERN
- Registro de libros leídos.
- Calificaciones de libros.
- Seguir amigos.
- Descubrimiento de libros.
- Interactividad social.
- Recomendaciones de lectura.



Títol

Creació d'una comunitat literària en línia: Desenvolupament d'una xarxa social de llibres.

Resume

Desenvolupament d'una aplicació web per a una xarxa social de llibres utilitzant el conjunt de tecnologies MERN (*MongoDB, Express, React, Node.js*). L'aplicació permet als usuaris portar un registre dels llibres llegits, valorar-los i seguir amics i/o altres lector. L'objectiu és crear una plataforma interactiva i social per als amants de la lectura, on puguem compartir recomanacions i descobrir nous llibres basats en les valoracions d'altres usuaris.

Paraules clau

- Comunitat literària
- Xarxa social de llibres.
- Aplicació web.
- Tecnologies MERN
- Registre de llibres llegits.
- Qualificacions de llibres.
- Seguir amics.
- Descobriment de llibres.
- Interactivitat social.
- Recomanacions de lectura.



Title

Creation of an online literary community: Development of a social network for books.

Abstract

Development of a web application for a book social network using the MERN technology stack (*MongoDB, Express, React, Node.js*). The application allows users to keep track of the books they have read, rate them, and follow friends and/or other readers. The goal is to create an interactive and social platform for book lovers, where they can share recommendations and discover new books based on the ratings of other users.

Key words

- Literary community.
- Book social network.
- Web application.
- MERN technologies.
- Record of books read.
- Book ratings.
- Follow friends.
- Book discovery.
- Social interactivity.
- Reading recommendations.

RESUMEN EJECUTIVO

La memoria del TFG del Grado en Tecnología Digital y Multimedia debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la tecnologías digitales y multimedia

CONCEPT (ABET)	CONCEPTO (traducción)	¿Cumple? (S/N)	¿Dónde? (páginas)
1. IDENTIFY:	1. IDENTIFICAR:		
1.1. Problem statement and opportunity	1.1. Planteamiento del problema y oportunidad	S	1-2, 4-5
1.2. Constraints (standards, codes, needs, requirements & specifications)	1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	S	2, 6-8
1.3. Setting of goals	1.3. Establecimiento de objetivos	S	2
2. FORMULATE:	2. FORMULAR:		
2.1. Creative solution generation (analysis)	2.1. Generación de soluciones creativas (análisis)	S	6-10
2.2. Evaluation of multiple solutions and decision-making (synthesis)	2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	S	3-4, 8-38
3. SOLVE:	3. RESOLVER:		
3.1. Fulfilment of goals	3.1. Evaluación del cumplimiento de objetivos	S	39-44
3.2. Overall impact and significance (contributions and practical recommendations)	3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	S	44



Índice

Capítulo 1.	Introducción	1
1.1	Motivación.....	1
1.2	Objetivos.....	2
1.3	Metodología.....	2
Capítulo 2.	Estado del arte.....	3
2.1	Servicios similares	3
2.1.1	Wattpad	3
2.1.2	Goodreads.....	3
2.1.3	LibraryThing	3
2.2	Tecnologías similares	3
2.2.1	Front-end	3
2.2.2	Back-end.....	4
2.2.3	Base de datos	4
2.3	Visión de mercado	4
Capítulo 3.	Desarrollo del proyecto.....	6
3.1	Análisis de requisitos	6
3.1.1	Requisitos funcionales.....	6
3.1.2	Requisitos no funcionales.....	7
3.2	Casos de uso	8
3.3	Diseño.....	11
3.3.1	Arquitectura.....	11
Capítulo 4.	Implementación.....	15
4.1	Front-end.....	15
4.1.1	HTML.....	15
4.1.2	CSS	16
4.1.3	React y JavaScript	18
4.2	Back-end.....	28
4.2.1	Iniciación del servidor	29
4.2.2	Iniciar sesión.....	29
4.2.3	Perfil	30



4.2.4	Iniciar lectura.....	31
4.2.5	Feed	35
4.2.6	Biblioteca.....	36
4.2.7	Barra de navegación	37
4.3	Base de datos	38
Capítulo 5.	Resultado final	39
Capítulo 6.	Conclusión	44
Capítulo 7.	Bibliografía	45

Capítulo 1. Introducción

1.1 Motivación

La idea para el proyecto surgió por mi madre y su amor por la lectura. En los últimos tiempos, he visto cómo las *booktokers*, *influencers* literarias en TikTok, comparten opiniones y sugerencias sobre sus lecturas y tienen un cuaderno físico donde registran todas ellas.

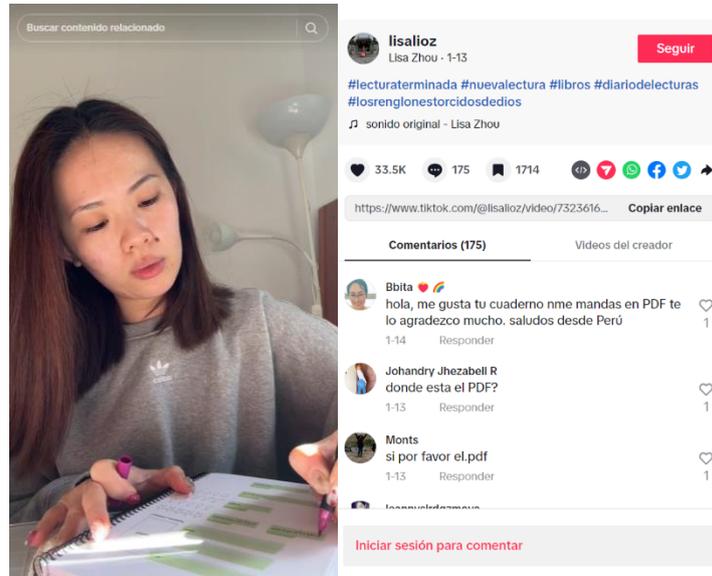


Figura 1. Diario de lecturas de @lisalioz [1]

Un ejemplo claro es el de la *influencer* Lisa Zhou [1], ilustrada en la Figura 1. Al finalizar una lectura, Lisa dedica tiempo y documenta, en un cuaderno, las características principales del libro como el título y autor/a, pero también sus pensamientos y opiniones. Además, en la imagen, se muestra cómo está utilizando el cuaderno y cómo una persona comenta dónde puede encontrarlo para utilizarlo.

Este tipo de videos me inspiraron a querer llevar el registro de libros de mi madre, ya que por tiempo ella no puede hacerlo. Sin embargo, comprar un cuaderno no estaba en mis opciones, quería que fuese más personal y único por lo que quise crearlo yo de manera digital. Me inspiré en varias páginas de cuadernos, Figura 2, que establecían un diseño simple y minimalista.

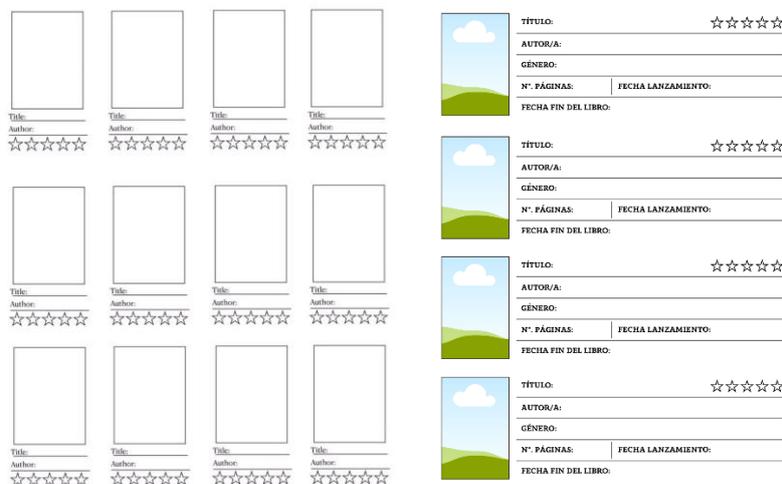


Figura 2. Páginas de cuadernos de lectura

Al empezar a realizarlo, pensé que podría ser una necesidad para otros lectores, el llevar un control detallado de sus lecturas y valoraciones de manera digital, sin necesidad de comprar un cuaderno o perder tiempo. Reflexioné sobre la idea y pensé: ¿por qué no crear una red social para los amantes de la lectura?

1.2 Objetivos

El objetivo principal de este proyecto es crear una comunidad literaria para los amantes de la lectura, permitiéndose explorar el mundo literario de una manera más moderna.

Actualmente las nuevas generaciones tienen predilección a las nuevas tecnologías, como son los teléfonos de última generación, tabletas, videojuegos, etc. Perdiendo de esta manera la oportunidad para iniciarse en el mundo de la lectura. No obstante, los lectores sí se han amoldado a estas tecnologías leyendo no sólo en formato físico sino en digital.

Por ello, al ser una red social, podría interesar a jóvenes por simple curiosidad, y a los lectores habituales a llevar su propio registro y conocer opiniones de otros lectores para nuevas lecturas.

Personalmente, este proyecto tiene como objetivo la superación personal. En el Grado de Tecnología Digital y Multimedia, se han explicado diferentes lenguajes de programación y hacer un proyecto que englobe la mayoría de las asignaturas me satisface y me enorgullece como persona. El primer paso que di en la carrera a las páginas web fue en segundo con la asignatura de tecnologías web, donde no reflejé todo lo que había aprendido y me frustró. No obstante, seguí aprendiendo y evolucionando hasta el punto de querer hacer un proyecto de estas características.

Estoy segura de que esta plataforma no sólo cumplirá con una necesidad personal, sino que, además, creo que podría ser un proyecto con salida comercial al mercado teniendo un impacto positivo y enriquecedor en el mundo de la lectura.

1.3 Metodología

Para el desarrollo de esta aplicación web, se seguirá una metodología estructurada que se divide en varias fases clave.

Primeramente, se realizará el análisis DAFO (Debilidades, Amenazas, Fortalezas, Oportunidades) para valorar las salidas al mercado. A continuación, se analizarán los requisitos y los casos de uso. En esta fase se identificarán y documentarán las necesidades y expectativas de los usuarios, así como los requisitos del sistema. La información generada será de utilidad para la siguiente fase, el diseño. En esta etapa se expondrá el diseño, incluyendo la arquitectura, la base de datos y la web. Se definirán las estructuras utilizadas y el modelo de flujo.

Finalmente, se realizará la implementación, donde se hablará de la estructura de archivos utilizados.

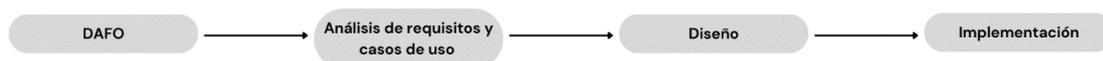


Figura 3. Metodología

Capítulo 2. Estado del arte

2.1 Servicios similares

Existen varias plataformas y servicios con funcionalidades semejantes a las descritas en este proyecto:

2.1.1 *Wattpad*

Wattpad es una plataforma en línea de lectura y escritura. Se trata de una red social donde los autores comparten sus creaciones y los lectores pueden votar las historias y comentar en cada capítulo o incluso en cada frase que haya dentro de él. Sin embargo, no se puede llevar el registro de libros externos a la aplicación que hayas leído.

2.1.2 *Goodreads*

Goodreads es una red social para lectores que permite realizar un seguimiento de sus lecturas, calificar libros, recibir recomendaciones y participar en grupos de difusión. No obstante, esta red social está destinada principalmente a personas de habla inglesa. La gran diferencia con mi proyecto radica en un enfoque más personal y versátil, tanto de sus libros en curso y las futuras lecturas, como el diseño y la interfaz completamente diferentes.

En Goodreads, los libros se almacenan en una carpeta en el perfil del usuario. Además, para acceder a la aplicación es obligatorio registrarse, aunque no se desee almacenar ningún libro. En cambio, en este proyecto se da prioridad a los libros almacenados, presentándose en orden cronológico, con información relevante y la puntuación que el usuario ha otorgado. Asimismo, se pueden explorar los libros de la biblioteca y elegir la siguiente lectura sin necesidad de registrarse.

2.1.3 *LibraryThing*

LibraryThing es un sitio web en forma de comunidad virtual que se encarga de categorizar libros, películas, música y otros medios. Los usuarios pueden publicar reseñas, descripciones y otra información de acceso público sobre una obra. En este caso, la atención no se centra únicamente en los libros y en sus lecturas, sino en la clasificación de las obras.

2.2 Tecnologías similares

Para la realización de un proyecto o desarrollo full-stack se pueden utilizar diferentes lenguajes de programación. A continuación, se presentan las tecnologías consideradas y la justificación de las elecciones para este proyecto.

2.2.1 *Front-end*

Para la parte de la interfaz de usuario del sitio web, los lenguajes más utilizados son Angular, Vue.js o React. En este caso se utilizará React por varias razones:

- React es una biblioteca de JavaScript, no un framework completo, lo que la hace más ligera y flexible.
- Se integra fácilmente con bibliotecas de terceros, lo que enriquece el proceso de desarrollo.
- Tiene una gran cantidad de recursos disponibles lo que facilita el soporte y la resolución de problemas.



2.2.2 *Back-end*

Para la parte del software del servidor se puede utilizar diferentes lenguajes de programación como Python, Ruby on Rails, PHP y Node.js con Express. En este caso, se ha escogido Express como framework y Node.js como entorno de ejecución. He escogido estos lenguajes por las siguientes razones:

- Node.js y Express son conocidos por su alto rendimiento y eficiencia.
- Utilizan JavaScript por lo que facilita la coherencia y el manejo del código.
- A través de npm (Node Package Manager), hay una gran cantidad de módulos disponibles que simplifican la integración de funcionalidades.

2.2.3 *Base de datos*

Para la base de datos se pueden utilizar opciones como MySQL, MongoDB, Casandra, PostgreSQL, entre otras. En mi caso, he escogido MongoDB, una base de datos NOSQL, no relacional, por varias razones:

- Almacena datos en formato de documentos JSON, lo que permite manejar datos de manera flexible y eficiente.
- Se integra de manera nativa con Node.js lo que mejora la eficiencia del desarrollo.
- Ofrece un rendimiento para operaciones de lectura y escritura rápidas, esenciales para este proyecto.

Para resumir, el uso del stack de tecnologías MERN (MongoDB, Express, React, Node.js), al estar todas basadas en el mismo lenguaje Javascript, facilita el desarrollo y ofrece flexibilidad, escalabilidad y rendimiento. Esto aumenta la eficiencia y garantiza que pueda adaptarse a futuras necesidades. Cabe destacar que estos lenguajes de programación que se han sido utilizados son aprendidos en las siguientes asignaturas:

- HTML, CSS y JavaScript → Tecnologías Web
- React, MongoDB y Express → Frameworks

2.3 **Visión de mercado**

Este proyecto tiene un gran potencial de salida al mercado, ya que no existen redes sociales literarias enfocadas exclusivamente en una audiencia de habla hispana. Asimismo, ninguna aplicación en la actualidad permite almacenar los libros leídos de una forma tan accesible y sencilla. La aplicación brinda al usuario la posibilidad de llevar un registro cronológico de sus lecturas, que contiene toda la información relevante y sus calificaciones de forma intuitiva.

A diferencia de otras plataformas, mi idea es permitir a los usuarios explorar la biblioteca sin necesidad de registrarse, lo cual probablemente atraerá a un mayor número de usuarios.

Para evaluar el verdadero potencial de mi red social en el mercado es esencial realizar un análisis DAFO. Esto permite identificar las debilidades y fortalezas internas, como las amenazas y oportunidades externas que podrían influir en el desarrollo y éxito.



Figura 4. Análisis DAFO

DEBILIDADES:

Al existir otros servicios con funcionalidades similares como Wattpat, Goodreads y LibraryThing, representa un gran desafío competitivo por la captación de usuarios en el mercado. A pesar de que el proyecto se centra en el habla hispana, las plataformas mencionadas anteriormente, tienen una posición dominante en otros idiomas. Sin embargo, esta diversidad lingüística puede presentar una gran oportunidad para diferenciarse y captar a usuarios que buscan preferencias literarias y culturales de habla hispana.

AMENAZAS:

Para asegurar el éxito de una nueva plataforma, como es el caso, puede resultar complicado convencer a los usuarios a adoptar la aplicación. Por otro lado, mantener la red social activa y con contenido relevante para que la comunidad esté activa será crucial para el éxito a largo plazo. Para ello se deberán proporcionar herramientas robustas e incentivar la participación mediante valoraciones personalizadas.

FORTALEZAS:

El bloque de tecnología MERN hace que el entorno sea robusto y escalable. La aplicación permite llevar un registro exhaustivo de los libros leídos por cada usuario, así como calificarlos y seguir a amigos lectores. Además, promueve la participación comunitaria en torno a la literatura.

La interacción de los usuarios es enriquecedora para los amantes de la lectura ya que promueve la exploración de nuevos géneros literarios y el descubrimiento de nuevos libros y autores. Asimismo, que el diseño sea intuitivo es fundamental. Se prioriza la experiencia del usuario al proporcionar una navegación sencilla y permite seleccionar lecturas sin necesidad de un registro obligatorio. Esto provoca un amplio alcance de audiencia incluidos aquellos que desean observar la web antes de darse de alta como usuario.

OPORTUNIDADES:

En la actualidad, no existen redes sociales literarias diseñadas para una audiencia de habla hispana. Este vacío de mercado es una oportunidad para captar a usuarios que busquen una red social en su propio idioma, y construir una comunidad de lectores hispanohablantes. Además, ofrece un atractivo entorno para un público más joven y tecnológico. Puede captar el interés al ofrecerles una manera moderna para descubrir nuevas lecturas y registrarlas de una manera más dinámica.

Por último, la red social, podrá generar ingresos a través de publicidad. En un futuro, se podrá expandir incluyendo enlaces directos de compra con librerías, facilitando a los usuarios la adquisición de su nueva lectura. Con ello, el negocio podrá ser sostenible a largo plazo.

Capítulo 3. Desarrollo del proyecto

3.1 Análisis de requisitos

Para el desarrollo de la página web, es esencial llevar a cabo un análisis detallado de requisitos. Este análisis permitirá identificar las necesidades y expectativas de los usuarios, así como definir los requisitos funcionales y no funcionales del sistema. A continuación, se presenta un análisis exhaustivo de estos.

3.1.1 Requisitos funcionales

Los requisitos funcionales son la declaración del comportamiento del sistema. Permiten ofrecer a los usuarios una experiencia de uso eficiente, intuitiva y enriquecedora. Al cumplir con estos requisitos, nos aseguramos de que la plataforma cumpla con las necesidades y cree una comunidad literaria activa. A continuación, se detallarán en formato tabla con el nombre de la función, su importancia o prioridad para el desarrollo y la definición.

