



UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Máster Universitario en Ingeniería Web

Trabajo Fin de Máster

Aplicación web de gestión de biblioteca

Autor

Jiaming Shi, Shenhui Cheng

Tutor

Santiago Alonso Villaverde

julio de 2024

AGRADECIMIENTOS

SHENHUI CHENG:

Quiero comenzar agradeciendo a mi compañero Jiaming, quien me invitó a participar en este proyecto. Sin su compañerismo, no habiéramos alcanzado nuestros objetivos. En segundo lugar, mi gratitud hacia este máster y hacia nuestros amables profesores por su enseñanza y profundos conocimientos, especialmente a nuestro tutor Santiago, quien siempre nos recordaba los plazos y nos animaba constantemente con su profesional tutoría. En tercer lugar, agradezco a todos los compañeros con quienes compartí este máster, por el enriquecedor intercambio de conocimientos. Por último, quiero agradecer a mi familia y a mí mismo por haber llegado hasta aquí.

JIAMING SHI:

En primer lugar, quiero expresar mi sincero agradecimiento a mi colaborador, Shenhui. Juntos hemos trabajado arduamente en esta investigación y redacción. Su apoyo y colaboración han sido esenciales para el éxito de nuestra tesis. Su sabiduría y perspicacia han enriquecido significativamente nuestro trabajo y me han impulsado a buscar la excelencia. En segundo lugar, agradezco a nuestros profesores, especialmente a nuestro tutor, por su orientación y ayuda durante nuestra investigación. Su rigor académico y conocimiento han proporcionado valiosas sugerencias y dirección. Finalmente, agradezco profundamente a mis padres por su apoyo incondicional y aliento. Gracias por su amor y comprensión, que me han permitido dedicarme plenamente a la investigación y alcanzar este logro. Una vez más, gracias a todos los que nos han ayudado y apoyado en la realización de esta tesis.



RESUMEN

La tecnología evoluciona rápidamente cada día, y en la actualidad, cada vez sentimos la necesidad de obtener información desde páginas web. En la vida cotidiana, todos somos ávidos consumidores de información, pero la forma en que la recibimos puede variar según nuestra edad. En nuestra infancia, aprender a leer un libro es el primer paso para adentrarnos en el mundo del conocimiento y la imaginación.

Desde entonces, ha surgido la idea de implementar una aplicación web de biblioteca, donde cualquier persona pueda acceder a la información y adquirir conocimientos. El propósito de esta implementación no es que las bibliotecas web actuales estén obsoletas, sino que la tecnología ha avanzado mucho, y cada mejora que realizamos permite a las personas acceder a la información de manera más fácil y rápida.

El objetivo de esta aplicación web de biblioteca es proporcionar una solución eficiente y moderna que simplifique la gestión de los recursos bibliotecarios y mejore la experiencia del usuario. Las funcionalidades incluyen la gestión de libros, usuarios, préstamos, devoluciones, autores, carteras electrónicas, información de la biblioteca e historiales de transacciones de las carteras electrónicas.

A través del front-end desarrollado en Angular, los usuarios pueden navegar fácilmente, tomar prestados y devolver libros, mientras que el back-end implementado con Spring Boot se encarga de la lógica de negocio, el acceso a la base de datos y la autenticación de usuarios.

Para la gestión del proyecto, se ha utilizado la metodología ágil (Scrum), que ofrece la ventaja de promover un enfoque iterativo y flexible en el desarrollo, adaptándose así a las necesidades cambiantes del proyecto.

PALABRAS CLAVE

Biblioteca, Angular, Spring Boot, Aplicación web, Metodología Scrum

ABSTRACT

Technology evolves rapidly every day, and nowadays, we increasingly feel the need to obtain information from web pages. In everyday life, we are all avid consumers of information, but the way we receive it may vary depending on our age. In our childhood, learning to read a book is the first step to delve into the world of knowledge and imagination.

Since then, the idea of implementing a web library application has emerged, where anyone can access information and acquire knowledge. The purpose of this implementation is not that current web libraries are obsolete, but that technology has advanced greatly, and every improvement we make allows people to access information more easily and quickly.

The goal of this web library application is to provide an efficient and modern solution that simplifies the management of library resources and enhances the user experience. Functionalities include book management, user management, loans, returns, authors, e-wallets, library information, and transaction histories of e-wallets.

Through the Angular-developed front-end, users can easily navigate, borrow, and return books, while the Spring Boot-implemented back-end handles business logic, database access, and user authentication.

For project management, the Agile methodology (Scrum) has been used, offering the advantage of promoting an iterative and flexible approach in development, thus adapting to the changing needs of the project.

KEYWORDS

Library, Angular, Spring Boot, Web Application, Scrum Methodology.



TABLA DE CONTENIDO

Agradecimientos.....	2
Resumen	3
Abstract.....	4
Tabla de Figuras.....	7
Objetivos.....	10
Objetivos específicos	10
Inception deck	11
1. ¿Por qué estamos aquí?.....	11
2. Elevator pitch	12
3. Caja del producto	12
4. Lista de Noes	13
5. Conoce a tus vecinos.....	15
6. Mostrar la solución	16
7. ¿Qué nos quita el sueño?.....	17
8. Tómale las medidas.....	18
9. Ser claros en lo que vamos a dar	18
10. Muestra lo que te va a costar	19
Product Backlog.....	21
SPRINT 1.....	23
1. Sprint Planning.....	23
2. Sprint Review	23
3. Retrospective	29
SPRINT 2.....	31
1. Sprint Planning.....	31
2. Sprint Review	31
3. Retrospective	47
SPRINT 3.....	48
1. Sprint Planning.....	48
2. Sprint Review	49
3. Retrospective	69
SPRINT 4.....	70
1. Sprint Planning.....	70
2. Sprint Review	72
3. Retrospective	101

Presentación del sistema resultante	102
1. Página de inicio.....	102
2. Vista de cliente registrado.....	106
3. Vista de administrador	116
4. Vista de administrador superior.....	124
5. Vista de usuario prohibido	128
Conclusiones.....	129
Posibles Ampliaciones	130
Bibliografía.....	131



TABLA DE FIGURAS

Figura 1. Imagen diseñada para la caja de producto.....	13
Figura 2. Stack tecnológico.....	16
Figura 3. Flujo de despliegue continuo.....	17
Figura 4. Estructura frontend y backend.....	23
Figura 5. Estructura de backend del proyecto.....	26
Figura 6. Estructura de frontend del proyecto.....	27
Figura 7. Burn down chart de Sprint 1.....	29
Figura 8. Diagrama de caso de diferentes roles de usuario.....	32
Figura 9. Diagrama de clase biblioteca-usuario.....	33
Figura 10. APIs Sprint 2.....	34
Figura 11. Diagrama de secuencia del proceso de inicio de sesión.....	36
Figura 12. Diagrama de secuencia de autenticación.....	37
Figura 13. Diagrama de secuencia de todas las solicitudes.....	38
Figura 14. Diagrama de componentes de frontend de sprint 2.....	39
Figura 15. Diagrama de actividad de inicio de sesión de usuario.....	40
Figura 16. Diagrama de actividad de registro de usuario.....	42
Figura 17. Pruebas de métodos de Sprint 2.....	45
Figura 18. Prueba de entorno de desarrollo.....	45
Figura 19. Resultado de las pruebas de Sprint 2.....	46
Figura 20. Cobertura de las pruebas.....	46
Figura 21. Burn down chart de Sprint 2.....	47
Figura 22. Diagrama de caso de uso monedero-registro de transacciones de sprint 3.....	49
Figura 23. Diagrama de caso de uso información personal-avatar de sprint 3.....	50
Figura 24. Diagrama de clase de Sprint 3.....	51
Figura 25. APIs de Sprint 3.....	52
Figura 26. Diagrama de actividad de recarga de Sprint 3.....	56
Figura 27. Diagrama de actividad de retiro de Sprint 3.....	58
Figura 28. Diagrama de actividad de cambio de contraseña de Sprint 3.....	60
Figura 29. Diagrama de actividad de restablecer contraseña de Sprint 3.....	62
Figura 30. Diagrama de componentes de Sprint 3.....	63
Figura 31. Diagrama de actividad de recarga saldo en frontend.....	64
Figura 32. Diagrama de actividad de retirar saldo en frontend.....	65
Figura 33. Pruebas (1) de métodos de Sprint 3.....	67
Figura 34. Pruebas (2) de métodos de Sprint 3.....	67
Figura 35. Cobertura de pruebas (1) de Sprint 3.....	68
Figura 36. Cobertura de pruebas (2) de Sprint 3.....	68
Figura 37. Pruebas superadas de Sprint 3.....	68
Figura 38. Burn down chart de Sprint 3.....	69
Figura 39. Caso de uso de gestión de libros de backend.....	72
Figura 40. Diagrama de clase de Sprint 4.....	73
Figura 41. APIs (1) de backend de Sprint 4.....	74
Figura 42. APIs (2) de backend de Sprint 4.....	74
Figura 43. APIs (3) de backend de Sprint 4.....	75
Figura 44. APIs (4) de backend de Sprint 4.....	75