Nombre	Registrar usuario
Prioridad	Alta
Descripción	Darse de alta como usuario en la web para poder ser partícipe de todas sus funcionalidades.

Tabla 1. Requisito funcional: registrar usuario

Nombre	Iniciar sesión
Prioridad	Alta
Descripción	Permite a los nuevos usuarios, acceder a su perfil con su correo electrónico y contraseña.

Tabla 2. Requisito funcional: iniciar sesión

Nombre	Cerrar sesión
Prioridad	Alta
Descripción	El usuario podrá salir de su cuenta de manera rápida y segura.

Tabla 3. Requisito funcional: cerrar sesión

Nombre	Buscar libro
Prioridad	Alta
Descripción	Acceso de cualquier persona a buscar cualquier libro registrado y valorado en la aplicación.

Tabla 4. Requisito funcional: buscar libro

Nombre	Registrar libro
Prioridad	Alta
Descripción	Permite a los usuarios dados de alta a añadir libros a su registro personal.

Tabla 5. Requisito funcional: registrar libro

Nombre	Seguimiento del libro
Prioridad	Media
Descripción	El usuario registrado, tiene la posibilidad de actualizar el estado de la lectura

	del libro indicando el número de páginas leídas.
--	--

Tabla 6. Requisito funcional: seguimiento del libro

Nombre	Calificar libro
Prioridad	Alta
Descripción	Valorar el libro previamente leído, calificándolo en formato de cinco estrellas pudiendo puntuar en módulos de media estrella.

Tabla 7. Requisito funcional: calificar libro

Nombre	Buscar usuarios
Prioridad	Media
Descripción	Permite a los usuarios registrados buscar a otros usuarios, previamente registrados en la plataforma que llamaremos “amigos”, para agregarlos o quitarlos del <i>feed</i> personal.

Tabla 8. Requisito funcional: buscar usuarios

Nombre	Añadir amigos
Prioridad	Media
Descripción	Función que realiza el usuario registrado para agregar a un amigo a su <i>feed</i> personal.

Tabla 9. Requisito funcional: añadir amigos

Nombre	Quitar amigos
Prioridad	Media
Descripción	Función que realiza el usuario registrado para eliminar a un amigo, previamente añadido, de su <i>feed</i> personal.

Tabla 10. Requisito funcional: quitar amigos

Nombre	Hacer invisible el <i>feed</i>
Prioridad	Media
Descripción	Permite a los usuarios registrados ocultar el <i>feed</i> de su perfil.

Tabla 11. Requisito funcional: hacer invisible el *feed*

Nombre	Hacer visible el <i>feed</i>
Prioridad	Media
Descripción	Permite a los usuarios registrados, clicando en una flecha, mostrar de nuevo el <i>feed</i> personal.

Tabla 12. Requisito funcional: hacer visible el *feed*

3.1.2 Requisitos no funcionales

Los requisitos no funcionales son fundamentales para asegurar que la aplicación ofrezca una experiencia al usuario satisfactoria, robusta y segura. Los siguientes elementos son esenciales para que el proyecto se adapte a las necesidades a medida que la comunidad literaria crezca y evolucione.

1. Escalabilidad → Al realizar las elecciones oportunas en la base de datos y su estructura.

2. Usabilidad → El diseño de la interfaz es intuitivo al utilizar un número reducido de botones, menús, y siempre elementos grandes y gráficos y con imágenes, en lugar de textos.
3. Diseño adaptativo → La página es accesible desde otros dispositivos, Figura 5.
4. Transiciones → Forma de animación que ocurren suavemente en un periodo de tiempo, para que se vea más atractivo.



Figura 5. Diseño adaptativo para móvil y tabletas

3.2 Casos de uso

Para comprender mejor la interacción del usuario con el sistema, se explican en diferentes casos de uso, con ello, se asegura que se cumplan todos los requisitos funcionales y garantiza que los componentes necesarios estén integrados.

Para ello, se exponen los siguientes tres casos de usos que resaltan las capacidades y características de la aplicación. Para definirlo de una manera más visual se empleará una tabla en la que se dividen los siguientes apartados:

5. Actores → usuarios o anónimos que serán de ejemplo en el caso de uso.
6. Precondición → condición que el usuario desea hacer para cumplir el caso de uso.
7. Escenarios → el camino que ha de seguir el usuario para cumplir su deseo.
8. Excepciones → posibles casos dentro de escenarios que se pueden dar para cumplir el caso de uso.
9. Postcondición → la posibilidad de realizar o guiar después de cumplir su caso de uso.

Caso de uso 1: Iniciar una nueva lectura	
Actor	Usuario no registrado.
Precondición	El usuario no tiene cuenta y quiere registrarse para iniciar su nueva lectura.
Escenarios	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accede a la página de Log in 2. Pulsa el botón de crear una cuenta. 3. Rellena el formulario de registro y pulsa el botón de crear cuenta. [Exc1] [Exc2] [Exc3] 4. Accede de nuevo a la página de Log in 5. Cumplimenta los campos de email y contraseña y pulsa el botón de iniciar sesión. [Exc4] [Exc5] 6. Es redirigido a su perfil. 7. Pulsa el botón de iniciar lectura del perfil o de la barra de navegación. 8. Busca su nueva lectura y pulsa el botón de añadir. [Exc6]. El libro se añade al bloque del seguimiento del libro.

	<p>9. Pulsa la portada del libro. 10. Elige en qué formato se va a leer el libro y cuántas páginas tiene. 11. Pulsa los botones de añadir para dar por finalizado el registro de su nueva lectura.</p>
Excepciones	<p>Exc1: Si el nombre de usuario existe, aparece un mensaje “El usuario ya existe” y no se podrá efectuar el registro. Exc2: Si el correo ya se ha registrado, aparece un mensaje “El correo ya ha sido utilizado” y no se podrá efectuar el registro. Exc3: Si las contraseñas no coinciden, aparecerá un mensaje “Las contraseñas no coinciden” y no se podrá efectuar el registro. Exc4: Si se escribe un correo que no está registrado, aparece un mensaje “Usuario no registrado” y no se podrá efectuar el inicio de sesión Exc5: Si se escribe una contraseña diferente a la que se ha registrado el usuario, aparecerá un mensaje “La contraseña es incorrecta” y no se podrá efectuar el inicio de sesión. Exc6: Si el título del libro no se encuentra, saldrá una alerta y deberás registrar el libro pulsando en el botón de registrarlo. Una vez se pulse el botón se tendrá que escribir el título, autor/a género, y copiar la URL de la imagen de la portada del libro que se desee registrar. Finalmente, se pulsa el botón añadir si se ha rellenado todos los campos de manera correcta.</p>
Postcondición	<p>El usuario podrá:</p> <ol style="list-style-type: none"> 1. Registrar más libros si desea leer 2 libros a la vez. 2. Seleccionar la página por la que se queda para saber el porcentaje del libro que lleva leído. 3. Ir al perfil y seguir a otros usuarios. 4. Cerrar sesión.

Tabla 13. Caso de uso 1: Iniciar una nueva lectura

Caso de uso 2: Buscar próxima lectura	
Actor	Anónimo.
Precondición	Buscar su próxima lectura, prestando atención de las calificaciones.
Escenarios	<p>El usuario no registrado:</p> <ol style="list-style-type: none"> 1. Accederá a la pantalla inicial. 2. Seleccionará en la barra de navegación el apartado de biblioteca. Una vez redirigido podrá: <ol style="list-style-type: none"> a. Buscar el título del libro directamente en el buscador y darle al botón para buscarlo [Exc1] b. Buscar todos los libros que se han registrado por orden alfabético → pulsando el botón libros. c. Buscar a un autor que le guste y encontrar los libros registrados de ese autor/a → pulsando el botón autor. Además, se podrá buscar al autor/a por la inicial de su nombre, solamente clicando en la letra adecuada. d. Buscar libros por el género que le apetezca o le guste leer → pulsando el botón género.
Excepciones	Exc1: Si nadie se ha leído el libro buscado, aparecerá un mensaje “EL LIBRO NO ESTÁ REGISTRADO”.
Postcondición	1. Si el usuario no registrado ya sabe el libro que será su próxima lectura, puede registrarse como usuario y registrar su libro.

Tabla 14. Caso de uso 2: Buscar próxima lectura

Caso de uso 3: Libro finalizado y añadir amigo	
Actor	Usuario.
Precondición	Calificar el libro leído y añadir un amigo en el <i>feed</i>
Escenarios	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accede a la página de Log in 2. Cumplimenta los campos de email y contraseña y pulsa el botón de iniciar sesión. [Exc1] [Exc2] 3. Es redirigido a su perfil. 4. Pulsa el botón de iniciar lectura del perfil o de la barra de navegación. 5. Selecciona en las estrellas y mueve el ratón para seleccionar la calificación del libro. 6. Pulsa el botón calificar para efectuar la valoración. 7. Es redirigido al perfil, y se da por finalizada la lectura del libro. 8. Busca en el <i>feed</i> el nombre de un usuario. 9. Selecciona el usuario y pulsa el botón añadir. [Exc3] [Exc4]
Excepciones	<p>Exc1: Si se escribe un correo que no está registrado, aparece un mensaje “Usuario no registrado” y no se podrá efectuar el inicio de sesión.</p> <p>Exc2: Si se escribe una contraseña diferente a la que se ha registrado el usuario, aparecerá un mensaje “La contraseña es incorrecta” y no se podrá efectuar el inicio de sesión.</p> <p>Exc3: Si el usuario no existe, mostrará un mensaje “USUARIO NO ENCONTRADO”</p> <p>Exc4: Si el usuario ya ha sido agregado, mostrará un mensaje “EL USUARIO YA ES TU AMIGO”</p>
Postcondición	<p>El usuario podrá:</p> <ol style="list-style-type: none"> 1. Quitar a su amigo de la misma manera que se ha añadido, pero pulsando el botón de quitar. 2. Iniciar una nueva lectura. 3. Cerrar sesión.

Tabla 15. Caso de uso 3: Libro finalizado y añadir amigo

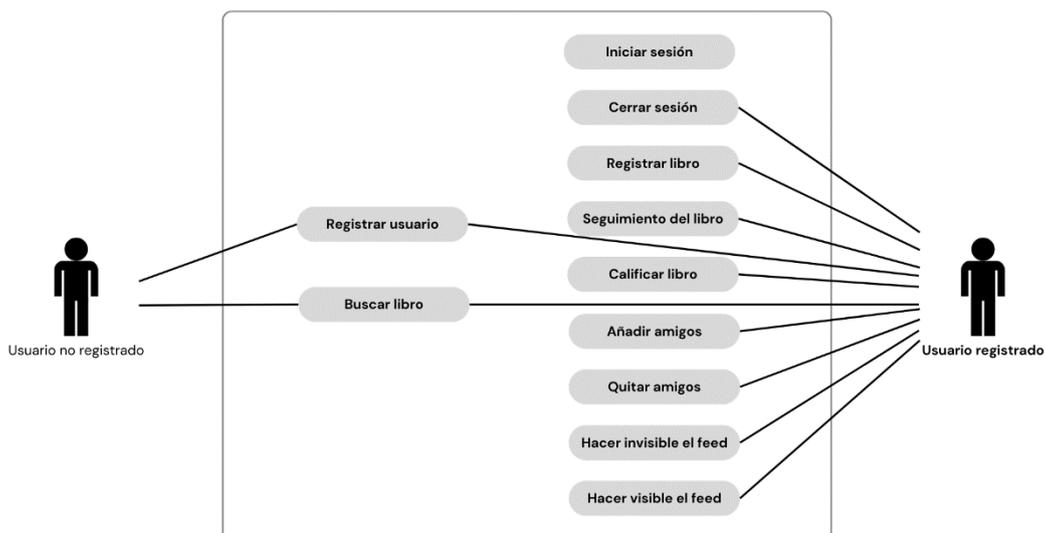


Figura 6. Diagrama de uso

En la Figura 6 podemos observar el diagrama de uso de dos actores de nuestra aplicación, donde se ve que un usuario registrado puede acceder a todas las funcionalidades. En cambio, el usuario no registrado, el de la izquierda, puede acceder a dos de ellas.

3.3 Diseño

3.3.1 Arquitectura

El proyecto sigue una arquitectura de tres capas independientes que se organizan en capa de presentación, de aplicación y de datos. Utiliza un modelo cliente/servidor en el que el cliente realiza peticiones y el servidor le responde. Este sistema se encarga de proteger los datos que se almacenan en el servidor para que un cliente o persona no autorizada no pueda acceder a ellos.

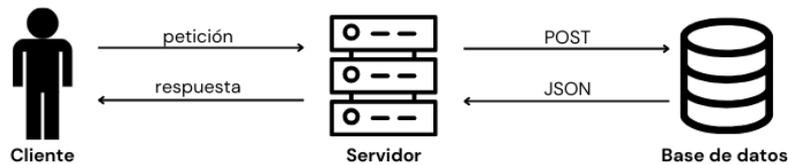


Figura 7. Arquitectura

El modelo de arquitectura es muy similar a *Orion Context Broker* [2], modelo estudiado en cuarto de GTDM en la asignatura de plataformas IoT. Los dos modelos se comunican mediante APIs, realizando peticiones del cliente al servidor o *broker*, y permiten la interoperabilidad entre diferentes sistemas o entornos.

Como observamos en la Figura 7, el cliente, a través de código JavaScript incrustado en la página web, manda una petición a la API del servidor. Antes de enviarle una respuesta, el servidor ha de llamar a la base de datos, en la mayoría de los casos con un POST, con la petición correspondiente, y ésta le responde con los datos en formato JSON. Una vez el servidor obtenga la información necesaria para dar respuesta a la petición del cliente, responde con dichos datos que se muestran de manera visual en el navegador del cliente.

3.3.1.1 Capa de presentación

Esta capa es la parte visible de la red social, es decir, es lo que ve el cliente. Los usuarios interactúan con la plataforma por lo que las funcionalidades clave han de ser accesibles con el mínimo esfuerzo. Para ello, previamente se han diseñado una serie de bocetos que representan la estructura y apariencia de la red social, que servirá de guía visual para luego el desarrollo de la interfaz y se consideren todas las necesidades de los usuarios.

A continuación, se presentarán los bocetos diseñados previamente. En este caso se diseñaron 5 páginas diferentes; página de inicio, inicio sesión, perfil, biblioteca y página de lectura.



Figura 7. Boceto página inicio

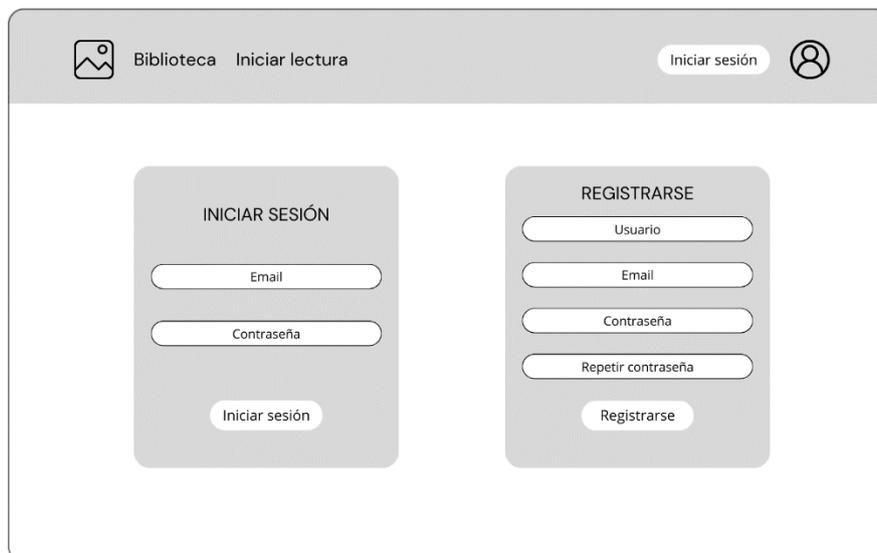


Figura 8. Boceto inicio sesión

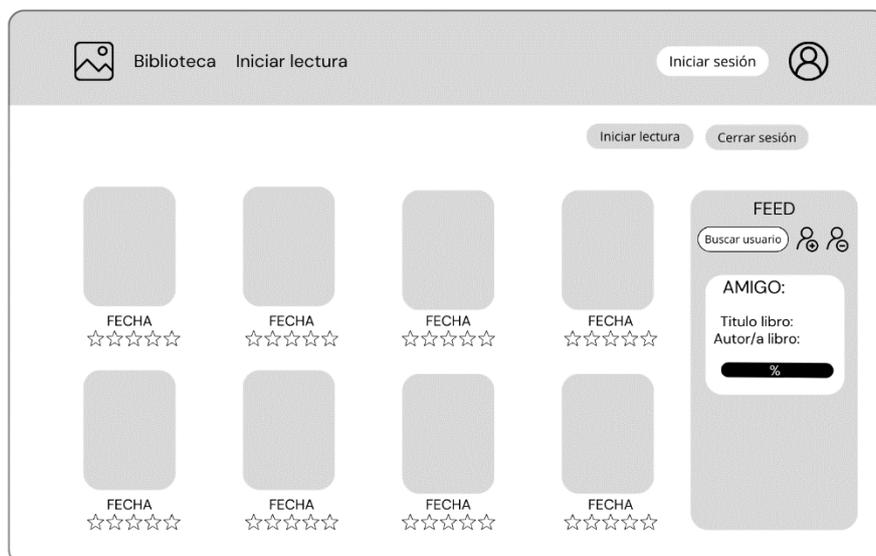


Figura 9. Boceto perfil con *feed*



Figura 10. Boceto perfil sin feed



Figura 11. Boceto biblioteca



Figura 12. Boceto página de lectura

3.3.1.2 Capa de aplicación

La capa de aplicación es el servidor, la parte fundamental para que funciones la red social. Este es el encargado de gestionar la lógica y entender las peticiones del cliente, en este caso de los usuarios.

Para ello, se emplearán distintos middlewares, softwares que actúan como intermediario entre las peticiones y respuestas del cliente y servidor. Su uso es importante para la gestión de autenticación de los usuarios y la validación de los datos.

3.3.1.3 Capa de datos

Esta capa se encarga de almacenar, gestionar y proporcionar la información para hacer posible el funcionamiento de la aplicación. Es fundamental asegurar los datos de los usuarios, para ello se desacopla y sólo el servidor adecuado puede obtener dicha información. A continuación, se mostrarán detalladamente la estructura que sigue la base de datos, en este caso una base NoSQL.