Figura 45. APIs (5) de backend de Sprint 4.....	75
Figura 46. APIs (6) de backend de Sprint 4.....	75
Figura 47. Diagrama de actividad del préstamo de un libro de backend de Sprint 4	86
Figura 48. Diagrama de actividad de la devolución de libros de backend de Sprint 4	88
Figura 49. Diagrama de actividad de confirmación del estado de devolución de libros de backend de Sprint 4.....	90
Figura 50. Diagrama de actividad de denegación del estado de devolución de libros de backend de Sprint 4.....	92
Figura 51. Diagrama de componentes de Sprint 4 frontend	93
Figura 52. Diagrama de actividad de préstamo de libros de frontend de Sprint 4	94
Figura 53. Diagrama de actividad de devolución de libros de frontend de Sprint 4	95
Figura 54. Diagrama de actividad de confirmación de la devolución de libros de frontend de Sprint 4	96
Figura 55. Pruebas (1) de Sprint 4	99
Figura 56. Pruebas (2) de Sprint 4	99
Figura 57. Resultado de pruebas (1) de Sprint 4	99
Figura 58. Resultado de pruebas (2) de Sprint 4	100
Figura 59. Resultado de pruebas (3) de Sprint 4	100
Figura 60. Resultado de pruebas (4) de Sprint 4	100
Figura 61. Burn down chart de Sprint 4.....	101
Figura 62. Pantalla de inicio (1)	102
Figura 63. Pantalla de inicio (2)	103
Figura 64. Pantalla de login	104
Figura 65. Pantalla de error de login	104
Figura 66. Pantalla de sign up (1)	105
Figura 67. Pantalla de sign up (2)	105
Figura 68. Pantalla de sign up (3)	106
Figura 69. Pantalla de inicio (1) de cliente registrado	107
Figura 70. Pantalla de inicio (2) de cliente registrado	107
Figura 71. Pantalla de encabezado de cliente registrado.....	108
Figura 72. Pantalla de libro de cliente registrado.....	108
Figura 73. Pantalla de lista de préstamo de libro	109
Figura 74. Pantalla de información de préstamo de libro	109
Figura 75. Pantalla de lista de devolución de libro.....	110
Figura 76. Pantalla de información de devolución de libro.....	110
Figura 77. Pantalla de perfil de usuario.....	111
Figura 78. Pantalla de página personal de usuario.....	111
Figura 79. Pantalla de recarga de saldo.....	112
Figura 80. Pantalla de retiro de saldo.....	113
Figura 81. Pantalla de registro de transacciones.....	114
Figura 82. Pantalla de perfil de usuario (1).....	114
Figura 83. Pantalla de perfil de usuario (2).....	115
Figura 84. Pantalla de perfil de usuario (3).....	115
Figura 85. Pantalla de página de inicio (1) de administrador	116
Figura 86. Pantalla de página de inicio (2) de administrador	116
Figura 87. Pantalla de página personal de administrador	117
Figura 88. Pantalla de gestión de usuarios.....	117
Figura 89. Pantalla de página personal de otros usuarios.....	118
Figura 90. Pantalla de gestión de libros.....	118



Figura 91. Pantalla de modificar libros	119
Figura 92. Pantalla de gestión de autores	120
Figura 93. Pantalla de perfil de autor	120
Figura 94. Pantalla de tipo de libros	121
Figura 95. Pantalla de caja de devolución de libros	121
Figura 96. Pantalla de comentarios de libros	122
Figura 97. Pantalla de lista de préstamo de libros de administrador (1)	122
Figura 98. Pantalla de lista de préstamo de libros de administrador (2)	123
Figura 99. Pantalla de lista de devolución de libros de administrador	124
Figura 100. Página de inicio de administrador superior (1)	124
Figura 101. Página de inicio de administrador superior (2)	125
Figura 102. Pantalla de gestión de usuarios de administrador superior	125
Figura 103. Pantalla de página personal de otros usuarios de administrador superior...	126
Figura 104. Pantalla de gestión de información de biblioteca de administrador superior	127
Figura 105. Pantalla de página de inicio de usuario prohibido	128

OBJETIVOS

El objetivo principal de este Trabajo Fin de Máster (TFM) es utilizar los conocimientos adquiridos durante el máster en Ingeniería Web para desarrollar una aplicación web destinada a la gestión de una biblioteca. Esto dará una solución eficiente y moderna que simplifique la gestión de los recursos bibliotecarios y mejore la experiencia del usuario. Permitirá a los clientes realizar funciones como préstamo y devolución de libros, mientras que los administradores de la biblioteca pueden gestionar los recursos de libros y órdenes.

Objetivos específicos

- a) Análisis y diseño de funcionalidades de la aplicación: Analizar en detalle los requerimientos de funcionalidad para “CLIENT”, “ADMINISTRATOR”, “ROOT” y “BAN”, diseñar módulos de funcionalidad correspondientes, incluyendo autenticación de usuarios, gestión de préstamos y devolución de libros, gestión de registro de libros (añadir información relacionado como autor, título, ISBN, tipo del libro etc.), gestión de autores (añadir información de presentación del autor), modificación de perfiles personales, visualización de registros de órdenes, gestión de billeteras electrónicas y la información general de la biblioteca.
- b) Gestión de permisos de la aplicación: Diseñar un sistema de permisos de usuario seguro y confiable para asegurar que los distintos roles de usuarios tengan los permisos correspondientes, por ejemplo, los clientes pueden prestar y devolver libros, mientras los administradores tienen un perfil superior pueden gestionar los préstamos y devolución de libros.
- c) Optimización de la experiencia del usuario y la interactividad: Enfocarse en mejorar la experiencia del usuario, incluyendo, pero no limitándose a, la amigabilidad de la interfaz, velocidad de respuesta, y optimización de la actualización de páginas. Asegurar una experiencia operativa fluida y conveniente para los usuarios al utilizar la aplicación, además de proporcionar confirmación por correo electrónico sobre transacciones exitosas de órdenes.
- d) Gestión y análisis de datos: Diseñar una estrategia adecuada de gestión de datos, incluyendo almacenamiento, consultas y análisis de datos como libros, autores y órdenes. Brindar soporte de datos para asistir en la toma de decisiones gerenciales más informadas.