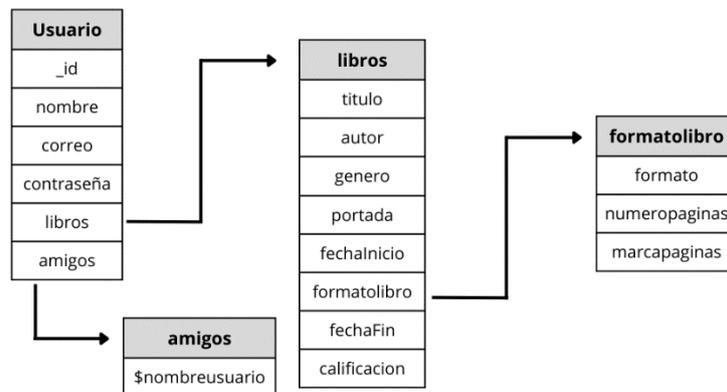


Figura 13. Diseño base de datos

Capítulo 4. Implementación

En este capítulo nos adentraremos en la implementación de nuestra red social de libros. Al desarrollar las especificaciones de requisitos y un boceto en la fase de diseño, es momento de empezar a desarrollarlo.

Antes de empezar se debería instalar Node.js [3], y después instalarse las siguientes librerías con el comando “npm install *paquete*”, siendo *paquete* el contenido de los módulos que se necesitarán para realizar el proyecto, que son los siguientes:

- Express → framework web que permite la utilización de middleware y manejo de solicitudes HTTP.
- Mongodb → interfaz para interactuar con la base de datos MongoDB desde una aplicación Node.js.
- Babel → traducción de jsx (lenguaje de react) a JavaScript.
- Body-parser → middleware para poder coger los valores del cuerpo de la petición.

4.1 Front-end

Para la parte visual de la aplicación se dividirá en los diferentes lenguajes utilizados.

4.1.1 HTML

HTML es un lenguaje de marcado que estructura toda la página web. Además, prácticamente es la base de internet, ya que la navegación web consiste en que cualquier navegador web puede solicitar al servidor web el archivo HTML a través de su localizador o URL, y es el mismo navegador quien lo interpreta y lo muestra al usuario.

En este caso, se ha almacenado en una carpeta *public* todos los archivos que el servidor necesitará para mostrar la página al usuario.

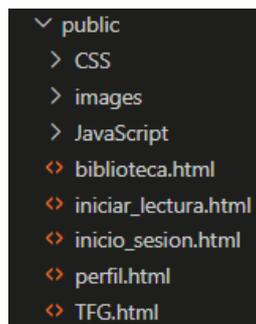


Figura 14. Estructura de ficheros HTML

Como podemos observar en la Figura 14, he creado cinco archivos HTML que coinciden con los cinco bocetos de diseño de la capa de presentación. Cabe destacar que todos los archivos contienen una barra de navegación, que explicaré más adelante, donde se podrá acceder a diferentes páginas dentro del proyecto.

Para empezar, tenemos el fichero “biblioteca.html”, que contiene un buscador en formato *input search* y tres botones al lado, para la búsqueda de libros dentro de la biblioteca. Además, se irán viendo unas fotografías con las portadas de los libros de la base de datos.

El siguiente archivo “iniciar_lectura.html” muestra tres secciones a modo de *fieldset*. En esta página se podrá registrar la nueva lectura, llevar el seguimiento de esa misma y calificar el libro

a gusto de cada usuario. Este fichero es bastante extenso y a continuación se explicarán las tres partes.

4.1.1.1 Seleccionar libro

La primera sección es la selección del libro, es decir el registro de la nueva lectura del usuario. En ella se muestra un buscador a modo de *input search* y un botón al lado, con el que se buscará el título o autor/a del libro para iniciar la lectura.

4.1.1.2 Seguimiento del libro

Esta sección de inicio aparecerá vacía a no ser que se haya seleccionado un libro previamente. Si es el caso, aparecerá la imagen de la portada del libro que se ha registrado, con el título y una barra en la parte inferior con un porcentaje centrado, como se muestra en la Figura 15.



Figura 15. Seguimiento del libro: portada, título y progreso

En este caso, la imagen del libro es un botón para acceder a un formulario y así seleccionar la página por la que se ha quedado el usuario, actuando como un marcapáginas cuando se lee un libro en formato físico. En cambio, la barra es un *div* que a medida que el usuario va cambiando la página, varía su porcentaje.

4.1.1.3 Calificar el libro

Para la tercera sección dentro del archivo “iniciar_lectura.html” se visualizará vacío si no se ha terminado un libro previamente. Si es el caso y se ha dado por acabada una lectura, se visualizará de la misma forma que la Figura 15, pero en vez de una barra aparecen cinco imágenes en forma de estrella y un botón que guarda dicha información en la base de datos del usuario.

Volviendo a la estructura de ficheros de la Figura 14, el siguiente fichero es el de “iniciar_sesión.html”. En él se podrá iniciar sesión y registrarse si el usuario aún no se ha dado de alta. A continuación, se muestra el “perfil.html”. En él se observan dos botones una sección a modo de *fieldset* y a la derecha como una especie de móvil (el *feed*) con un buscador y dos imágenes en las que se observa un usuario con un signo más (+) y otro menos (-), para agregar y quitar a amigos.

Por último, encontramos el archivo llamado “TFG.html” que es la pantalla principal de la red social de libros. Encontraremos un texto dando la bienvenida a todos los clientes/usuarios que entren en la aplicación y una imagen estética y adecuada al proyecto.

4.1.2 CSS

CSS es un lenguaje que se encarga del diseño visual y la disposición de los elementos de la página web. Los archivos utilizados con este lenguaje se almacenan en una carpeta nombrada “CSS” dentro de la carpeta *public* como se observa en la Figura 16.

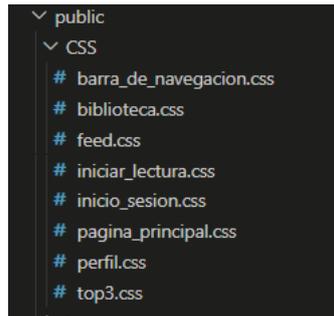


Figura 16. Estructura de ficheros CSS

En este caso, observamos por la Figura 16, ocho ficheros CSS, siendo cinco de ellos el diseño visual de los archivos html explicados anteriormente. No obstante, podemos ver los archivos “barra_de_navegacion.css”, “feed.css” y “top3.css”. Se han creado esos tres más para facilitar el diseño de la página, ya que por ejemplo la barra de navegación se repetirá en todas las páginas web de la aplicación.

El fichero “feed.css” se ha independizado del perfil por si se quiere quitar la visibilidad, sólo habrá que ocultar ese fichero directo.

Por último, el archivo “top3.css” se encarga de la disposición de los libros, tanto en el perfil del usuario como en la biblioteca. Se ha creado para agilizar el proyecto y que todos los libros tengan la misma estructura y diseño.

Por otro lado, utilizar y seguir una paleta de colores adecuada es fundamental para el diseño de cualquier proyecto, ya que los colores influyen en la experiencia del usuario y la percepción general de un sitio. En mi caso, los libros me recuerdan al color marrón, a hojas de papel antiguo y a la estación de otoño. Para mostrar de forma visual la paleta de colores utilizada, he creado una imagen con todos los colores usados con su valor hexadecimal, Figura 17.

#000000
#940B92
#915d45
#bb8e80
#ddb0a1
#F2CEC2
#f7ddd4
#f1e4e0
#ffffff

Figura 17. Paleta de colores

El único color a resaltar es el morado, pero se ha utilizado para avisos o escritos donde se observe de una manera más llamativa y diferencial, así como el progreso de las lecturas de los diferentes amigos dentro del feed.

Cabe añadir que, siguiendo los requisitos de usabilidad y diseño intuitivo, todos los botones tienen una transición con *hover* de tres segundos, en el que se agranda a escala 1.1 o 1.2 y cambian de color. Esto se ha utilizado para que el usuario entienda que está seleccionando el botón antes de clicarlo. Además, se ha utilizado también el método *hover* como transición para la calificación del libro. No obstante, no sólo se ha realizado con CSS, sino que se ha creado mediante condiciones en el JavaScript que se explicará con profundidad en el siguiente apartado.

4.1.3 React y JavaScript

React es una librería que permite crear interfaces de usuario con código abierto, basada en el lenguaje JavaScript. Es altamente funcional debido a su gran capacidad para integrarse con otros elementos. Su sintaxis es JavaScript XML, más conocido como JSX, el cual es una combinación del lenguaje HTML y JavaScript.

En este proyecto, he creado una carpeta nombrada *src* donde se hallarán todos los ficheros JSX. Cuando el servidor se inicie, los archivos de esa carpeta se compilarán y traducirán a formato JS en una carpeta llamada *JavaScript* dentro de *public*, para poder ser utilizados en los ficheros HTML ya mencionados. Para entenderlo mejor, se puede observar la Figura 18, en ella se ve como están nombrados los mismos archivos, pero en formatos diferentes.

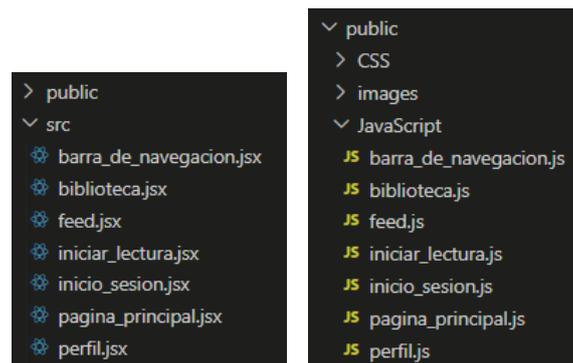


Figura 18. Estructura de ficheros React y JavaScript

En este apartado, se va a explicar a fondo el funcionamiento de los diferentes ficheros. Aunque se seguirá un orden lógico de forma estructurada y las funcionalidades resulten más comprensibles.

4.1.3.1 Barra_de_navegacion.jsx

Este archivo es un *header*, es decir, una cabecera para todos los archivos. Se trata de dos imágenes, una con el logotipo de la aplicación y otra de un usuario, un botón para iniciar sesión y dos enlaces con las páginas de biblioteca y de iniciar lectura.



Figura 19. Barra de navegación



Figura 20. Barra de navegación adaptada con desplegable

Las peculiaridades de este fichero son la adición de otra imagen para un menú desplegable cuando la pantalla se adapte a diferentes dispositivos (observando las Figuras 19 y 20), y la realización de dos condiciones. Estas se han creado para el enlace de iniciar sesión y del perfil del usuario, ya que no se podrán acceder al contenido de ellas si no se ha iniciado sesión previamente. Si se presionan dichas páginas sin haberse identificado, se redirigirá a la página de iniciar sesión.

4.1.3.2 Pagina_principal.jsx

La estructura del archivo jsx, es sencilla, ya que sólo contiene dos textos y una imagen (ver Figura 21).



Figura 21. Página principal

4.1.3.3 Iniciar_sesion.jsx

Este archivo ha sido creado a partir de un código abierto de Rahul Saini - @helpme_coder [4] [5], modificando los colores y el texto en español. Además, se han añadido funciones para conexiones cliente/servidor y eliminado los botones de redes sociales, como se aprecia en las Figuras 22 y 23.

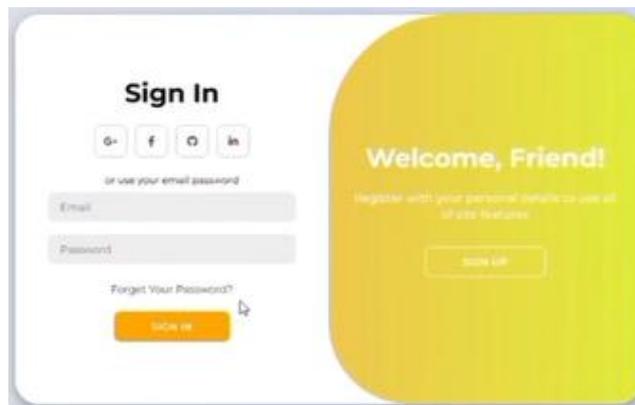


Figura 22. Inicio de sesión código abierto @helpme_coder

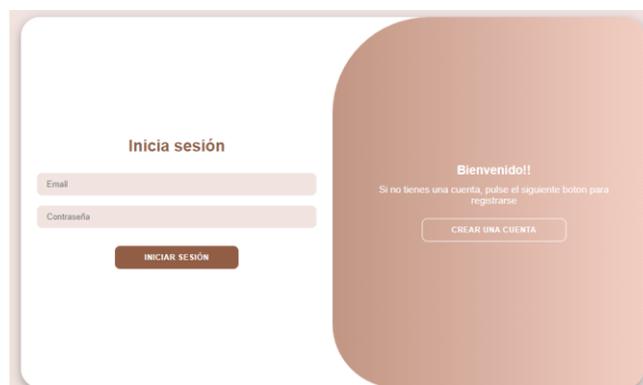


Figura 23. Inicio de sesión propio

Por último, se almacena como variable el nombre de usuario que inicia sesión para utilizarlo en otras páginas.

4.1.3.4 Perfil.jsx

El fichero “perfil.jsx” muestra información única para cada usuario registrado, por lo que se recuperará la variable de usuario para obtener los libros almacenados de ese mismo. En este archivo, se han creado dos botones importantes, uno para redirigir al usuario a iniciar su lectura, y otro para cerrar sesión.

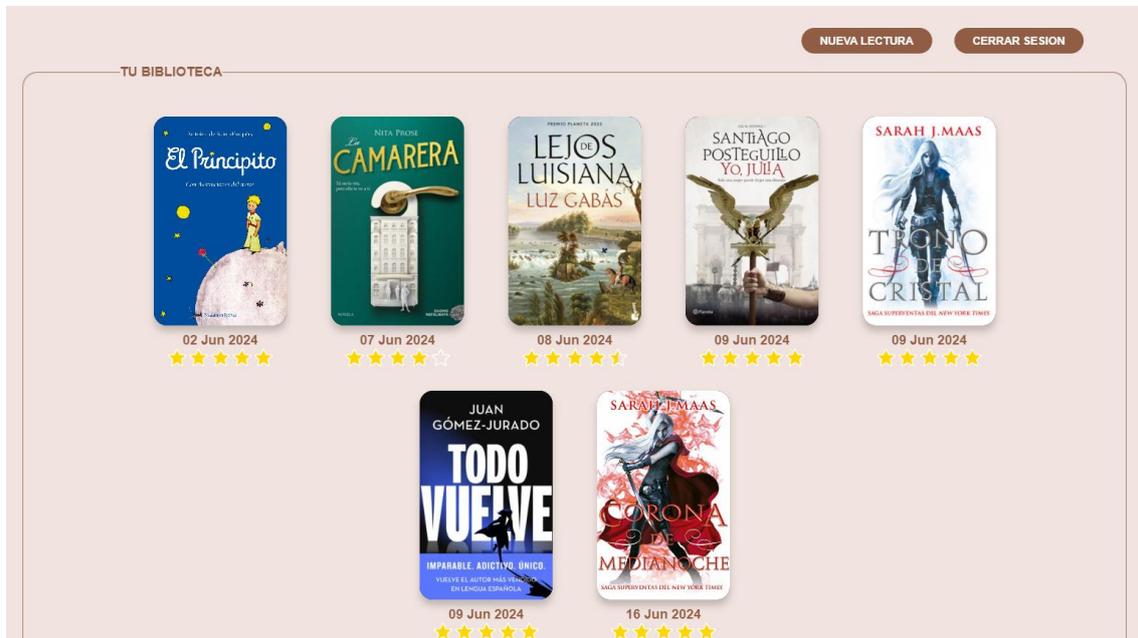


Figura 24. Biblioteca del usuario

Para la parte de la biblioteca de cada usuario, Figura 24, se ha creado un objeto que incluye el título del libro y el nombre del usuario. Este se envía al servidor para obtener toda la información sobre él y devuelve un objeto de datos.

Una vez tenemos la información del libro que hemos leído, utilizamos la URL de la imagen de la portada del libro, la fecha de finalización y el valor de calificación en formato estrellas. El título de cada libro leído y terminado se adjuntará en una variable en forma de *array*, que serán los diferentes libros que se verán en la biblioteca del perfil.

Es importante saber que cada portada de un libro es como una carta. Cuando se pasa por encima de la imagen, se puede observar los datos relevantes de cada libro, como el título, el autor/a, el género y la fecha de inicio y fin (Figura 25).



Figura 25. Información del libro en el perfil

Por último, las estrellas de calificación son cinco imágenes puestas una al lado de la otra. Dependiendo de cada valor, se verán unas estrellas u otras. Este proceso se explicará más con

detalle en el fichero “iniciar_lectura.jsx” en el apartado de calificación del libro (apartado 4.1.3.7.3).

4.1.3.5 Feed.jsx

El *feed* es el archivo que da nombre a este proyecto, es decir, la red social. Este sirve como el centro donde los usuarios pueden interactuar, buscar y ver el progreso de lectura de sus amigos. El funcionamiento del *feed* depende de varias variables esenciales que se almacenan y recuperan desde el almacenamiento local del navegador. Esas son para la visibilidad del archivo, una lista de los usuarios registrados en la plataforma, una lista de amigos para cada usuario y un array inicialmente vacío para almacenar los nombres buscados.



Figura 26. Feed

Como se observa en la Figura 26, se ha creado un buscador, un *input search*, y dos botones con dos imágenes, añadir y quitar amigo. Cabe destacar, que a medida que la variable usada para todos los usuarios registrados en la plataforma sirve como *datalist*, es decir, como un desplegable de opciones, que a medida que se escribe aparecen los distintos usuarios adecuados al escrito. Cuando un usuario escribe el nombre de un amigo y pulsa la imagen de agregar, aparece en formato *fieldset* el usuario introducido, con el libro que se está leyendo y el porcentaje. Esto se ha creado para los amigos que quieren realizar una lectura conjunta saber que estáis leyendo a la par. También puede servir, para saber que lee un amigo tuyo y luego preguntar su reseña o calificación, o simplemente para cotillear como se creó en la aplicación Spotify, como se muestra en la Figura 27.

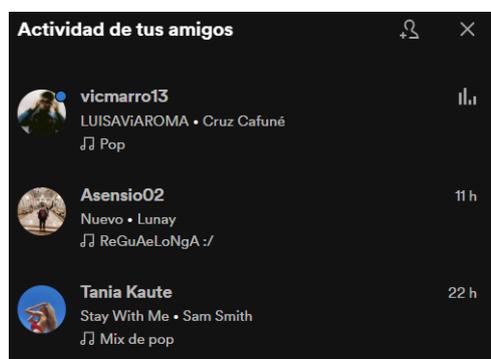


Figura 27. Feed de la aplicación Spotify

Por último, para quitar a un amigo del *feed* porque ya no te interesa ver lo que se está leyendo, sólo hace falta volver a escribir el nombre del usuario y darle clic a la imagen de quitar amigo.