INCEPTION DECK

En el desarrollo de proyectos de software, particularmente en metodologías ágiles y enfoques de planificación iterativa, la fase inicial es crucial para establecer una fundación sólida. El Inception Deck, concepto acuñado y popularizado por Jonathan Rasmusson en su libro "The Agile Samurai", es una herramienta estratégica diseñada para alinear al equipo del proyecto con las necesidades del usuario y los objetivos empresariales [1]. En el contexto de nuestro proyecto, este documento estratégico es fundamental para lanzar el desarrollo de una aplicación web de gestión de bibliotecas. Nuestro enfoque se centrará en optimizar la gestión de recursos bibliográficos de manera adaptativa y eficiente, promoviendo una operatividad más ágil dentro de las bibliotecas.

El objetivo principal del Inception Deck en este contexto es asegurar que todas las partes interesadas tengan una comprensión clara y concisa del alcance del proyecto, los riesgos, las suposiciones, y la visión general del producto a desarrollar. A través de esta herramienta, el equipo podrá identificar y acordar las prioridades fundamentales del proyecto, los criterios de éxito, y establecer una hoja de ruta clara que guíe todas las fases de desarrollo de la aplicación web de gestión de biblioteca.

A lo largo de esta sección, se delinearán la estructura del Inception Deck, describiendo cada uno de sus componentes clave y cómo estos contribuyen al entendimiento y éxito del proyecto. Se abordarán temas como la definición del problema, la propuesta de valor, los usuarios y clientes, el modelo de negocio, y las estimaciones preliminares de tiempo y recursos, entre otros. Al final de este proceso, el equipo de desarrollo deberá tener una visión compartida del proyecto y estar preparado para avanzar en la planificación y ejecución con una base sólida y compartida. Asegurando que todas las fases del proyecto se alineen con las necesidades reales y expectativas de los usuarios finales.

1. ¿Por qué estamos aquí?

Nos encontramos aquí porque reconocemos la necesidad urgente de modernizar las herramientas de gestión en las bibliotecas, que actualmente se enfrentan a desafíos significativos debido a sistemas obsoletos que complican tanto la experiencia del usuario como la administración interna. Nuestra nueva aplicación web pretende revolucionar la manera en que las bibliotecas gestionan sus operaciones diarias, mejorando la experiencia de los usuarios mediante una interfaz intuitiva y funcionalidades avanzadas que simplifican los procesos de préstamo y devolución de libros.

El objetivo fundamental de este proyecto es ofrecer una solución integral para la gestión de biblioteca, proporcionando una plataforma eficaz para manejar todos los recursos de la biblioteca, desde libros hasta revistas. Nuestra aplicación web permitirá a los administradores monitorear de manera sencilla el estado de préstamo y las fechas de vencimiento de los recursos disponibles. Y para los usuarios como clientes, simplificará significativamente los procesos de préstamo y devolución de libros, abordando asuntos como libros vencidos y multas, lo que minimizará errores y la necesidad de intervención manual.

Además, para los usuarios como clientes, nuestra aplicación web ofrecerá una serie de operaciones como proceso de registro fácil, autenticación segura, manejo de carteras electrónicas, edición de datos personales y administración de multas. Además, se implementará un sistema de recordatorios automáticos para la devolución de libros y actualización constante de información relevante. Esta combinación de funciones garantizará una gestión bibliotecaria más eficiente y enfocada en la satisfacción del cliente.

2. Elevator pitch

¿Alguna vez has experimentado frustraciones al usar sistemas de gestión de bibliotecas obsoletos, tanto como usuario buscando libros o como administrador tratando de mantener el orden y la eficiencia? ¡Nuestra aplicación web de gestión de biblioteca es la elección ideal para ti! Está diseñada para simplificar y modernizar integralmente todas las operaciones de tu biblioteca. Ofrecemos una gestión completa que abarca desde el préstamo hasta la devolución de libros, optimizando las tareas diarias de los bibliotecarios y mejorando simultáneamente la experiencia del usuario. Con nuestra plataforma, los usuarios pueden acceder fácilmente a la información de los libros, realizar seguimientos en tiempo real de sus pedidos y consultar el estado de su cartera electrónica. Implementamos tecnología de punta para garantizar la máxima seguridad de los datos y proteger la privacidad de todos los usuarios. Únete a nosotros y eleva el nivel de servicio de tu biblioteca, ofreciendo una experiencia más fluida y segura tanto para administradores como para usuarios clientes.

3. Caja del producto

3 beneficios importantes:

- La aplicación web de biblioteca implementada es eficiente y moderna que simplifica la gestión de los recursos bibliotecarios.
- La aplicación web de biblioteca implementada mejora y optimiza la experiencia del usuario.
- La aplicación web de biblioteca implementada es segura y confiable que protege los datos de los distintos usuarios y que asegura con distintos roles ejercer operaciones diferentes.

Eslogan:

- ¡Accede, Explora, Protege: ¡Tu Biblioteca Digital Segura y Moderna!



Imagen:



Figura 1. Imagen diseñada para la caja de producto

4. Lista de Noes

La "Lista de Noes" es diseñada para establecer claramente las limitaciones y exclusiones del proyecto desde el principio. Esta lista ayuda a alinear expectativas y evitar malentendidos entre las partes interesadas, proporcionando un marco claro de lo que está dentro del alcance, fuera del alcance y las cuestiones que aún no se han resuelto. A continuación, se presenta la lista de noes para este proyecto, organizada en tres categorías: dentro del alcance, fuera del alcance y no resueltas.

Dentro del alcance:

- **Gestión de roles de usuario:**

A) usuario "ROOT": Será un tipo de usuario administrador superior, en nuestra aplicación web este usuario será único e intransferible. Se crea automáticamente tras la inicialización del servidor y podrá gestionar todos los datos de la plataforma.

B) usuario "ADMINISTRATOR": Será un tipo de usuario administrador también, será concedida por usuario "ROOT" u otros usuarios "ADMINISTRATOR". Este usuario podrá gestionar la información de libros.

C) usuario "CLIENT": Será un tipo de usuario registrado. El cliente podrá usar la aplicación web implementada rellenando el formulario con número de teléfono, email y una contraseña personalizada para obtener una cuenta y hacer el login.

D) usuario "BAN": Será un tipo de usuario cliente también, sin embargo, este usuario no podrá procesar ninguna solicitud por lo que ha sido bloqueado por el usuario administrador de que no se ha devuelto el libro en el plazo indicado.

- **Registro de cuenta personal:** Cualquier usuario podrá registrar una cuenta personal con su número de teléfono, pero el número de teléfono será único para cada usuario siendo que es la clave para hacer el login.
- **Inicio de sesión con la cuenta personal:** Cualquier usuario siendo no de tipo "BAN" podrá iniciar sesión con su cuenta para acceder al sistema web y hacer operaciones correspondientes.
- **Edición de información personal:** Los usuarios de tipo "CLIENT" y "ADMINISTRATOR" podrán modificar sus informaciones personales utilizando sus cuentas, tales como nombre, correo electrónico, biografía, fecha de nacimiento y contraseña. Cabe destacar que el usuario de tipo "ROOT" tendrá derecho de editar la información personal de todos los usuarios siendo un administrador superior único e intransferible.
- **Gestión de cartera electrónica:** Los usuarios de tipo "CLIENT" y "ADMINISTRATOR" podrán recargar su cartera electrónica, ver saldo y realizar otras operaciones. Sin embargo, el usuario de tipo "ROOT" podrá gestionar la cartera electrónica de todos los usuarios.
- **Consulta de pedidos personales:** Cualquier usuario siendo no de tipo "BAN" podrá ver todos los pedidos de préstamo y devolución.
- **Consulta de transacciones de cartera personal:** Los usuarios de tipo "CLIENT" y "ADMINISTRATOR" podrán ver el historial de transacciones de su cartera. Sin embargo, el usuario de tipo "ROOT" podrá ver el historial de transacciones de todos los usuarios.
- **Gestión de libros:** Los usuarios de tipo "ADMINISTRATOR" y "ROOT" podrán gestionar libros, incluyendo añadir, modificar y eliminar libros. Pero si un libro se encuentra en un estado prestado, ninguno de estos usuarios podrá eliminar el libro.
- **Búsqueda de libros:** Cualquier usuario (incluyendo no registrados) podrá consultar la información de libros en la biblioteca.
- **Préstamo de libros:** Sólo los usuarios de tipo "CLIENT" podrán hacer la operación de préstamo de libros. El plazo máximo de devolución es de 3 meses. Una vez el cliente confirme la operación, el sistema web reducirá el importe correspondiente de la cartera del cliente.
- **Devolución de libros:** Sólo los usuarios de tipo "CLIENT" podrán hacer la operación de devolución de libros. Una vez el cliente confirme la operación, el libro estará en el buzón de devolución, esperando que el administrador confirme la situación real de la devolución del libro y deposite el importe correspondiente a la cartera del cliente según las reglas (deducción proporcional por daño o retraso).
- **Añadir libros a la lista de favoritos:** Sólo los usuarios de tipo "CLIENT" podrán añadir libros a su lista de favoritos para seguir su situación. Además, el usuario de tipo "ROOT" tiene derecho de gestionar estas listas de favoritos.
- **Gestión de pedidos:** Sólo los usuarios de tipo "CLIENT" podrán consultar sus propios pedidos de préstamo y devolución. Sin embargo, los usuarios de tipo "ADMINISTRATOR" y "ROOT" podrán ver los pedidos de cualquier persona.



- **Gestión de devolución:** Los usuarios de tipo “ADMINISTRATOR” y “ROOT” podrán gestionar los libros devueltos, confirmar la devolución y depositar el importe correspondiente según las reglas de devolución a la cartera del usuario.

Fuera del alcance:

- **Seguridad en las transacciones:** La aplicación web no contempla la seguridad en la recarga de la cartera electrónica con tarjeta bancaria y retirar fondos.
- **Compartir reseñas y experiencias de lectura:** La aplicación web no contempla la funcionalidad de compartir reseñas y experiencias de lectura por usuario.

No resueltas:

- No se ha resuelto la integración con sistemas de gestión de bibliotecas existentes: Todavía no se ha decidido cómo se integrará la nueva aplicación web con los sistemas de gestión de bibliotecas ya en uso.
- No se ha resuelto cómo manejar las devoluciones de libros fuera de horario: Aún no se ha determinado un proceso claro para la devolución de libros fuera del horario de atención de la biblioteca, una vez que el cliente devuelve el libro necesita que el administrador confirme la situación real de la devolución de libro.
- Lectura en línea de libros: La aplicación web no contempla la lectura en línea de los libros disponibles.

5. Conoce a tus vecinos

En cualquier proyecto, comprender quiénes son las partes interesadas y cómo interactúan es crucial para el éxito. Este apartado del inception deck nos permite mapear a todos los actores claves.

Equipo de Desarrollo:

El equipo de desarrollo estará compuesto por:

- **Ingeniero Senior:** Definirá metas, diseñará historias de usuario y dirigirá la estrategia del equipo.
- **Ingeniero Junior:** Implementará el desarrollo de las historias de usuario siguiendo la estrategia establecida por el Ingeniero Senior.

Equipo de Diseño:

- **Diseñador UI/UX:** Responsable del diseño de la interfaz y de la experiencia del usuario.

Equipo de Operaciones:

- Encargado del despliegue, mantenimiento y supervisión del proyecto para asegurar su funcionamiento continuo.

Equipo de Marketing:

- Encargado de promover la aplicación entre usuarios potenciales y recopilar comentarios sobre su uso.

Equipo de Atención al Cliente:

- Responsable de resolver problemas de los usuarios en el uso de la aplicación.

Usuarios finales:

- **Personal de la Biblioteca:** administrador de la biblioteca responsable de supervisar y gestionar recursos, préstamos y devoluciones a través de la aplicación.
- **Cientes:** Usuarios regulares de la biblioteca que utilizan la aplicación para realizar consultas, préstamos y devoluciones, así como para acceder a servicios adicionales.
- **Usuarios de la app:** Todos los usuarios que usarán la aplicación serán encargados de probar la aplicación y son los principales usuarios de este servicio. Su feedback es esencial para el equipo.

6. Mostrar la solución

Este proyecto es un sitio web que se desplegará en internet, accesible para los usuarios a través de cualquier dispositivo conectado a internet y que tenga un navegador. Los usuarios interactúan con el frontal web para realizar solicitudes de datos al backend.

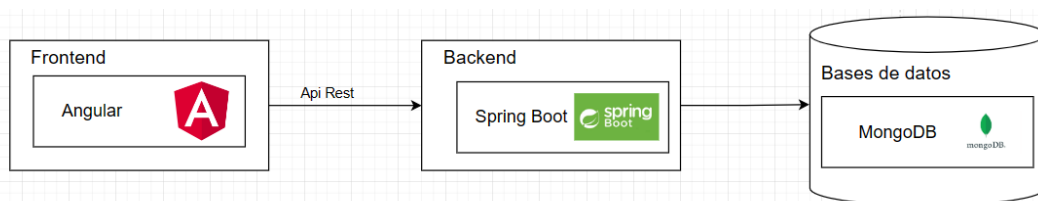


Figura 2. Stack tecnológico

- **Frontend:** Responsable de presentar la interfaz gráfica de la aplicación al usuario, nuestro equipo ha decidido utilizar Angular, gracias a su conjunto de herramientas y bibliotecas que permiten a nuestro equipo de desarrollo construir el proyecto de manera fácil y rápida.
- **Backend:** Encargado de gestionar datos y ejecutar la lógica, hemos decidido utilizar Spring Boot para construir el backend en Java y habilitar la comunicación con el frontend a través de una API Rest.
- **Base de Datos:** Encargada de almacenar datos, sólo es accesible por el backend. Debido a la gran cantidad de datos de libros en la biblioteca y para garantizar la velocidad y eficiencia de las consultas, hemos decidido utilizar MongoDB.

Control de versiones del Proyecto:

- Con el objetivo de gestionar eficazmente las versiones del proyecto y fomentar la colaboración en equipo, hemos optado por utilizar GitHub como una

plataforma de control de versiones. GitHub proporciona herramientas avanzadas de control de versiones, especialmente basada en sus características de Git, lo que nos permite realizar un seguimiento, crear ramas y fusionar el código de manera sencilla.

- Crearemos dos repositorios independientes en GitHub, uno dedicado a la gestión del proyecto Angular en el frontend y otro para el proyecto Spring Boot en el backend. Esta estructura nos ayuda a organizar y mantener el código del proyecto de manera eficiente, permitiendo que los miembros del equipo se centren en sus responsabilidades específicas.
- En el proceso de desarrollo, se seguirá estrictamente una estrategia de gestión de ramas, distinguiendo claramente entre la rama “master” y la rama “develop”, con el fin de separar diferentes versiones y evitar interferencias entre historias de usuario. Para una gestión eficiente del proyecto, para cada historia de usuario o tarea técnica relacionada, se creará una rama específica bajo la rama “develop”. El propósito de esto es asegurar que, después de que todas las pruebas hayan sido aprobadas y confirmada la rama con los cambios, esta se pueda fusionar con la rama “develop”. Sólo cuando todas las historias de usuario dentro de una fase de sprint han sido completadas, se fusionará la rama “develop” con la rama “master”. Adoptar esta estrategia de gestión de ramas permite asegurar efectivamente la estabilidad de la rama “master”, apoyando así el desarrollo continuo y robusto del proyecto.
- Se creará un flujo de integración continua para cada repositorio de nuestro proyecto. Cuando la rama 'master' completa la fusión con la rama 'develop', el código se construirá, probará, desplegará y publicará automáticamente en sus respectivos entornos de trabajo.