4.1.3.6 Biblioteca.jsx

Este archivo se ha realizado en base al archivo “perfil.jsx”. La gran diferencia, es que las cartas (las portadas de los libros con la información de cada uno) tienen la misma estructura y se van añadiendo conforme un libro ha sido calificado. No obstante, la información de cada libro es diferente. Como podemos ver en la Figura 28, aparece el título y autor del libro, la media de todas las calificaciones, el género literario y el número de lecturas. Es importante para los lectores saber cuántas personas o valoraciones tiene un libro, por lo que, si sólo se pone la media, el usuario no sabe cuántas opiniones tiene. En este caso y como se observa, “El principito” tiene una media de 4.17 siendo leído y calificado por tres personas diferentes. En cambio, “Nosotros en la luna” tiene una media de 5, pero sólo leído por una persona.



Figura 28. Información del libro en la biblioteca

Por otra parte, se puede facilitar la búsqueda de un libro, ya sea directamente buscándolo en el *input search* u ordenando los títulos, los autores o los géneros por orden alfabético. Para indagar el título de un libro u autor, se realiza en forma de *datalist*, esto quiere decir, que se ha creado una variable local con todos los libros y sus autores que a medida que se califica y finaliza una lectura, se añade a la biblioteca. Esto se realiza para facilitar al usuario su búsqueda.

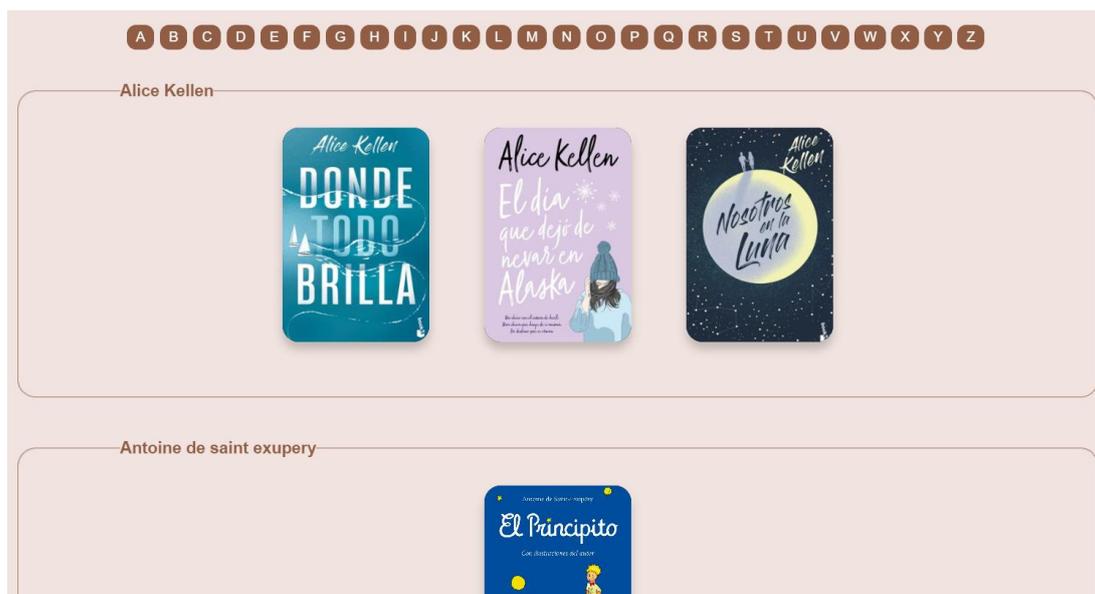


Figura 29. Ordenar por autores

Para ordenar los autores, se ha realizado de una manera más visual y accesible. Como se puede observar en la Figura 29, se han creado veintiséis botones con cada letra del abecedario y una sección, *fieldset* para cada autor/a. Cuando se quiere buscar un autor en concreto, sólo hace falta pulsar el botón con la letra correspondiente y se hallan todos los autores sólo con el nombre cuya inicial es la seleccionada, como se puede ver en la Figura 30 se ha seleccionado el botón de la letra S.

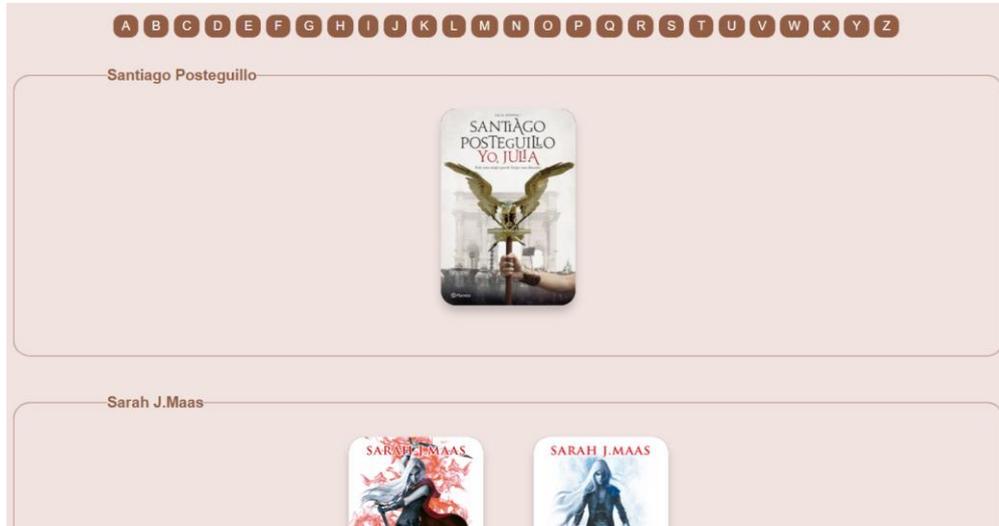


Figura 30. Ordenar autores cuya inicial del nombre empieza por S

Por último, la disposición del orden de los géneros es muy similar a la de los autores, lo único que no existen los botones del abecedario para acceder al género, ya que son muchas menos secciones y se puede deslizar en la misma página.

4.1.3.7 Iniciar_lectura.jsx

Este fichero es una pieza fundamental en la gestión y la experiencia lectora del usuario en la aplicación. Este componente es el más extenso y el que permite empezar un nuevo libro, registrar su progreso y calificarlo. A continuación, se explicará cada una de esas secciones.

4.1.3.7.1 Seleccionar el libro

Para empezar a explicar, es importante saber qué variables se utilizarán. En este caso será el nombre del usuario y los títulos ya leídos y calificados con sus autores. A continuación, como se observa en la Figura 31, y como se ha explicado anteriormente (en el punto 4.1.1.1), se encuentra un buscador en el que el usuario ha de escribir el título del libro para registrarlo. Como podemos observar, a medida que vamos escribiendo aparecen posibles opciones a elegir. Si por algún casual se pulsase el botón de añadir y no existiese el libro en la base de datos, le aparecerá una alerta, como se puede ver en la Figura 32 para registrar el libro.

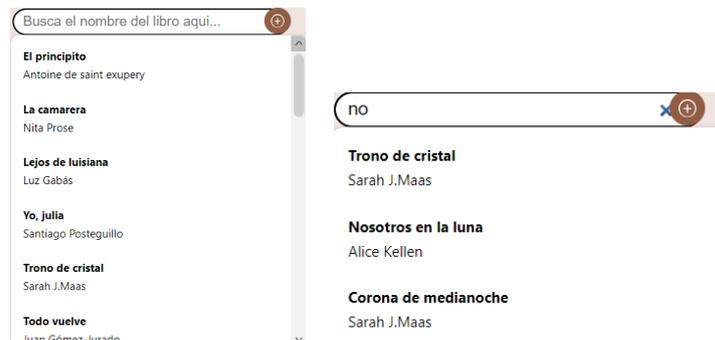


Figura 31. Buscar nueva lectura



Figura 32. Alerta al registrar el libro

Una vez pulsado el botón OK de la Figura 32 o esperando tres segundos a que se cierre sola la alerta, aparecerá un botón para registrar el libro, Figura 33.



Figura 33. Notificación registrar libro no existente

Al apretar dicho botón, aparecerá un formulario a rellenar todos los campos, tal y como aparece en la Figura 34. Los tres primeros campos, tienen una función *onInput*, es decir, cuando se pulse la entrada de texto *input*, que inicializa la primera letra con mayúscula.



no

Título
Título del libro

Autor/a
Nombre del autor o autora

Género
Nombre del autor o autora

Portada
Pega la url de la imagen de la portada del libro

AÑADIR LIBRO

Figura 34. Registrar el libro no existente

Por una parte, los campos de autor/a y género, son *datalists* similares a la Figura 31, pero sólo con los nombres de los autores o sólo con los géneros literarios, dependiendo de cada caso.

Por otro lado, el texto a rellenar de la portada es un enlace de internet. Este tiene que estar en formato adecuado, si no pondrá un mensaje como se ve en la Figura 35.



Portada

<https://th.bing.com/th/id/OIP.991k52kXAmh4uEX8Hum6JwH>

Escriba un correcto url, en este caso no ha sido así

AÑADIR LIBRO

Figura 35 URL de la portada del libro

Para que la URL esté de manera adecuada, ha de buscar en internet el título del libro que se quiera registrar, añadiendo “portada libro”. Una vez seleccionada la imagen, se ha de copiar su

vínculo, no la imagen en sí. Si se ha realizado de manera correcta, aparecerá, como se aprecia en la Figura 36, una previsualización de la portada del libro.



Título
Alas de sangre

Autor/a
Rebeca Yarros

Género
Fantasía

Portada
<https://cdn.kobo.com/book-images/Images/9b675ece-dde6-4>

AÑADIR LIBRO

Figura 36. Previsualización de la portada de un libro no registrado

Al terminar de rellenar todos los campos, se podrá añadir el libro. Al pulsar el botón, se almacena en una variable el título y el autor que se ha registrado y actualiza automáticamente la página donde se podrá observar que el libro ha sido seleccionado como nueva lectura y ha pasado a la sección de seguimiento del libro.

Cabe destacar, que todo el proceso de registrar el libro no será necesario si el libro que se desea leer el usuario ya está en la base de datos, es decir, que ya ha sido leído por otro usuario. Simplemente, dicha persona habrá seleccionado un libro de las opciones del buscador y al pulsar añadir, automáticamente habrá pasado a la siguiente sección.

4.1.3.7.2 Seguimiento del libro

Para este apartado, es muy importante saber que se van a realizar tres formularios, pero todos están previamente ocultos. La actualización automática del registro del libro, la podemos observar en la Figura 37. En este apartado, se pueden tener todos los libros que se desee, puesto que hay lectores, que quieren leer dos libros a la vez, o simplemente, puedes registrar previamente tus libros comprados, para saber cuáles tienes pendiente y así hacer seguimiento de ellos. Cada usuario puede hacer con sus registros y seguimientos lo que se quiera, eso hace que esta red social sea única e íntima. En este caso se observa que un lector ha registrado cuatro libros para llevar su seguimiento.



Figura 37. Seguimiento de varios libros registrados

Como se ha explicado anteriormente (apartado 4.1.1.2) la imagen es un botón que visualiza y oculta un formulario cada vez que es pulsado. Dicho formulario, Figura 38, se ha de rellenar pulsando el *input radio* adecuado al libro. Es decir, si el libro que el usuario va a empezar a leer es en formato bolsillo, seleccionará el botón circular a la izquierda del texto “De bolsillo”. Cuando uno de ellos esté escogido, se abrirá automáticamente otro formulario, como se aprecia en la Figura 39, para escribir el número de páginas que tiene dicho libro. En el caso de que se quiera enviar el formulario sin rellenar el segundo formulario, este no le dejará. Es obligatorio cumplimentar ambos antes de ser emitidos.

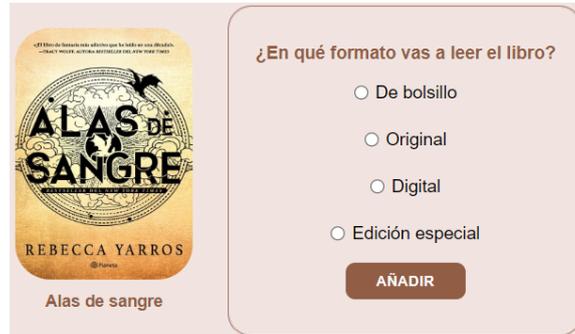


Figura 38. Seguimiento del libro: elección de formato

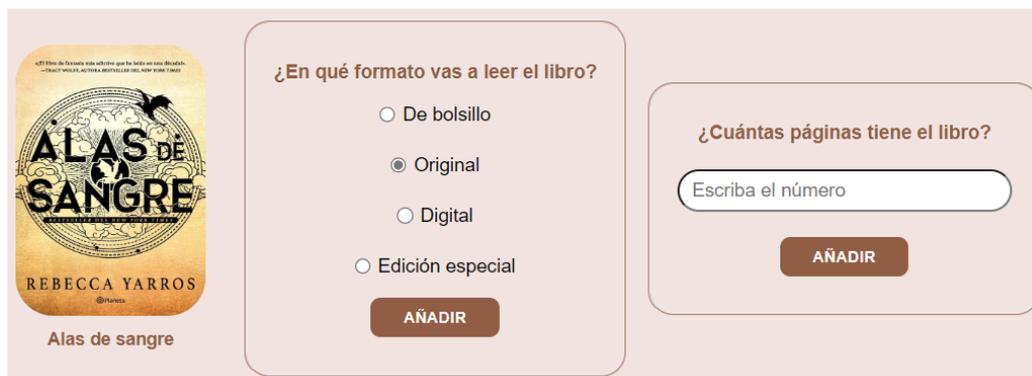


Figura 39. Seguimiento del libro: número de páginas

Una vez enviados ambos, no volverán a visibilizarse ninguno, y aparecerá debajo del título una barra con un porcentaje, ya explicados anteriormente (apartado 4.1.1.2) y mostrada en la Figura 15. Al presionar la portada del libro, aparecerá el tercer formulario, Figura 40. Este se podrá rellenar tantas veces como se quiera, ya que su utilidad es similar a la de un marcapáginas.

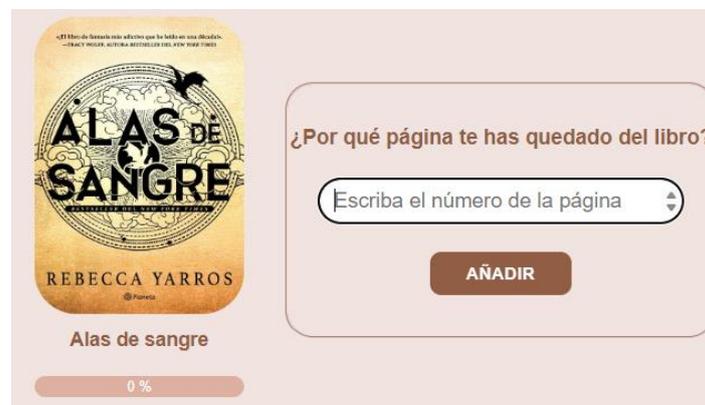


Figura 40. Seguimiento del libro: marcapáginas

Siempre que se añada el número de la página, se calculará el porcentaje del libro leído utilizando la cifra actual entre la cantidad de páginas totales, multiplicado todo ello por cien. No obstante, se realizará el mismo calculo para el ancho de la barra de progreso, pero siendo el

valor total 150 píxeles. Dicho esto, se puede apreciar en la Figura 41 el aumento del progreso del libro.

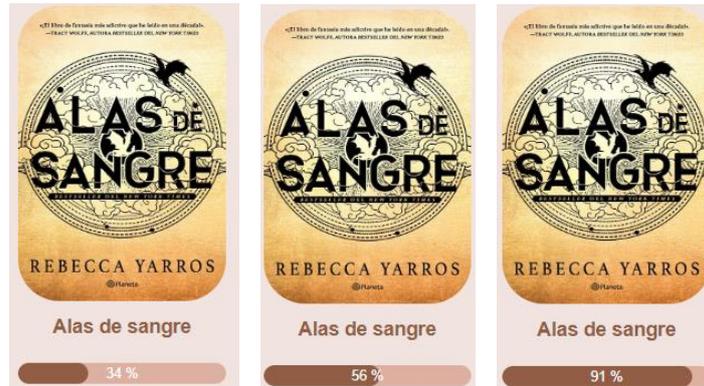


Figura 41. Seguimiento del libro: progreso

Al acabar la lectura, es decir, cuando el valor escrito y enviado del formulario coincida con el número de páginas totales, aparecerá un aviso como aparece en la Figura 42. Cuando desaparezca el aviso, eliminará de la variable local el título y autor/a para que desaparezca de esta sección, pero lo añadirá en otra variable que se utilizará para calificar el libro. Dicho esto, se actualizará la página automáticamente y la portada y el título aparecerá en la siguiente sección.



Figura 42. Notificación finalización de un libro

4.1.3.7.3 Calificar el libro leído

El apartado de calificar el libro es una sección esencial para la evaluación de las lecturas realizadas por los usuarios. En esta sección se permite puntuar cada libro leído proporcionando una referencia valiosa para otros usuarios.



Figura 43. Calificar libro

Para la calificación se utilizan cinco imágenes de estrellas vacías una al lado de otra, como se observa en la Figura 43. En cada una de las estrellas hay dos *inputs radio*, es decir, cada imagen está compuesta de dos botones invisibles, Figura 44. Como se observa cada círculo corresponde a una calificación del 1 al 10, los números impares son mitades de estrella y los pares, estrellas completas.



Figura 44. Calificación formato estrella

Esta función se ha creado con tres fotografías, estrella vacía, media y llena, creadas por mí. Además, se han creado condicionantes para todos los valores de los *inputs radio*. Se ha creado una transición para que el usuario sienta que está moviendo la puntuación y se van llenando o vaciando las estrellas conforme mueve el ratón. Para ello podemos observar la Figura 45. En ella se observa cómo se han pintado las tres primeras estrellas sido que se ha pulsado el tercer *input*. Esto sucede porque el ratón está por encima del sexto.



Figura 45. Calificación formato estrella transición

Es importante saber que al ir cambiando conforme se mueve el ratón por encima, el usuario ha de clicar cuando vea la puntuación elegida. Y finalmente cuando esté todo escogido, hay que enviarlo pulsando el botón calificar, que nos redirigirá automáticamente a la página del perfil del usuario, para que se observe que se ha añadido de manera adecuada.

4.2 Back-end

El desarrollo *back-end* es la parte del desarrollo web, es decir, el servidor, que se centra en cómo funcionan las páginas, manejando la lógica y la integración de las APIs. Para ello se han creado dos ficheros, uno con la instrucción “*npm init*” para iniciar un proyecto y otro que llamaremos “*index*” o “*index.js*” donde estarán todos los *middlewares*.