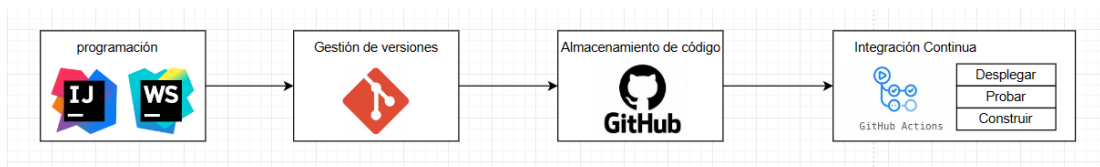


Figura 3. Flujo de despliegue continuo

7. ¿Qué nos quita el sueño?

- **Riesgo de Tiempo de Desarrollo:** El desarrollo de este proyecto tiene un límite de tiempo de 3 meses. Durante este periodo, cualquier error técnico grave podría resultar en la imposibilidad de completar el proyecto a tiempo.
- **Riesgo de Control de Calidad:** La presión del tiempo puede conducir a descuidar las pruebas y el control de calidad, lo que afectaría la estabilidad y calidad final del producto.

8. Tómale las medidas

Para una mejor planificación del tiempo del proyecto y asegurar la entrega puntual del mismo, haremos estimaciones basadas en la guía Scrum [2]. Utilizando prácticamente la técnica de Planning Poker [3] para facilitar estas estimaciones. Para cada elemento se describe la tarea, el valor de negocio (una escala de 1 a 100, donde 100 indica el máximo valor para el negocio) y el esfuerzo.

Elemento	Tipo	Valor	Esfuerzo
Gestión del proyecto	Técnico	80	6
Gestión de CI/CD	Técnico	70	4
Gestión de BD	Técnico	85	1
QA	Técnico	75	8
Gestión de usuario	Historia de usuario	65	20
Gestión de la información de usuario	Historia de usuario	60	29
Gestión de libros	Historia de usuario	90	25
Gestión de devolución de libros	Historia de usuario	70	13
Gestión de préstamos de libros	Historia de usuario	70	18
Gestión de autores	Historia de usuario	40	13
Gestión de E-wallet	Historia de usuario	70	20
Gestión de información de biblioteca	Historia de usuario	55	4
Gestión de historiales de transacciones de E-wallet	Historia de usuario	65	6

9. Ser claros en lo que vamos a dar

Es importante entender que la perfecta ejecución de un proyecto requiere tanto de circunstancias favorables como de recursos adecuados. En condiciones ideales, el proyecto debería avanzar según el cronograma establecido y entregarse puntualmente. Sin embargo, es habitual que surjan imprevistos inevitables durante el desarrollo del proyecto. Por esta razón, hemos decidido anticiparnos y, al desarrollar soluciones correspondientes, planificar posibles contratiempos. Hemos elaborado una tabla de prioridades para enumerar las áreas del proyecto en las que podríamos hacer concesiones. Hemos asignado puntajes de prioridad del 0 al 10, donde 10 es la máxima prioridad y 0 es la mínima.

Factores Clave	Prioridad	Razón	Solución
Tiempo	10	No podemos retrasar la entrega del TFM.	Seguir estrictamente el cronograma, en caso de problemas técnicos, asignar tiempo adicional para resolverlos.
Interfaz	5	Una interfaz atractiva mejora la experiencia del usuario, pero las funcionalidades son más importantes.	En situación con tiempo suficiente, mejoraremos la UI/UX, pero en situación de tiempo limitado, nos centraremos en ofrecer sistema de servicios básicas.



Calidad	10	La calidad es fundamental para la reputación del equipo de desarrollo.	Si surgen problemas durante el desarrollo, temporalmente eliminaremos historias de menor valor para garantizar la calidad.
Alcance	1	Las funciones principales del proyecto no cambiarán, pero algunas funciones de menor valor podrían ser reducidas.	En reuniones previas a la iteración, podríamos decidir eliminar historias de poco valor para asegurar tiempo suficiente para las funciones principales.
Seguridad	8	La seguridad de los datos de los usuarios es de mayor importancia.	En el proceso de desarrollo, dedicaremos más tiempo a probar la seguridad de los datos para garantizar la máxima protección.
Presupuesto	8	El presupuesto es fundamental para el desarrollo del proyecto.	Reduciremos gastos innecesarios y reservaremos los fondos necesarios para asegurar el desarrollo fluido del proyecto.

10. Muestra lo que te va a costar

En esta sección vamos a estimar los costes necesarios para la elaboración de este proyecto, se clasifican en diferentes tipos que se mostrarán en continuación.

- **Coste de personal:**

Equipo	Posición	Salario bruto anual empleado	Coste bruto mensual empresa	Cantidad	Meses para el proyecto	Coste
Departamento de Desarrollo	Senior	36.200€	3.010€	1	1	3.010€
	Junior	20.450€	1.700€	1	1	1.700€
Departamento de Diseño	UX/UI Diseñador	23.000€	1.900€	1	0.5	950€
Departamento de Sistema	Técnico de sistema	20.000€	1.670€	1	0.5	835€
					Total	6495€

La columna “Salario bruto anual empleado” se obtiene mediante un promedio estadístico [4] [5].

- **Coste de infraestructura:**

Item	Precio/mes	Cantidad	Total
Ordenador	50€	3	200€
Otro	10€	3	40€
		Total	240€

Reservamos una parte de coste para el concepto de “Otro” en caso de que necesitemos comprar algunos cables para el funcionamiento de ordenadores. Los

precios de alquiler de ordenadores son precios estándar que se obtienen mediante la plataforma Grover [6].

- **Costes totales para tres meses (tiempo del proyecto):**

Item	Precio/mes
Coste de Personal	8.098€
Coste de infraestructura	240€
Total	8.338€



PRODUCT BACKLOG

ID	Título	Descripción	SP	Prioridad
1	Crear proyecto Spring	Como desarrollador de backend, quiero iniciar un proyecto Spring para establecer la base de nuestro servicio backend, asegurando que utilice las dependencias adecuadas y esté listo para futuras extensiones.	3	Muy alta
2	Crear proyecto Angular	Como desarrollador de frontend, quiero iniciar un proyecto Angular para empezar a construir la interfaz de usuario de nuestra aplicación web.	3	Muy alta
3	Crear CI para Spring	Como desarrollador de backend, quiero configurar un flujo de trabajo de integración continua para nuestro proyecto Spring para automatizar pruebas y despliegues, mejorando la calidad del código.	2	alta
4	Crear CI para Angular	Como desarrollador de frontend, quiero configurar un flujo de trabajo de integración continua para nuestro proyecto Angular para asegurar que cada cambio cumple con los estándares de calidad antes de su despliegue.	2	alta
5	Configurar BD	Como desarrollador de backend, quiero configurar una base de datos para almacenar y gestionar los datos de la aplicación de manera segura y eficiente.	1	alta
6	Configurar HTTPS	Como desarrollador de backend, quiero configurar HTTPS en nuestro servidor Spring para asegurar que toda la comunicación entre el cliente y el servidor sea segura.	1	baja
7	Inicio de sesión de usuarios	Como usuario, quiero poder iniciar sesión en el sistema, para acceder a mis datos y funcionalidades personalizadas.	5	alta
8	Registro de usuarios	Como nuevo usuario, quiero poder registrarme en el sistema, para poder crear una cuenta y utilizar los servicios disponibles.	3	alta
9	Gestión de usuarios	Como administrador, quiero gestionar los usuarios para poder editar, bloquear y asignar roles a los usuarios.	8	media
10	Personalización de menús basada en roles	Como usuario, quiero ver un menú de navegación que refleje mis permisos y rol, para acceder solo a las funciones permitidas para mi perfil.	2	media
11	Gestión de la información de biblioteca	Como administrador, quiero gestionar la información de la biblioteca, para mantener los datos actualizados y organizados de manera fácil y accesible.	2	media
12	Visualización de Información de biblioteca	Como usuario, quiero ver la información de la biblioteca.	2	media
13	Modificación de datos personales del usuario	Como usuario, quiero modificar mi información personal, para mantener mis datos actualizados.	5	media
14	Consulta de avatar de usuario	Como usuario, quiero poder consultar mi avatar.	3	media
15	Actualización de contraseña de usuario	Como usuario, quiero cambiar mi contraseña, para mantener mi cuenta segura.	5	media
16	Autenticación de rol de usuarios	Como usuario, quiero que los distintos roles en la plataforma solo puedan acceder a las funciones y datos relacionados con su rol, para asegurar que se respete la confidencialidad y la integridad de la información en función de las responsabilidades de cada usuario.	5	media
17	Consulta de saldo de monedero personal	Como usuario, quiero consultar mi monedero personal, para ver mi saldo disponible.	3	alta
18	Simulación de recarga de monedero electrónico	Como usuario, quiero recargar mi monedero electrónico, para agregar fondos a mi cuenta.	3	media
19	Simulación de retiro de monedero electrónico	Como usuario, quiero retirar fondos de mi monedero electrónico, para usar mi saldo fuera del sistema.	2	media

20	Consulta de transacciones de monedero electrónico	Como usuario, quiero ver el historial de mis transacciones, para revisar los detalles de mis movimientos financieros.	3	media
21	Envío de correo de notificación tras transacción de monedero electrónico	Como usuario, quiero recibir un correo electrónico de notificación tras una transacción exitosa, para estar informado sobre mis movimientos financieros.	3	baja
22	Configuración personalizada de notificaciones de correo para transacciones de monedero electrónico	Como usuario, quiero configurar mis preferencias de notificación por correo electrónico sobre transacciones exitosas de monedero electrónico, para recibir sólo la información que considero relevante.	3	baja
23	Desarrollo de página personalizada de usuario	Como usuario, quiero ver mi página personalizada, para acceder rápidamente a la información y funciones más importantes.	3	media
24	Modificación del avatar de usuario	Como usuario, quiero cambiar mi avatar, para actualizar mi imagen de perfil.	3	media
25	Gestión de libros	Como administrador, quiero crear, modificar, eliminar, buscar y consultar libros, para mantener organizada la colección de la biblioteca.	5	alta
26	Gestión de autores de libros	Como administrador, quiero crear, consultar, eliminar y modificar autores de libros, para mantener organizada la información de los autores.	3	alta
27	Gestión de etiquetas de categorías de libros	Como administrador, quiero crear, consultar, modificar y eliminar etiquetas de categorías de libros, para organizar mejor los libros en la biblioteca.	3	alta
28	Libro del día	Como usuario, quiero ver la funcionalidad de "Libro del Día" para descubrir nuevos libros recomendados.	2	baja
29	Préstamo de libros y reducción de saldo en monedero electrónico	Como usuario, quiero realizar préstamos de libros, para poder leer libros y gestionar mi saldo de monedero electrónico.	5	baja
30	Configuraciones personalizadas de notificaciones de préstamo	Como usuario, quiero configurar mis preferencias para recibir notificaciones por correo sobre órdenes de préstamo, para estar informado sobre mis actividades de préstamo.	3	baja
31	Devolución de libros	Como usuario, quiero devolver libros de biblioteca, para completar mis órdenes de préstamo.	5	media
32	Gestión de cajas de devolución de libros	Como administrador, quiero gestionar las cajas de devolución de libros, para organizar y controlar el proceso de devolución de libros.	5	media
33	Feedback sobre situación de devolución de libros	Como administrador, quiero proporcionar feedback sobre la situación de devolución de libros, para informar sobre cualquier problema o comentario.	3	media
34	Consulta de órdenes de préstamo y devolución	Como usuario, quiero consultar mis órdenes de préstamo y devolución, para llevar un seguimiento de mis actividades de préstamo y devolución.	5	media
35	Consulta de resumen de información de órdenes	Como usuario, quiero ver un resumen de la información de mis órdenes, para tener una visión general de mis actividades de préstamo y devolución.	3	media
36	Agregación y eliminación de libros a la lista de favoritos	Como usuario, quiero agregar o eliminar libros a mi lista de favoritos, para tener una lista personalizada.	2	media
37	Página principal del autor del libro	Como usuario, quiero ver una página principal del autor del libro, para conocer más sobre el autor y ver otras obras suyas registradas en la biblioteca, con un menú contextual para facilitar la navegación.	3	media



SPRINT 1

1. Sprint Planning

Periodo: 10/02/2024 - 18/02/2024

Participantes: JIAMING SHI, SHENHUI CHENG

Tarea principal: La tarea principal de este sprint consiste en preparar la infraestructura necesaria para el desarrollo del próximo proyecto. Esto incluye la creación de la estructura inicial y la instalación y configuración de diversas herramientas de desarrollo esenciales. Esta fase es crucial para asegurar la estabilidad del entorno de desarrollo y facilitar una colaboración eficaz dentro del equipo.

Abreviatura:

- ID: ID de la historia de usuario
- T-ID: ID de la tarea
- Est.: esfuerzo estimado
- TR: tiempo real consumido

Historias de usuario involucradas en este sprint:

ID	Título	T-ID	Tarea	Est.	TR
1	Crear proyecto Spring	1001	Inicializar proyecto Spring	3	6h
2	Crear proyecto Angular	1002	Inicializar proyecto Angular	3	3h
3	Crear CI para Spring	1003	Configurar el flujo de trabajo de integración continua para el proyecto Spring	2	2h
4	Crear CI para Angular	1004	Configurar el flujo de trabajo de integración continua para el proyecto Angular	2	2h
5	Configurar BD	1005	Configuración de la base de datos	1	2h
6	Configurar HTTPS	1006	Configuración de clave HTTPS en el backend	1	2h

2. Sprint Review

En este Sprint se ha definido la estructura del proyecto, así como la configuración del entorno para poder empezar a trabajar en los siguientes Sprints, sirviendo como una base para la producción futura.

Estructura Lógica de Frontend y Backend:

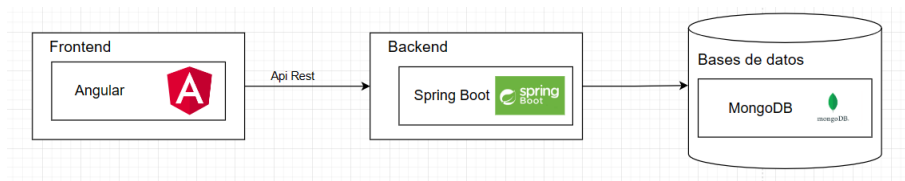


Figura 4. Estructura frontend y backend

Decidimos utilizar una arquitectura de separación front-end/back-end. El núcleo del backend es Spring Boot, utilizado para crear y gestionar interfaces de programación

de aplicaciones (APIs), facilitando así la comunicación bidireccional y el intercambio de datos con el frontend. El frontend se desarrolla utilizando el framework Angular, que permite una interacción dinámica con las APIs proporcionadas por el backend, promoviendo de esta manera un flujo de datos coherente y eficiente entre las dos capas de la aplicación.

En cuanto al almacenamiento y gestión de datos, el backend integra la base de datos NoSQL MongoDB debido a su flexibilidad y escalabilidad en la gestión de grandes volúmenes de datos, así como su capacidad para soportar estructuras de datos complejas [7], lo cual es crucial para la implementación efectiva de un sistema de gestión de bibliotecas. Esta configuración no solo asegura la persistencia y seguridad de los datos, sino que también garantiza una gestión eficiente de los mismos, facilitando operaciones como la consulta, inserción, actualización y eliminación de registros de manera óptima.