El archivo para iniciar el programa contiene el nombre de todos los módulos instalados y se han añadido tres comandos que son los siguientes:

- *Compilar* → Traduce todos los archivos de la carpeta *src* a lenguaje JavaScript y los almacena en *public/JavaScript*. (Se ha explicado anteriormente en el apartado 4.1.3 y se puede observar en la Figura 18).
- *Actualizar* → Renueva la carpeta *public* cada vez que se modifique cualquier código dentro de la carpeta *src*
- *Start* → Envía la información para que se compile automáticamente y que acceda al fichero *index.js*.

Ahora profundizaremos en el fichero *index*. En él se necesitará acceder a las librerías *express*, *path* y *body-parser*. Además, se accederá a una función del fichero de la base de datos que se explicará más adelante (punto 4.3).

El fichero utilizará una parte estática del sitio web donde podrá obtener el directorio *public*, y con el *body-parser* podremos obtener los valores dentro del cuerpo del archivo cuando se envía con el método *POST*.

A continuación, se mostrarán diferentes Figuras (de la 46 a la 63) con un diagrama de flujo. Esto se realiza para explicar de forma visual el funcionamiento del servidor con el cliente y la base de datos, para la funcionalidad de cada página. Además, observaremos, como el usuario envía una información, el servidor lo reúne y lo remite la base de datos y esta le responderá en formato *JSON*. El servidor tratará de mostrarle al cliente de forma visual en la página adecuada. Se estructurará por el siguiente orden:

1. Iniciación del servidor.
2. Página de inicio de sesión.
3. Página del perfil del usuario.

4. Página de iniciar lectura.
5. *Feed*.
6. Biblioteca.
7. Barra de navegación.

Cada imagen, responderá con diferentes colores de flecha. Si es de color negro, significa que sólo puede realizar esa petición o respuesta, es decir, sólo hay un camino posible. Si la flecha es de color marrón, la base de datos habrá respondido de manera correcta a la petición y la respuesta del servidor en caso afirmativo. En cambio, si sucede algún error, las flechas serán de color morado.

4.2.1 *Iniciación del servidor*

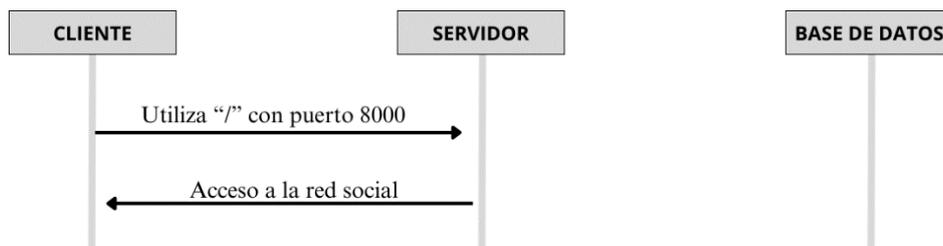


Figura 46. Diagrama de flujo: iniciar el servidor

4.2.2 *Iniciar sesión*

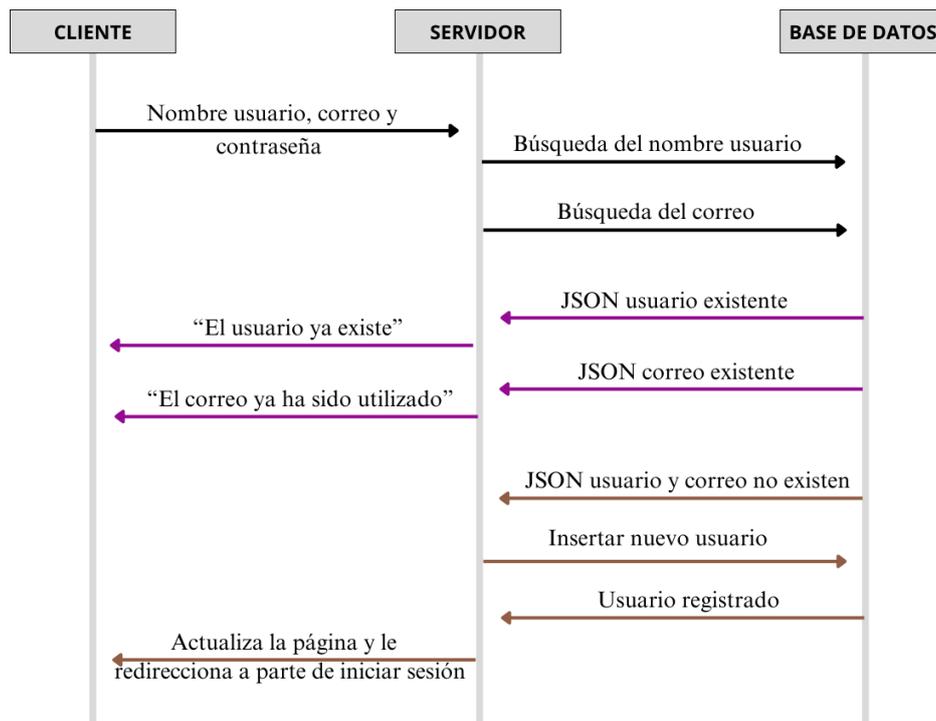


Figura 47. Diagrama de flujo: función registrar usuario

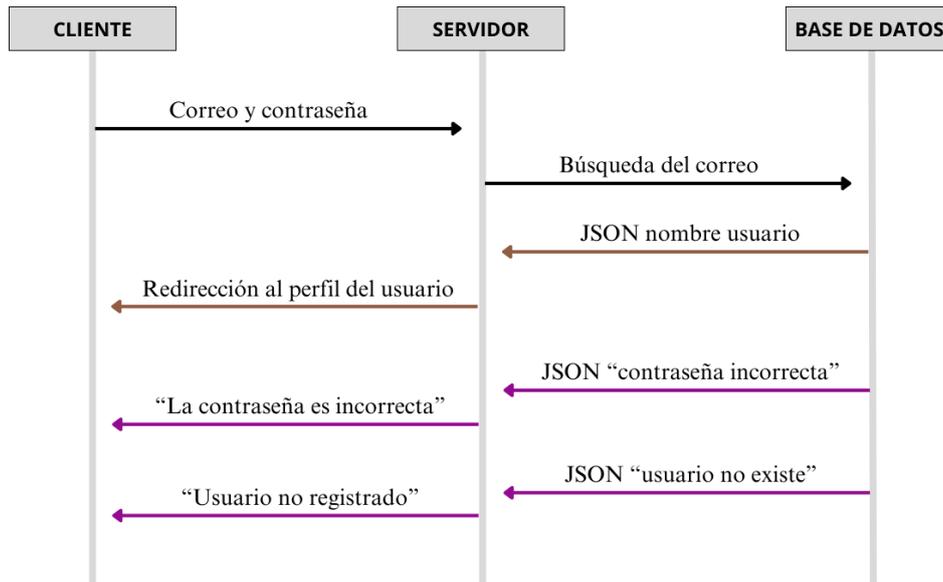


Figura 48. Diagrama de flujo: función de iniciar sesión

4.2.3 Perfil

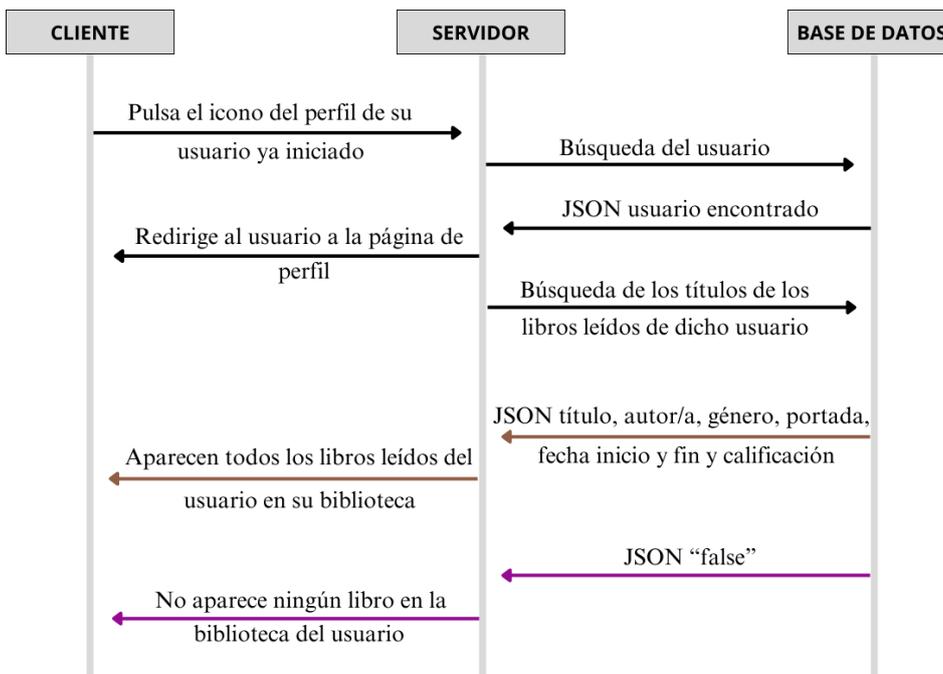


Figura 49. Diagrama de flujo: acceder al perfil



Figura 50. Diagrama de flujo: cierre de sesión

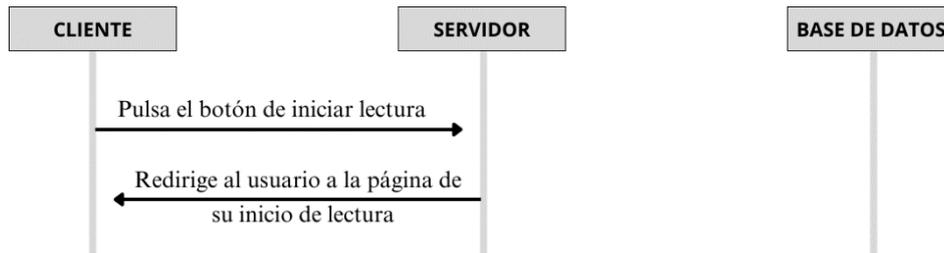


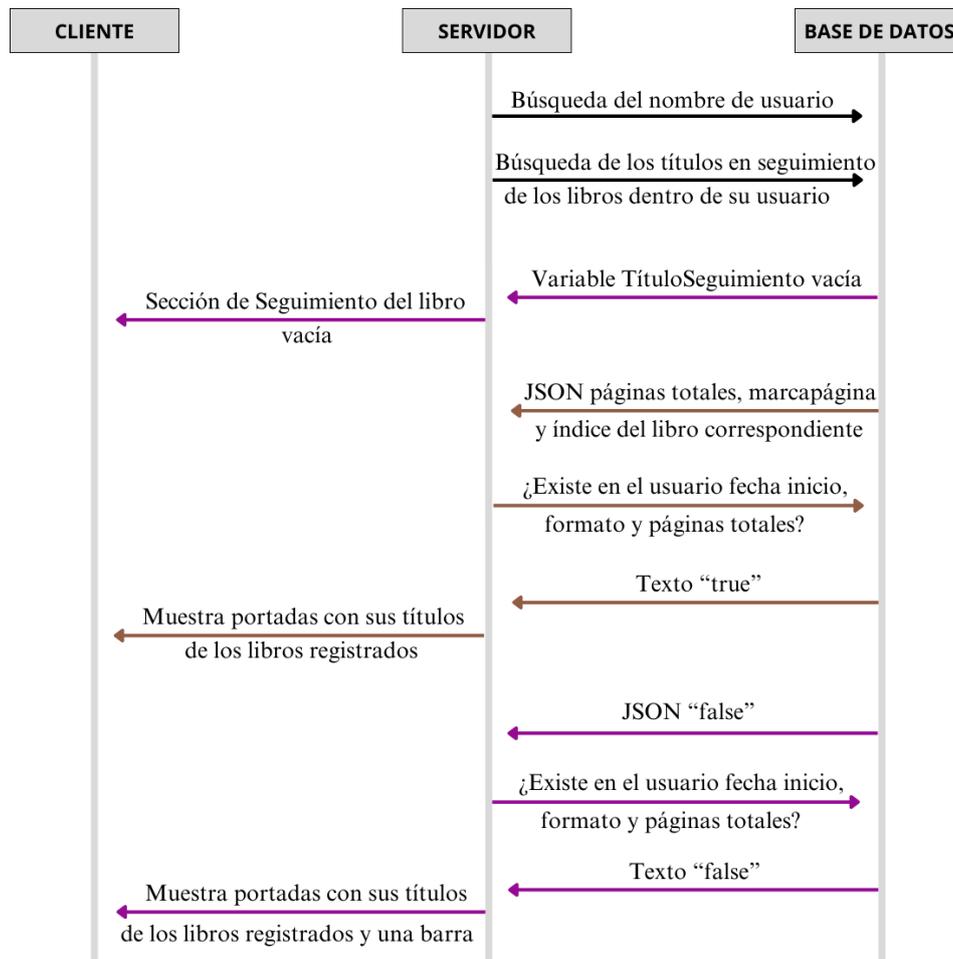
Figura 51. Diagrama de flujo: iniciar lectura

4.2.4 Iniciar lectura

Para la página de iniciar lectura, se van a crear tres variables vacías en formato *array*:

- TítuloSeguimiento → Al finalizar el registro de un libro, el título se añadirá a esta variable. Se utiliza para poder mostrar las portadas y títulos en la sección de seguimiento.
- TítuloCalificar → Cuando un usuario finaliza un libro, se eliminará de la lista TítuloSeguimiento y se añade a esta. Su función es mostrar el título y la imagen de la portada.
- TítuloLibroLeído → Al acabar de valorar un libro, se extraerá de la variable TítuloCalificar dicho título puntuado y se añadirá a esta variable. Su utilidad es almacenar todos los títulos leídos del usuario para utilizarlo en la biblioteca del perfil.

Esta información es importante, ya que cada vez que se inicie la página, se deberán mostrar los libros de las variables TítuloSeguimiento y TítuloCalificar. Si están vacíos significa que no se ha registrado ni está leyendo ningún libro, pero ha calificado los que ha terminado.



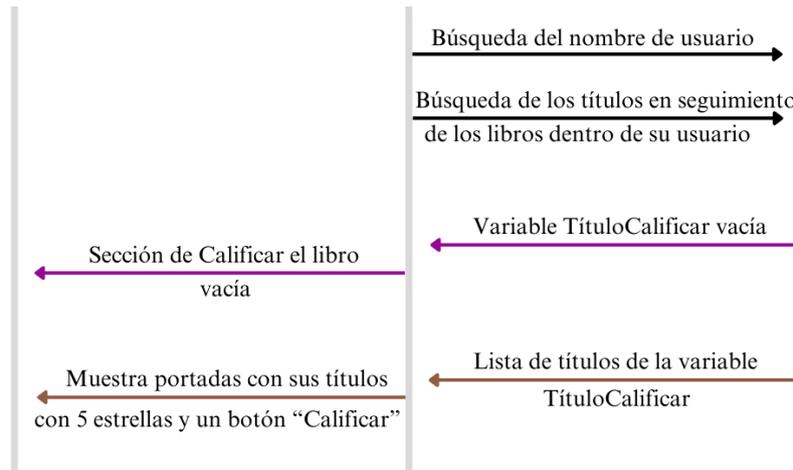
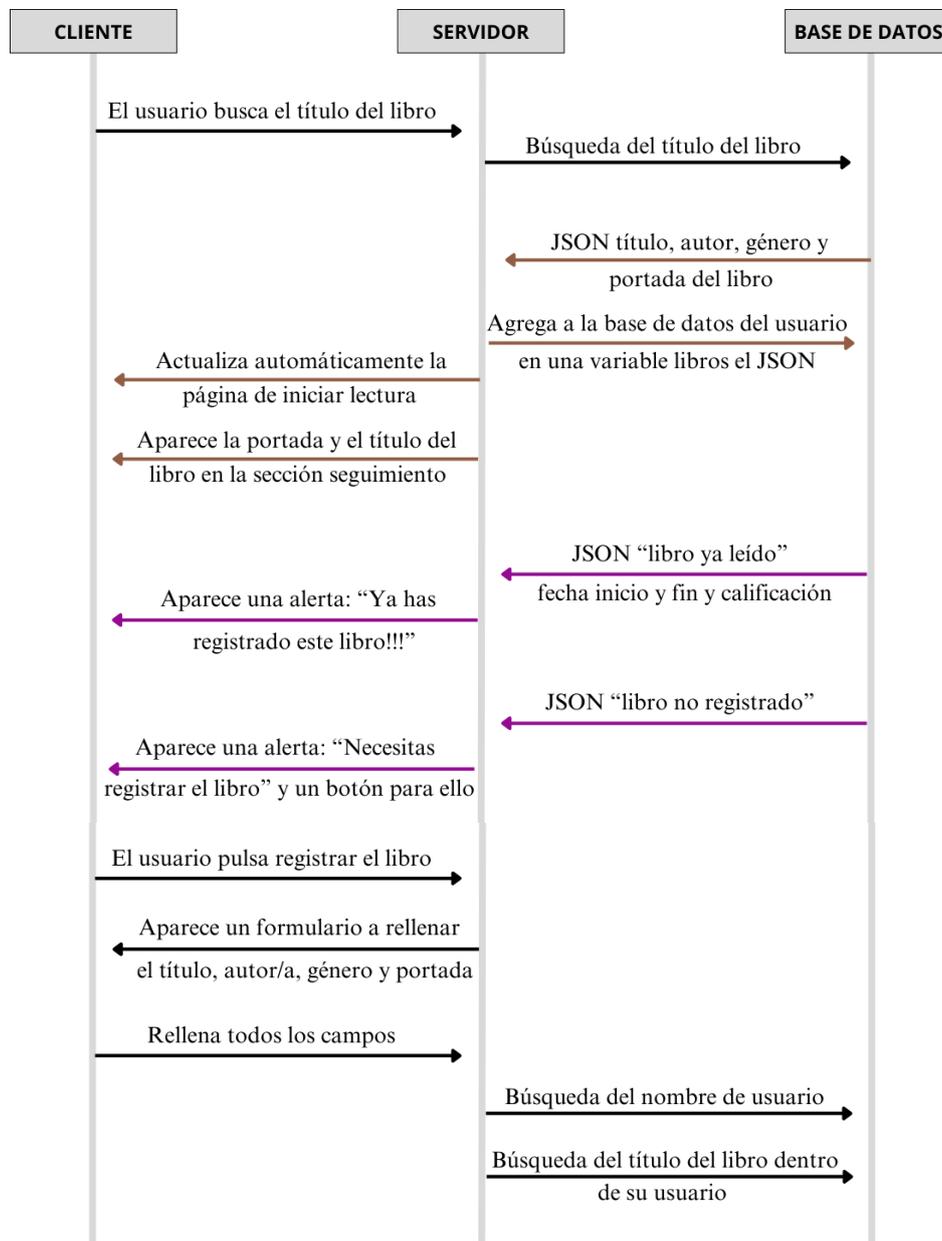