Arquitectura de backend:

El backend se desarrolla con el lenguaje de programación Java, utilizando el marco Spring con las dependencias de Maven. Con esto se logra un proceso de desarrollo conveniente y bien organizado [8]. Hemos seleccionado una plantilla de proyecto vacío de Spring Boot como nuestra base inicial de desarrollo y hemos añadido en el pom.xml las dependencias necesarias para el desarrollo siguiente.

Las dependencias utilizadas en pom.xml:

Spring Boot y Seguridad

- **Spring Boot Starter:** Proporciona configuración automática básica y dependencias de Spring Boot.
- **Spring Boot Starter Web:** Soporte para aplicaciones web, incluyendo RESTful, utilizando Spring MVC. Usa Tomcat como el contenedor predeterminado.
- **Spring Boot Starter OAuth2 Client:** Para integrar funcionalidades de OAuth2 en la aplicación, permitiendo autenticación y autorización.
- **Spring Boot Starter Security:** Integración de Spring Security para autenticación y autorización.
- **Spring Security LDAP:** Para autenticar y administrar usuarios con un servidor LDAP.
- **Spring Boot Starter Data MongoDB:** Proporciona integración con MongoDB, incluyendo repositorios automáticos y mapeo de documentos.
- **Spring Boot Starter Actuator:** Ayuda a monitorear y gestionar la aplicación con endpoints de producción-ready.
- **Spring Boot Starter Mail:** Para el envío de correos electrónicos.

Documentación API

- **SpringDoc OpenAPI UI:** Genera documentación de la API en formato OpenAPI, accesible a través de una interfaz de usuario.



Generación de Tokens y PDF

- **JWT (Java Web Token):** Biblioteca para generar y verificar tokens JWT, útil para la autenticación de API.

Utilidades de Desarrollo

- **Lombok:** Biblioteca que ayuda a reducir el código repetitivo en Java, como getters, setters, equals, hashCode, y toString mediante anotaciones.
- **Jacoco:** Plugin para generar informes de cobertura de código.

Otras Dependencias Específicas

- **Flapdoodle Embed MongoDB:** Utilidad para utilizar una instancia de MongoDB embebida en tests, facilitando las pruebas integradas sin necesidad de una instancia real de MongoDB.

Dependencias de Pruebas

- **Spring Boot Starter Test:** Incluye soporte para pruebas con JUnit, Mockito, Hamcrest y otras utilidades de prueba.
- **Spring Security Test:** Proporciona soporte para pruebas de componentes de seguridad.
- **Spring Boot Starter Webflux:** Para aplicaciones reactivas, incluyendo soporte para WebClient y WebTestClient en pruebas.

De acuerdo con la decisión de la primera reunión de sprint, adoptaremos la arquitectura hexagonal para el desarrollo del backend. La arquitectura hexagonal es un patrón de diseño que favorece el mantenimiento y la prueba del sistema, caracterizado por su flexibilidad, sostenibilidad y excelente capacidad de desacoplamiento [9].

A continuación, se presenta la organización estructural del backend del proyecto [10]:

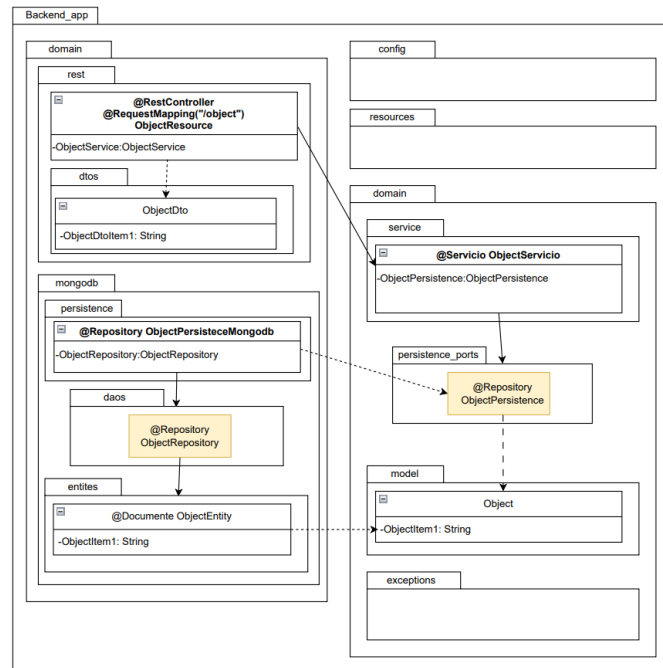


Figura 5. Estructura de backend del proyecto

Flujo de procesamiento de solicitudes:

Cuando un usuario envía una solicitud a través de la API REST, la solicitud primero llega al adaptador REST. El adaptador REST procesa la solicitud, convirtiéndola en una llamada que los servicios de dominio pueden entender, y realiza alguna validación o transformación inicial de datos. El servicio ejecuta la lógica de negocio y llama a las interfaces de la capa de persistencia para interactuar con la base de datos. Estas interfaces de persistencia son implementadas por las clases en `mongodb/persistence/`, que usan los DAOs definidos en `mongodb/daos/` para interactuar con MongoDB. Una vez completada la lógica de negocio, los datos de respuesta son enviados de vuelta al adaptador REST, que luego los devuelve al usuario.

Capa Dominio:

- **model/**: Contiene las entidades de negocio o modelos de dominio del proyecto. Estas son estructuras de datos que reflejan las reglas y características del negocio del proyecto.
- **services/**: Contiene los servicios con la lógica de negocio, manejando decisiones o acciones de negocio complejas. Estos servicios operan sobre los modelos de dominio y pueden llamar a interfaces de la capa de persistencia.
- **Persistence_ports/**: Define interfaces para la persistencia de datos. Estas interfaces son implementadas por la capa de adaptadores, interactuando con la base de datos.
- **exceptions/**: Excepciones personalizadas, usadas para manejar situaciones de error específicas dentro de la lógica del dominio.



Capa Adaptadores:

- **rest/**: El adaptador REST maneja todas las operaciones relacionadas con solicitudes y respuestas HTTP. Cuando un sistema externo interactúa con la aplicación a través de la API REST, convierte la solicitud en un formato que la capa de dominio puede entender y luego llama al servicio de dominio correspondiente.
- **mongodb/**: Implementa las interfaces de la capa de dominio para la persistencia, proporcionando una implementación concreta para MongoDB.
- **daos/**: Los objetos de acceso a datos (DAOs) proporcionan métodos para interactuar con MongoDB, como consultas y operaciones de eliminación.
- **entities/**: Define las clases de entidad que se mapean a las colecciones de MongoDB.
- **persistence/**: Contiene implementaciones de DAOs que realizan las interfaces de persistencia definidas en la capa de dominio.

Arquitectura del Frontend:

El frontend utiliza el marco Angular, construido sobre TypeScript, HTML y CSS, siguiendo el patrón de arquitectura estándar de Angular. La estructura del proyecto es clara, organizando el código y las funcionalidades principalmente a través de las características de componentización de Angular, lo que permite la modularidad y reutilización del código [11]. Cada componente encapsula una lógica de negocio y vista específica, haciendo que el proyecto sea fácil de mantener y expandir.