Figura 52. Diagrama de flujo: página iniciar lectura



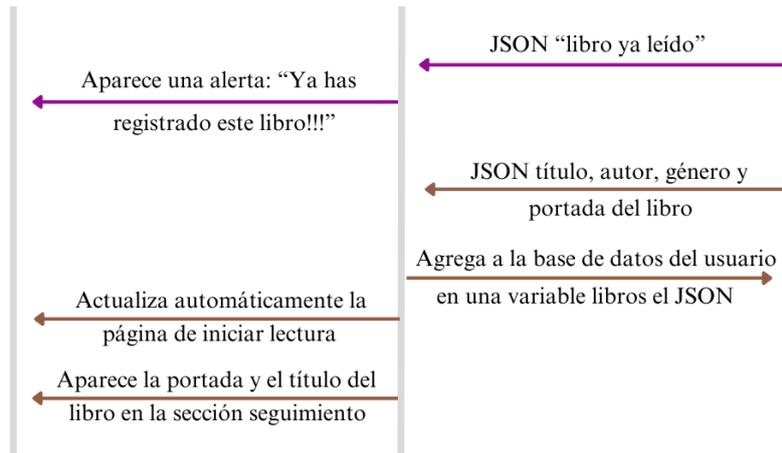
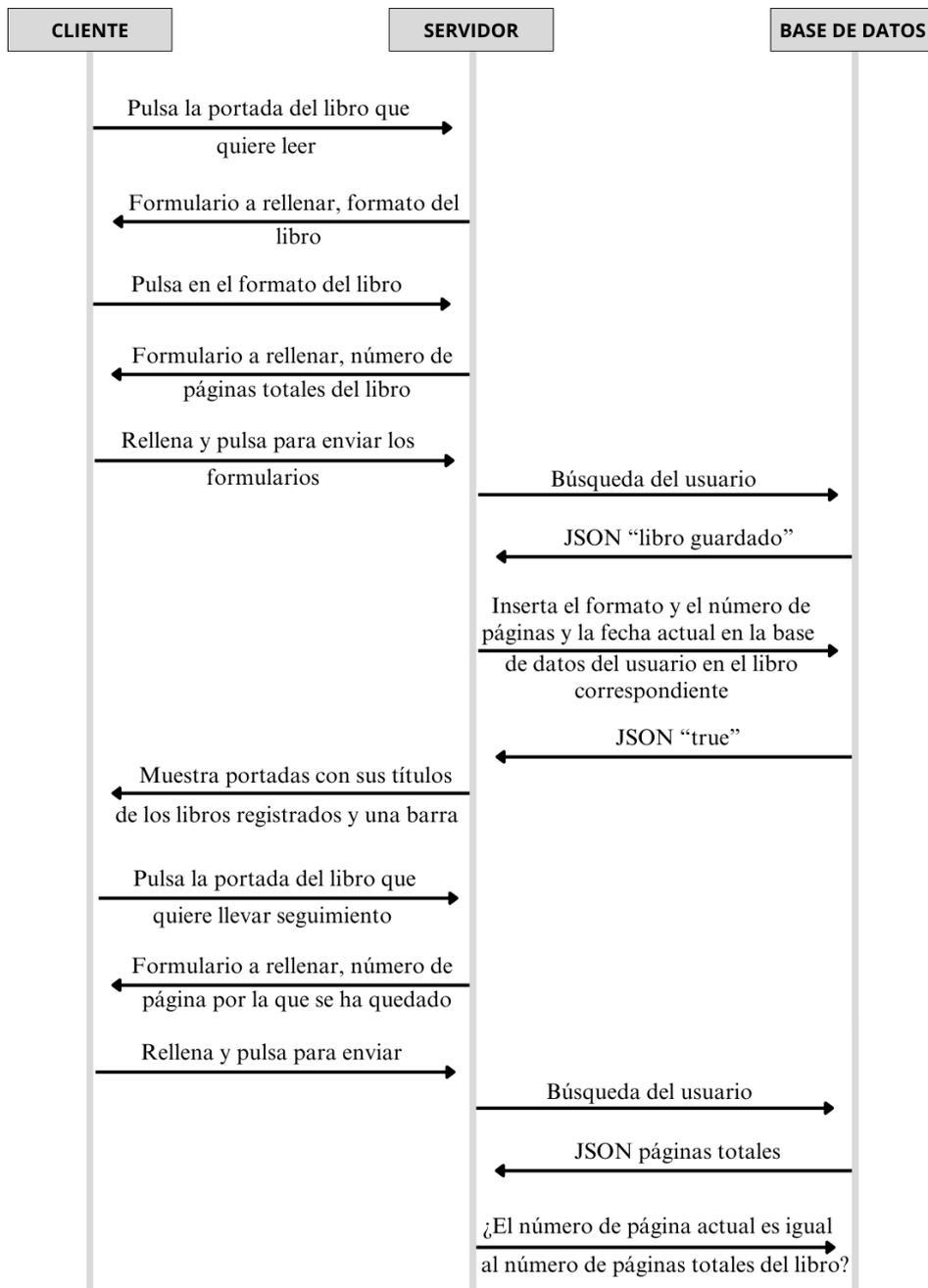


Figura 53. Diagrama de flujo: registrar libro



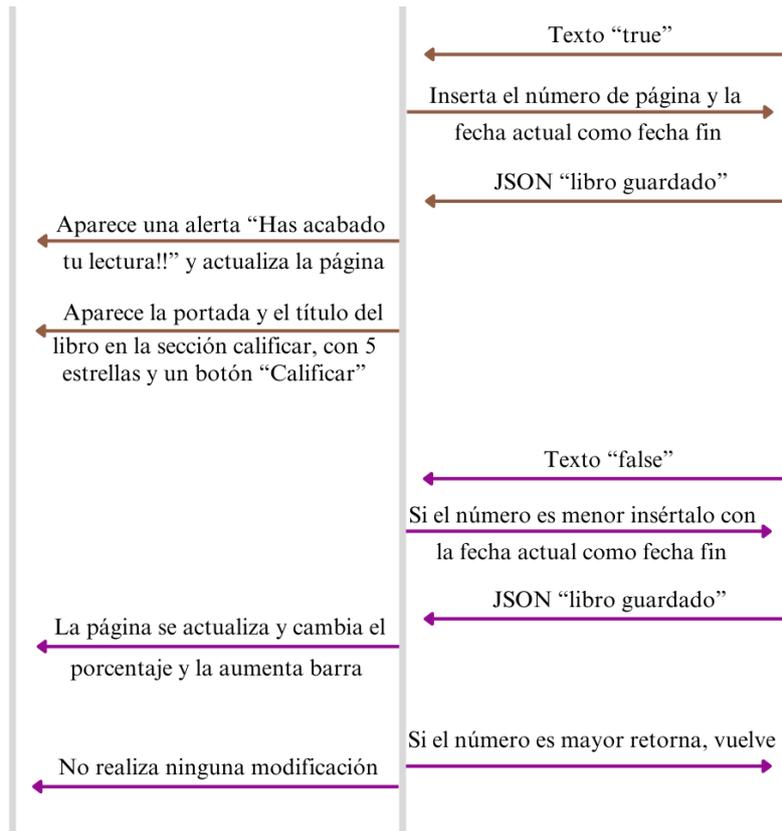


Figura 54. Diagrama de flujo: seguimiento del libro

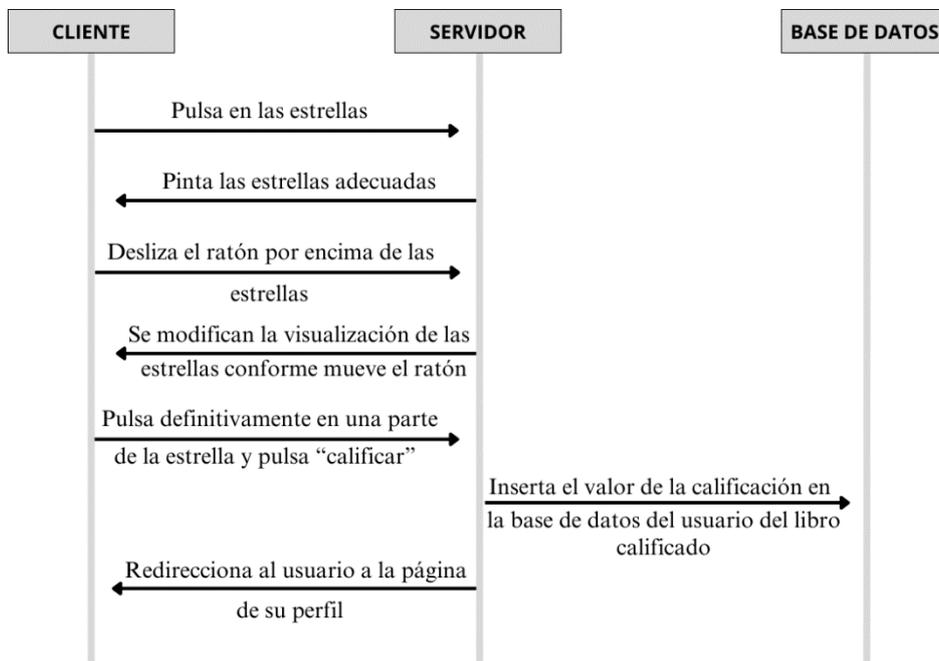


Figura 55. Diagrama de flujo: calificar libro

4.2.5 Feed

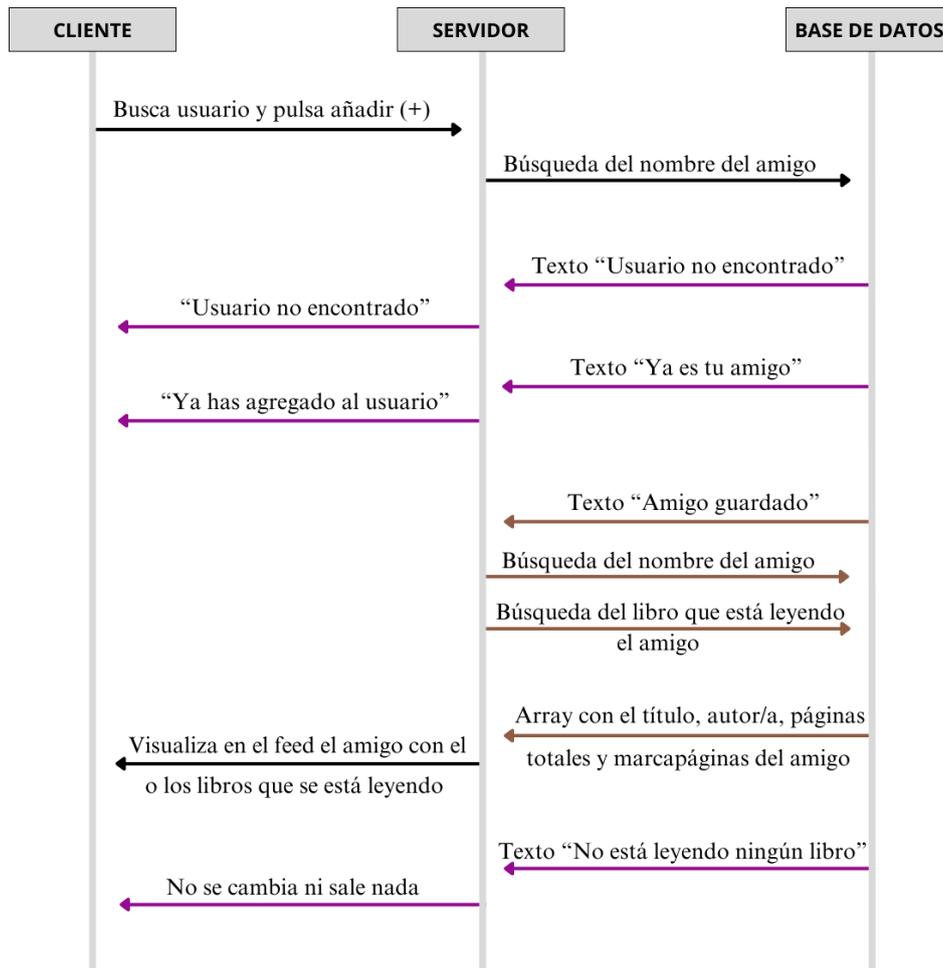


Figura 56. Diagrama de flujo: agregar amigo al feed

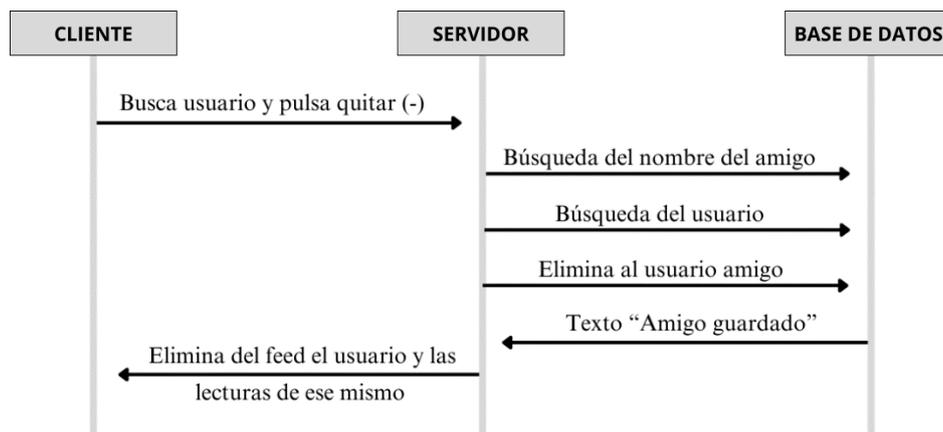


Figura 57. Diagrama de flujo: quitar un amigo del feed

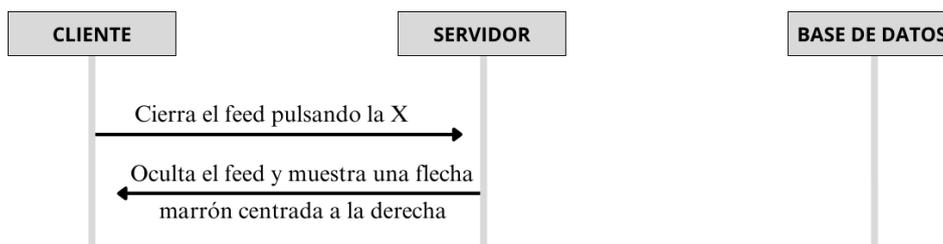


Figura 58. Diagrama de flujo: cerrar y ocultar el feed

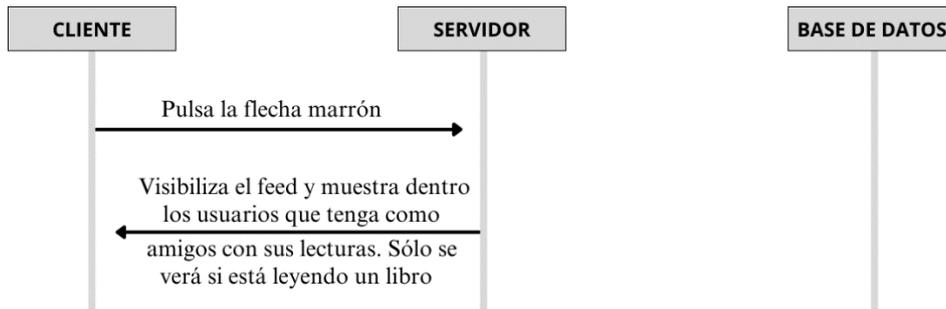


Figura 59. Diagrama de flujo: hacer visible el feed

4.2.6 Biblioteca

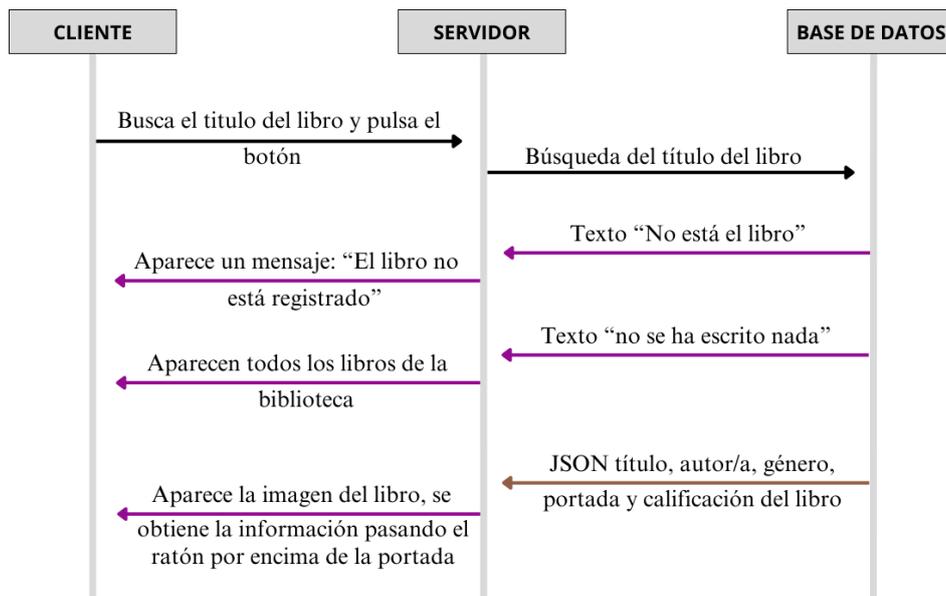


Figura 60. Diagrama de flujo: búsqueda de un libro en la biblioteca

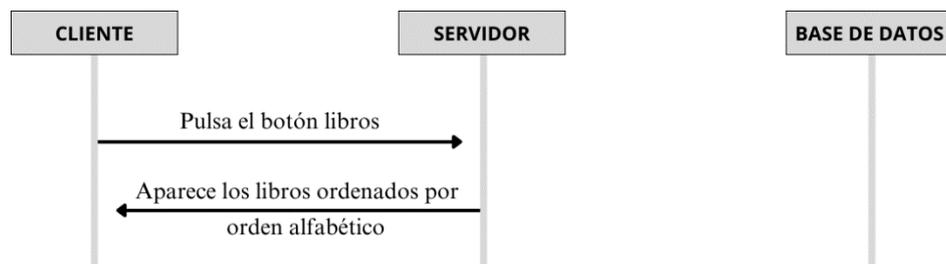
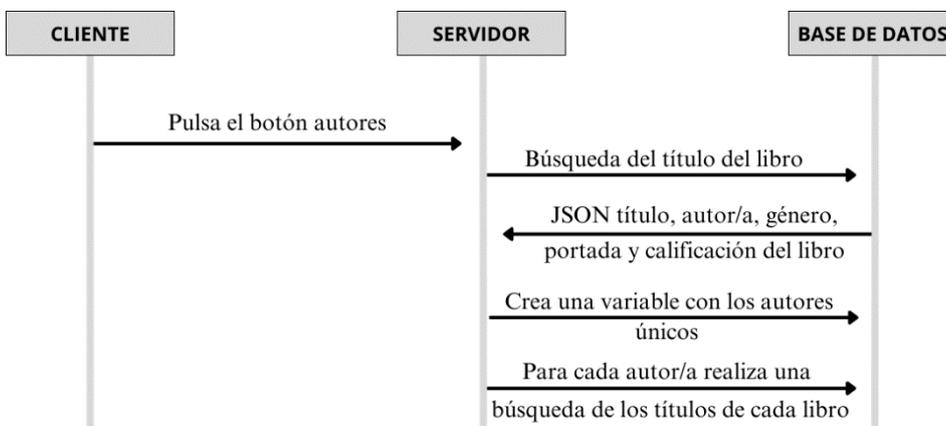


Figura 61. Diagrama de flujo: ordenar alfabéticamente por títulos



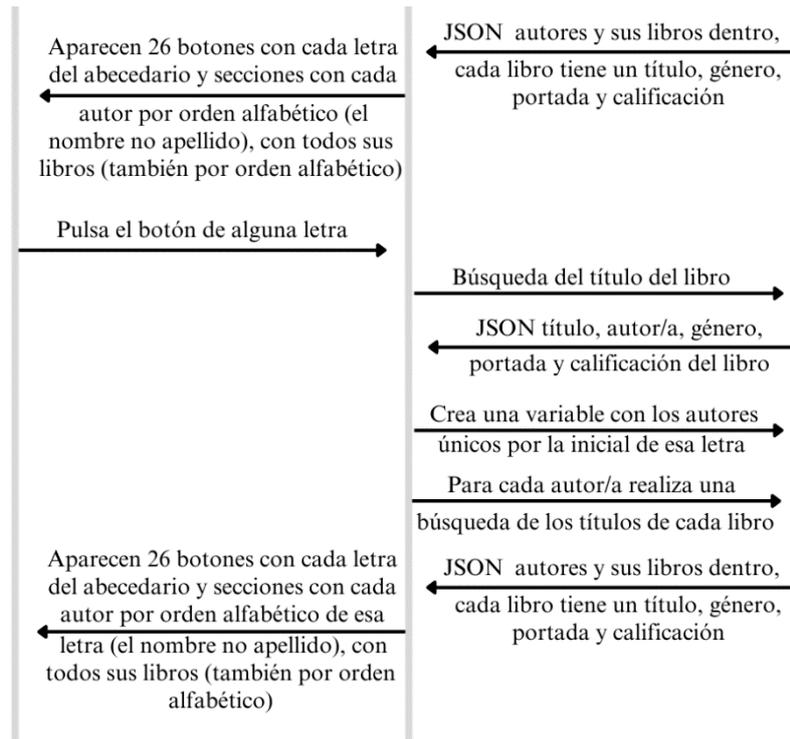


Figura 62. Diagrama de flujo: ordenar alfabéticamente por nombre del autor/a

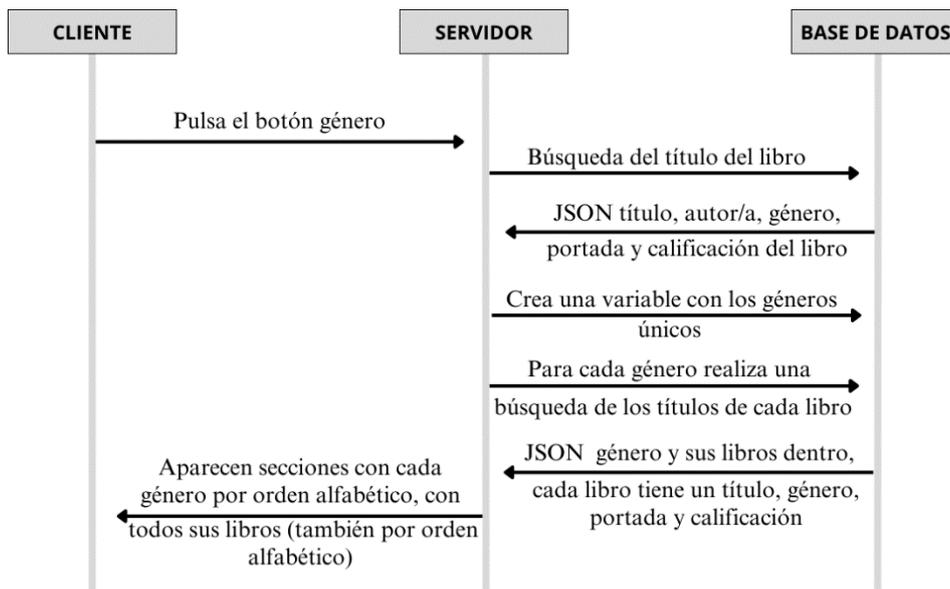


Figura 63. Diagrama de flujo: ordenar alfabéticamente por géneros

4.2.7 Barra de navegación

En este caso, todos los botones que existen realizan la misma función de Figura 51. Las únicas excepciones, son los botones de perfil e iniciar lectura. Estas utilizan una variable local en la que detecta si hay o no un usuario en funcionamiento, es decir, que ha iniciado sesión, explicado en el apartado 4.1.3.1. Dicha variable es almacenada cuando se inicia sesión y se utilizará en la página de perfil y de iniciar lectura.



4.3 Base de datos

Para la base de datos, se ha utilizado una NoSQL gestionada por MongoDB para almacenar y gestionar la información de los usuarios. Se ha utilizado MongoDB Atlas para gestionarlo en la nube, para ello se ha creado una base de datos con una contraseña en la que solo las personas que tengan dicha información podrán acceder a ella.

Se ha creado un fichero donde se definirá la URI, con el nombre de la clase y su contraseña correspondiente, y se creará un cliente con la utilización de *MongoClient* y *ServerApiVersion* desde la librería *mongodb*. Además, se crearán dos funciones, una para establecer la conexión con la base de datos, en mi caso llamada “TFG” y con referencia a la colección “usuarios”. Y la segunda para garantizar que la colección esté disponible antes de devolverla, ya que la exportaremos para usarlo en el fichero *index*.

Capítulo 5. Resultado final

En este capítulo, se observará el resultado final mediante una serie de imágenes de la red social desarrollada. Además, en las imágenes se podrá observar la URL que irá cambiando conforme cambien las páginas.

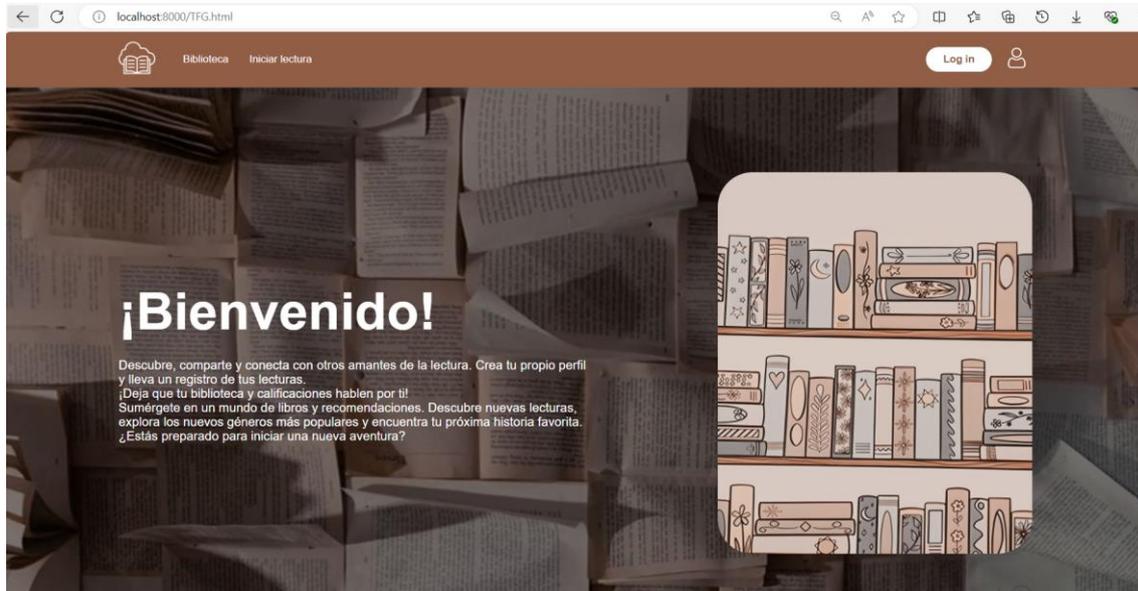


Figura 64. Resultado final: pantalla de inicio

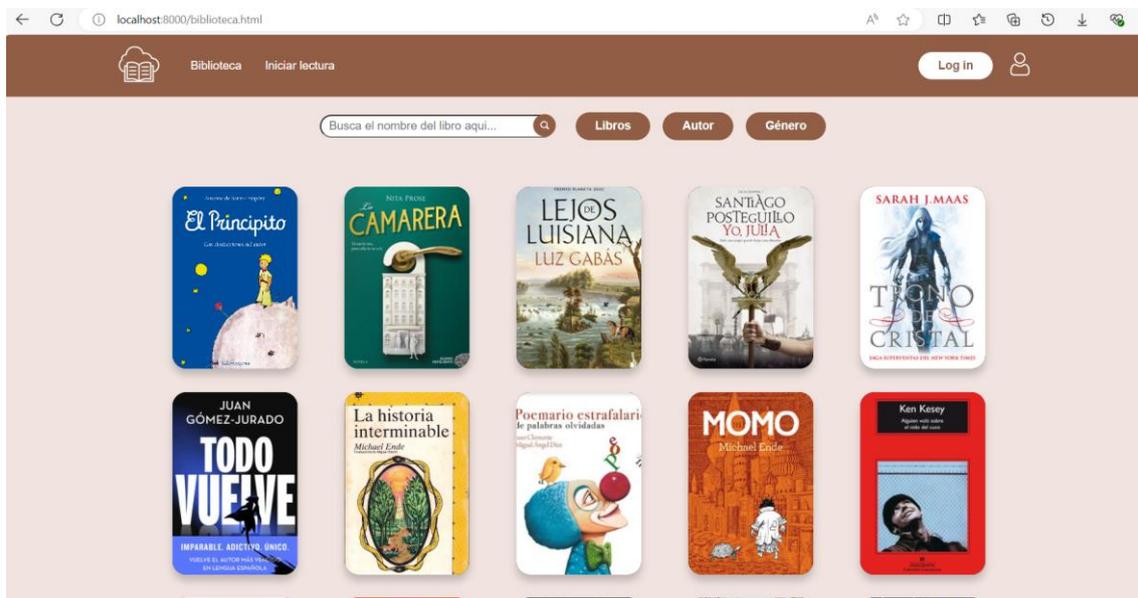


Figura 65. Resultado final: biblioteca

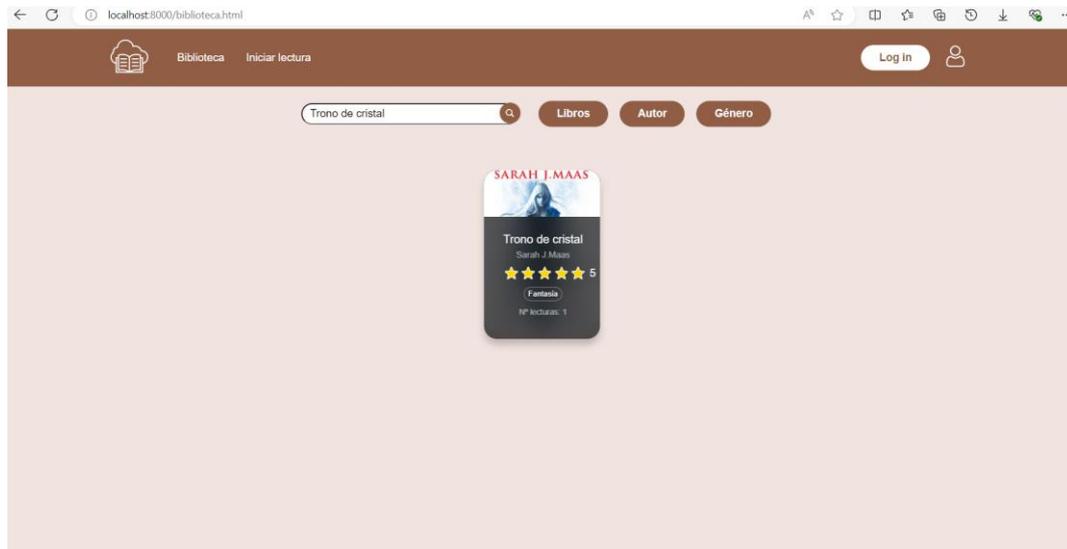


Figura 66. Resultado final: búsqueda de un libro en la biblioteca

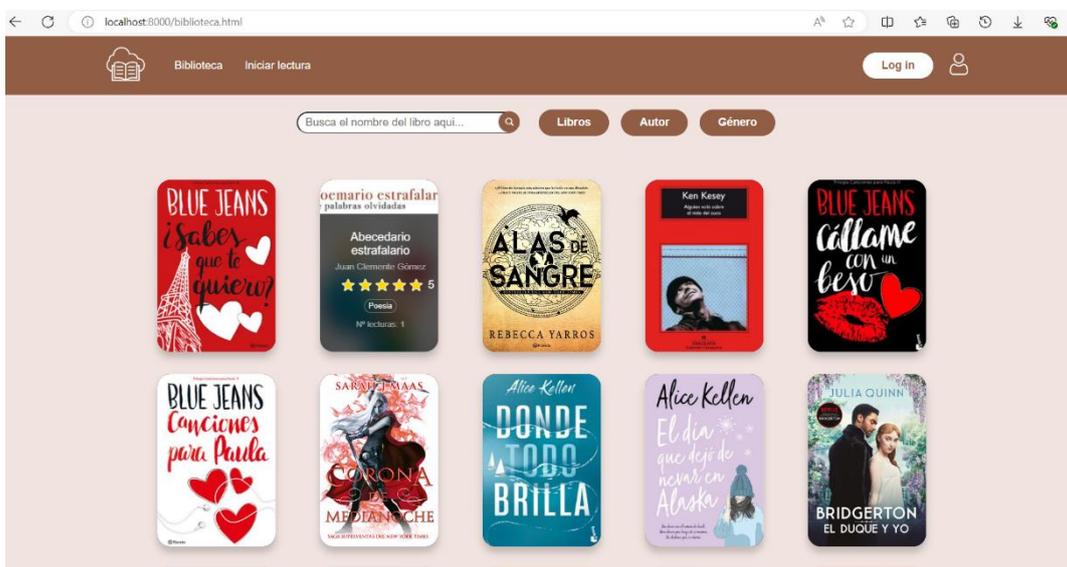


Figura 67. Resultado final: libros por orden alfabético

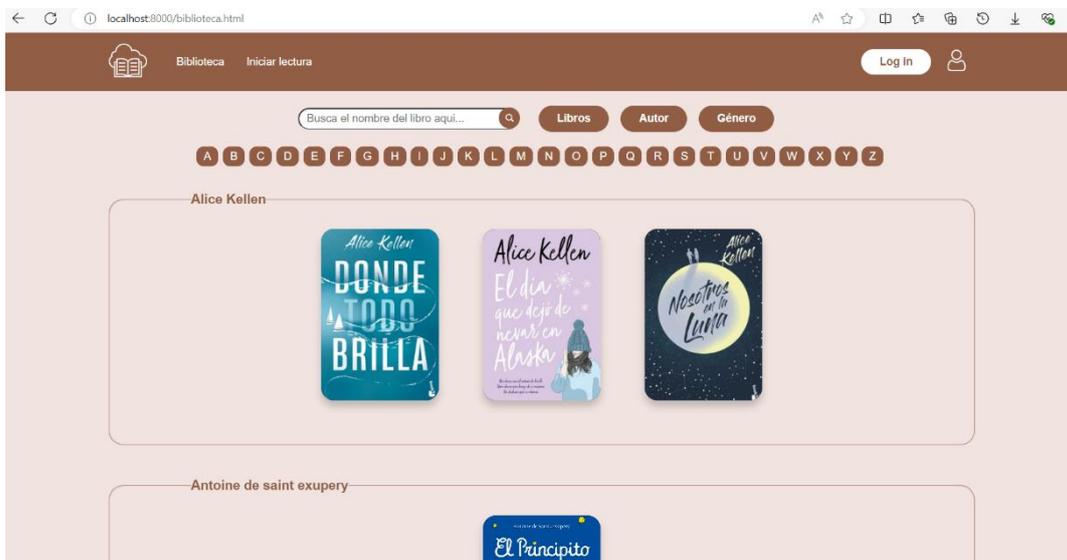


Figura 68. Resultado final: libros por autores orden alfabético

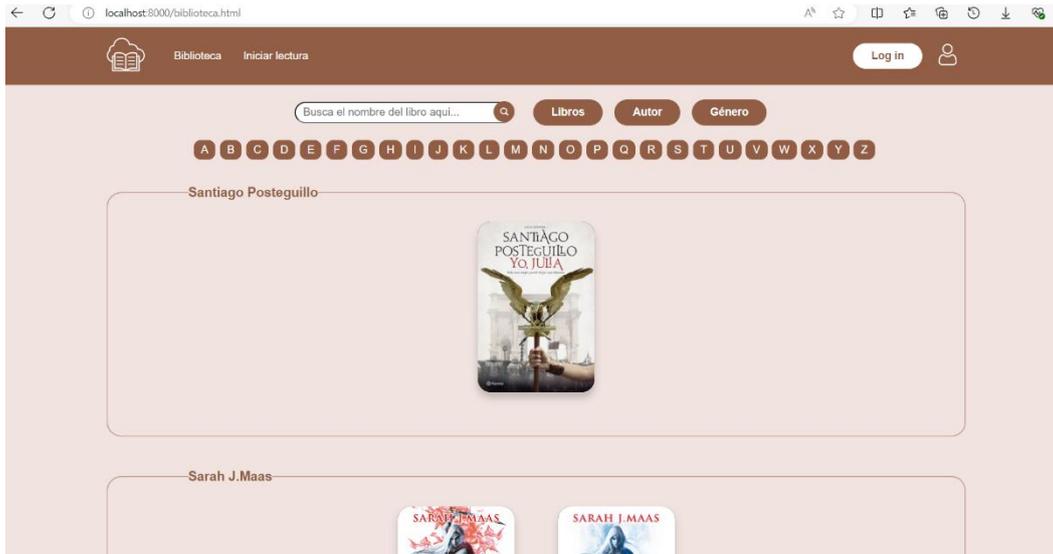


Figura 69. Resultado final: libros por autores por la letra S

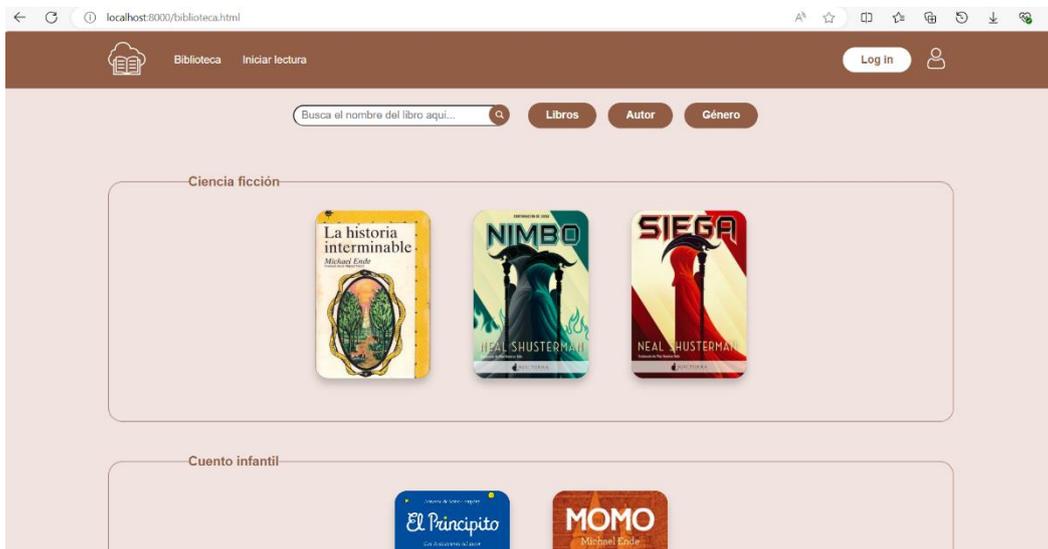


Figura 70. Resultado final: libros por géneros en orden alfabético

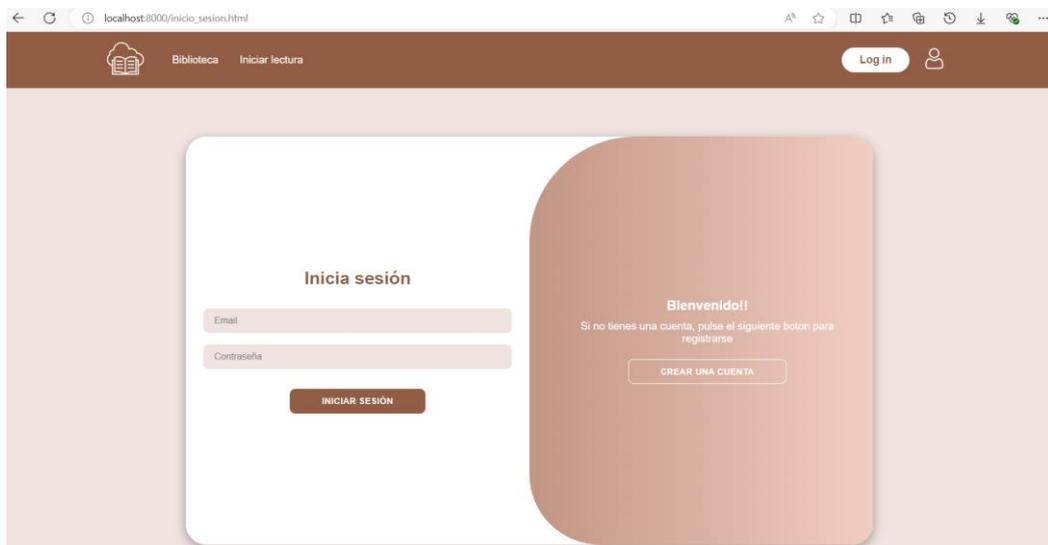


Figura 71. Resultado final: iniciar sesión

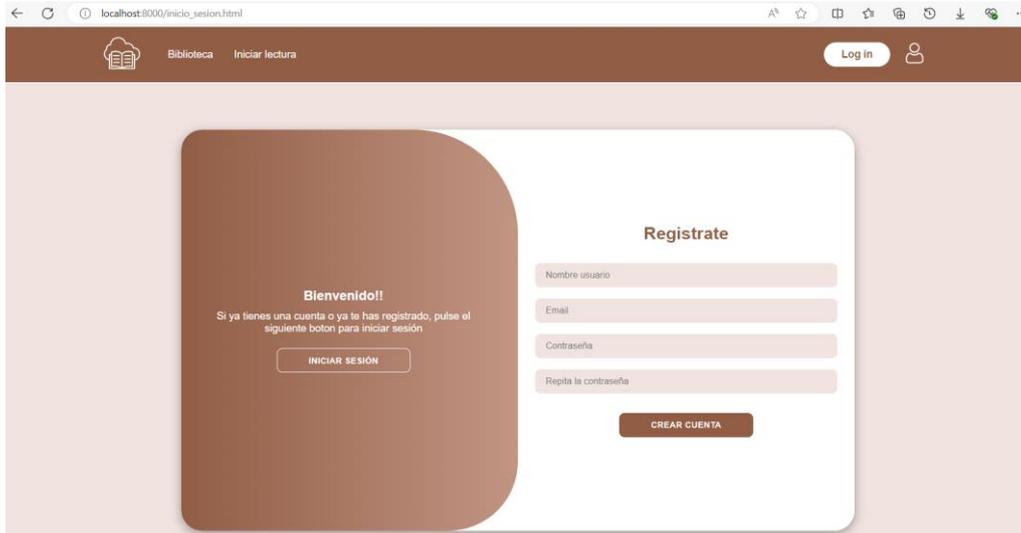


Figura 72. Resultado final: registrar usuario

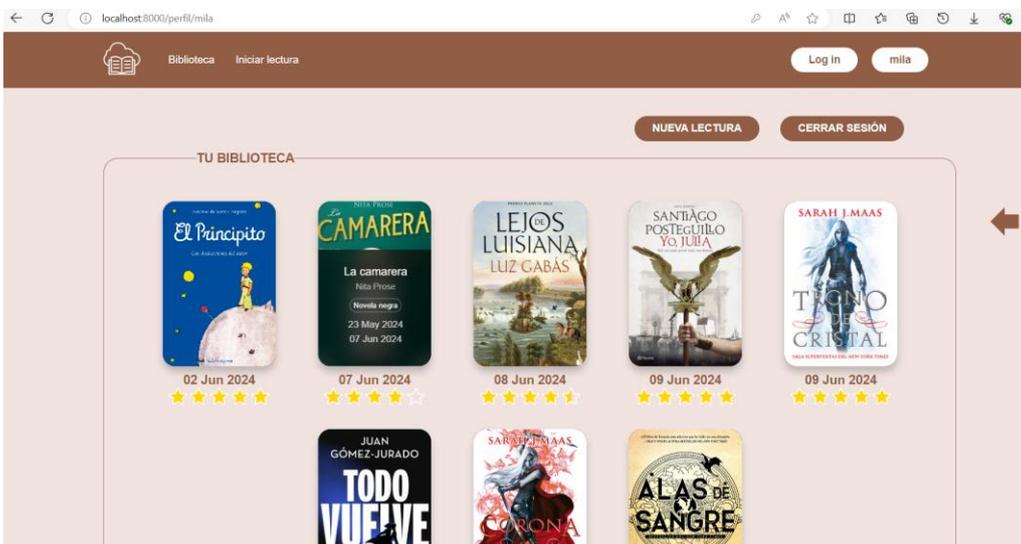


Figura 73. Resultado final: perfil usuario "mila"

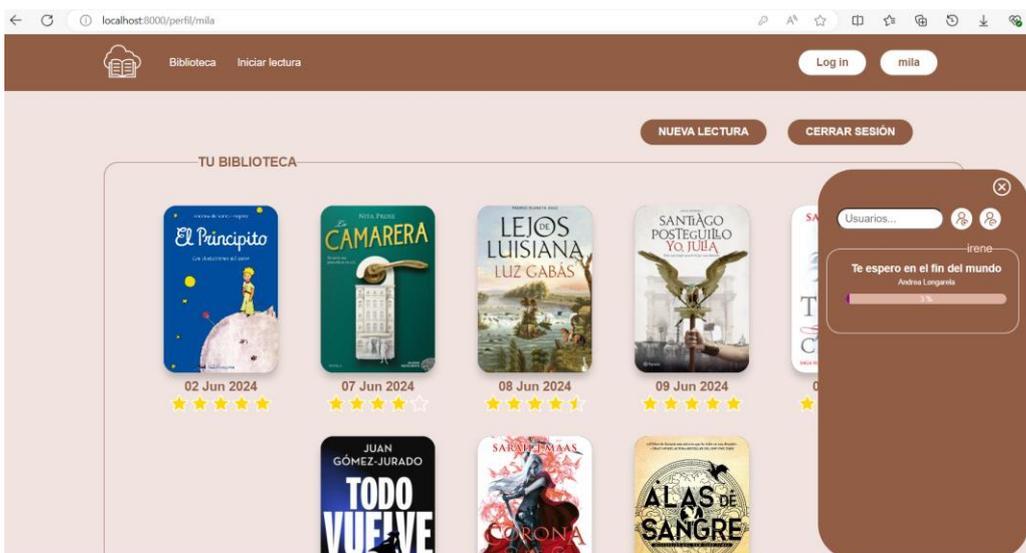


Figura 74. Resultado final: perfil usuario "mila" con el feed visible y con un amigo

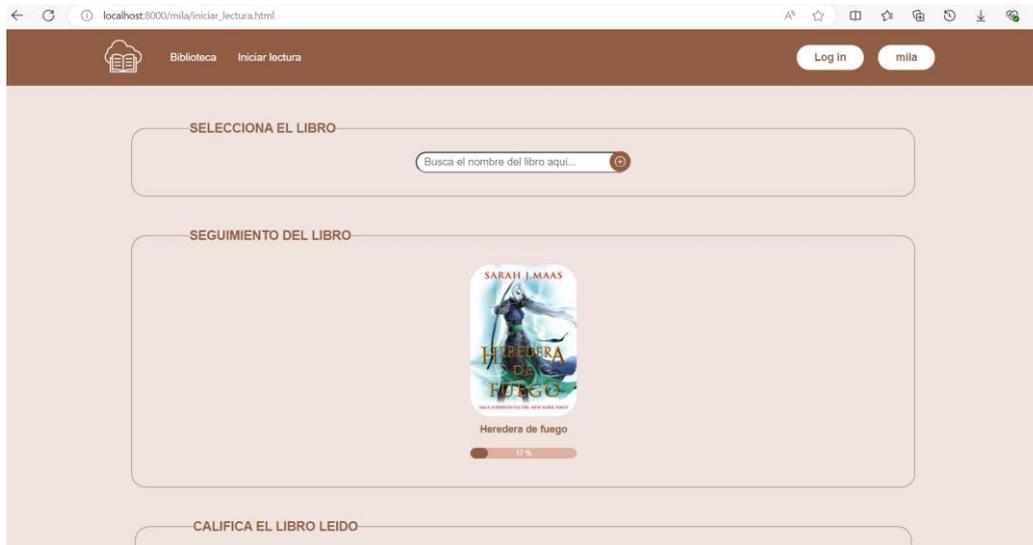


Figura 75. Resultado final: iniciar lectura usuario “mila” con un libro activo

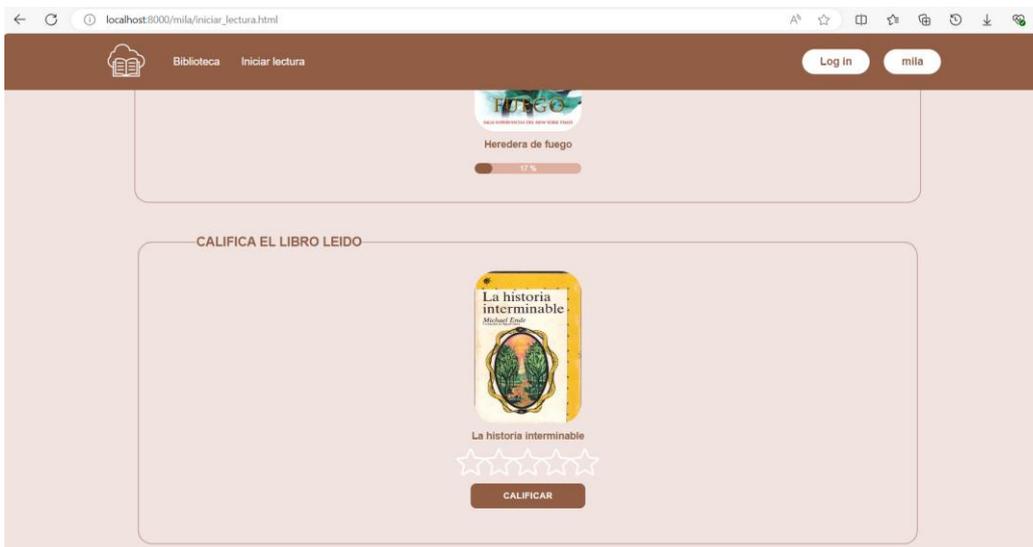


Figura 76. Resultado final: iniciar lectura usuario “mila” con un libro por calificar

Capítulo 6. Conclusión

El enfoque principal de este proyecto ha sido la creación de una comunidad literaria. A lo largo de este documento, se han especificado las diferentes funcionalidades y requisitos, así como el desarrollo de la red social de libros, donde se puede registrar, calificar, y llevar un seguimiento de lecturas, además de tener una biblioteca personal para cada usuario. Asimismo, la plataforma desarrollada permite a los usuarios explorar el mundo literario de una manera moderna y accesible, donde poder conocer nuevos libros.

El objetivo de este proyecto era combinar el amor por la lectura con las tecnologías actuales, ofreciendo una plataforma que atraiga también a las nuevas generaciones y enriquezca la experiencia de los lectores habituales. Este documento ha abordado una variedad de objetivos tanto técnicos como personales, todos los cuales han sido alcanzados con éxito.

El diseño intuitivo y la integración de tecnologías actuales hacen que la plataforma sea atractiva para las nuevas generaciones, que suelen preferir medios digitales. Por lo tanto, esta interfaz puede cumplir en gran medida con el objetivo de atraer a un público joven.

Desde mi punto de vista, este proyecto ha sido una gran superación personal y profesional. La implementación de diferentes lenguajes de programación refleja el aprendizaje y los conocimientos adquiridos previos.

Cabe destacar, que este proyecto se ha subido al GitHub como código abierto [6]. Esta decisión la he tomado porque en el mundo de la programación, el aprendizaje es continuo. Al mantener el código abierto, facilita un proceso de mejora y crecimiento constante. Además, otros desarrolladores pueden contribuir añadiendo nuevas funcionalidades y optimizando y mejorando la calidad del código.

Capítulo 7. Bibliografía

- [1] Zhou, L. [@lisalioz]. (13 de enero, 2024). #lecturaterminada #nuevalectura #libros #diariodelecturas #losrenglonestorcidosdedios [Video]. TikTok. https://www.tiktok.com/@lisalioz/video/7323616251786431777?is_from_webapp=1&sender_device=pc&web_id=7318063096093902369
- [2] Orion Context Broker (OCB) - Aprende FIWARE en Español. (s. f.). <https://fiware-training.readthedocs.io/es-mx/latest/ecosistemaFIWARE/ocb/>
- [3] Node.js Foundation. (2019). Node.js. Node.js. <https://nodejs.org/>
- [4] Saini, R. (s.f.). Bento. <https://bento.me/helpmecoder>
- [5] Saini, R. [@helpme_coder]. (25 de octubre, 2023). Creating modern sign in and sign up forms on a web page is now easier than ever with #HTML, #CSS [Video]. Instagram. https://www.instagram.com/reel/Cy0uv5zPlgw/?utm_source=ig_web_copy_link&igsh=MzRlODBiNWFiZA%3D%3D
- [6] Lagarda, I. [@irenelagarda]. (19 de junio, 2024). irenelagarda-Desarrollo-de-una-red-social-de-libros-full-stack-MERN [Repositorio]. GitHub. <https://github.com/irenelagarda/irenelagarda-Desarrollo-de-una-red-social-de-libros-full-stack-MERN>
- [7] W3Schools. (1998). W3Schools Online Web Tutorials. W3schools.com; W3Schools. <https://www.w3schools.com/>
- [8] Palma, I. (2022, 15 julio). Cómo crear una aplicación React con un Backend de Node: La guía completa. freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/como-crear-una-aplicacion-react-con-un-backend-de-node-la-guia-completa/>
- [9] La API classList de Javascript - Javascript en español. (s. f.). Lenguaje JS. <https://lenguajejs.com/javascript/dom/manipular-clases-css/>
- [10] Migrating to Express 4. (s. f.). <https://expressjs.com/en/guide/migrating-4.html#example-migration>
- [11] Almo. (2024, 5 marzo). Arquitectura de 3 Capas: Simplificando el Diseño de Software - Desarrollo de software. Desarrollo de software - Desarrollo-de-software.com. <https://desarrollo-de-software.com/arquitectura-de-3-capas-simplificando-el-diseno-de->
- [12] Jain, A. (2022, 15 diciembre). Qué es la gestión de requisitos: definición, beneficios, mejores herramientas. Visure Solutions. <https://visuresolutions.com/es/gu%C3%ADa-de-trazabilidad-de-gesti%C3%B3n-de-requisitos/requerimientos-funcionales/#:~:text=Un%20requisito%20funcional%20es%20una%20declaraci%C3%B3n%20de%20c%C3%B3mo>
- [13] Desarrolladoresweb.org. (2023, 8 noviembre). ▷Qué es HTML para qué sirve y cómo funciona - Desarrolladores web. Desarrolladores Web. <https://desarrolladoresweb.org/html/que-es-html/>
- [14] Coppola, M. (s.f.). ¿Qué es React y para qué sirve? Blog.hubspot.es. <https://blog.hubspot.es/website/que-es-react>
- [15] Martínez, P. [@fjmz-UPV]. (2 de febrero, 2024). Material-de-clase [Repositorio]. GitHub. <https://github.com/FRAMEWORKS-23-24/Material-de-clase>
- [16] Moser, L. (s.f.). Custom alert box codeopen.io. <https://codepen.io/XemsDoom/pen/JjOBoG>



[17] Peña, L. (s.f.). *skill adobe* codeopen.io. <https://codepen.io/leninalbertop/pen/zYjvJX>

[18] Sam, P. (s.f.). *Drop-Down list search using Datalist* codeopen.io.
<https://codepen.io/rpsthecoder/pen/yJvRPE>