A continuación, se presenta la organización estructural del frontend del proyecto:

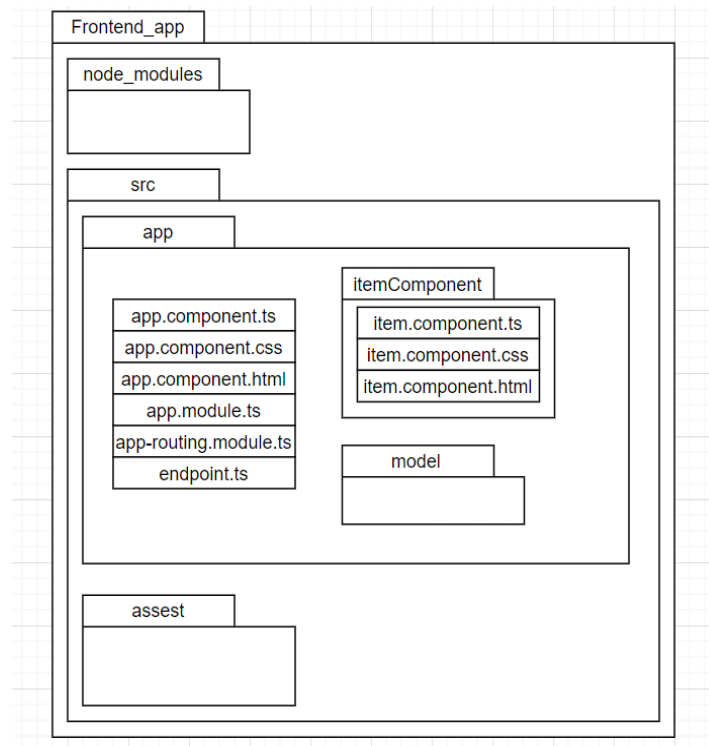


Figura 6. Estructura de frontend del proyecto

Componentes Principales:

- **app.component.ts:** Clase TypeScript del componente raíz, que juega un papel fundamental en la lógica de la aplicación. La clase definida en este archivo se encarga de manejar los principales comportamientos y la gestión de datos a nivel de la aplicación, siendo el punto de partida de la aplicación Angular.
- **app.component.css:** Este archivo de hoja de estilo está diseñado específicamente para el componente raíz, proporcionando un estilo visual consistente para la aplicación. A través de las reglas CSS, define la apariencia y el estilo visual general de la interfaz de la aplicación.
- **app.component.html:** Este archivo HTML define la plantilla de vista del componente raíz. Constituye la estructura de diseño principal de la aplicación, interactuando con la clase TypeScript a través de la sintaxis de plantillas de Angular.
- **app.module.ts:** Como módulo central de la aplicación Angular, este archivo es responsable de organizar y declarar los componentes, servicios y otras dependencias utilizadas en la aplicación. Es la clave de la arquitectura modular, asegurando una distribución e integración adecuadas de las unidades funcionales de la aplicación.
- **app-routing.module.ts:** Este archivo es responsable de definir y gestionar la configuración de rutas de la aplicación. Es la base para la navegación de páginas y la gestión de rutas en una aplicación de una sola página (SPA), permitiendo controlar la presentación y la lógica de navegación de diferentes componentes.
- **endpoint.ts:** Especifica los puntos de acceso de API utilizados para la interacción entre la aplicación y los servicios backend. Actúa como un puente para la comunicación de datos y lógica de negocio en la arquitectura de frontend y backend separados.

Estructura del Directorio del Proyecto:

- **app/:** Directorio principal del proyecto, que contiene todos los componentes de Angular, incluido el componente raíz y todos sus componentes secundarios. Esta estructura de directorio refleja el enfoque de desarrollo basado en componentes, donde cada componente tiene su propio alcance y responsabilidad.
- **components/:** Este directorio se utiliza para almacenar los componentes básicos que construyen la interfaz de usuario de la aplicación. Cada componente generalmente incluye su propio modelo, estilo y lógica, contribuyendo juntos a la interfaz de usuario de la aplicación.
- **models/:** Utilizado para almacenar los modelos de datos e interfaces de la aplicación. Estas clases TypeScript o interfaces definen la estructura de datos que se maneja y se pasa en la aplicación, siendo crucial para garantizar la seguridad del tipo y la coherencia de los datos.
- **assets/:** Almacena todos los recursos estáticos utilizados en la aplicación, como imágenes, iconos, archivos de fuente, etc. Estos recursos son fundamentales para construir los elementos visuales de la aplicación, mejorando la experiencia del usuario.



Esta forma de organización no solo mejora la mantenibilidad y escalabilidad del código, sino que también, a través de los principios de diseño modular y basado en componentes, respalda las complejas necesidades de desarrollo de la aplicación. Además, esta estructura promueve la baja dependencia entre funciones, lo que facilita una colaboración y gestión de proyectos más eficientes.

Burn-down chart

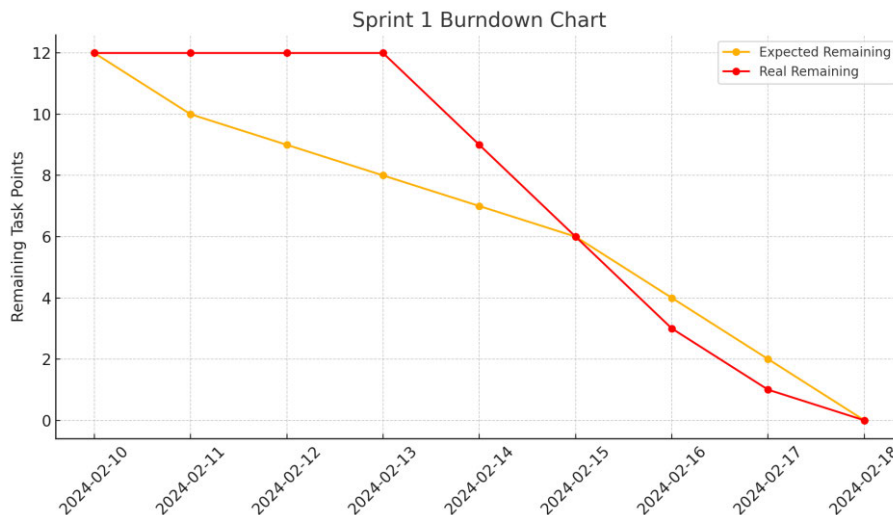


Figura 7. Burn down chart de Sprint 1

Debido a problemas de red, nuestro equipo no pudo configurar las dependencias del proyecto, lo que provocó un estancamiento del progreso hasta el 15 de febrero. Afortunadamente, las tareas posteriores fueron relativamente simples y tuvimos más tiempo disponible, por lo que pudimos completar este sprint dentro del plazo establecido. Al analizar la situación, nos dimos cuenta de que no habíamos previsto adecuadamente los problemas potenciales antes de este sprint. Esto demuestra que los riesgos inesperados pueden tener un gran impacto en el progreso del proyecto. Para evitar que este problema vuelva a ocurrir, hemos decidido discutir y evaluar exhaustivamente los posibles problemas imprevistos antes de futuros sprints.

3. Retrospective

Lo que fue bien:

- Hemos conseguido configurar y utilizar todas las dependencias necesarias tanto en el frontend como en el backend a pesar de los desafíos iniciales.

Lo que se puede mejorar:

1. **Lentitud en el acceso a GitHub:** Dado que el desarrollo se realiza desde China continental, la velocidad de subida y descarga desde GitHub es notoriamente lenta, afectando el progreso del proyecto.
2. **Tiempo de descarga de dependencias extendido:** La configuración de las dependencias se ve afectada negativamente por la ubicación geográfica, prolongando el proceso de descarga y retrasando aún más el progreso.

3. **Imprecisión del temporizador:** El uso de temporizadores tradicionales para el seguimiento del tiempo de desarrollo del proyecto no proporciona una medición precisa de la duración real del desarrollo.

Acciones a tomar:

1. **Mejorar la velocidad de acceso a GitHub:**

- Usar VPN: Implementar una conexión VPN para mejorar significativamente la velocidad de acceso a GitHub, optimizando así la experiencia de subida y descarga de código.

2. **Optimizar la gestión de dependencias:**

- Configurar fuentes nacionales: Establecer espejos de repositorios de dependencias dentro de China para acelerar la velocidad de descarga de las mismas.

3. **Mejorar la precisión del seguimiento del tiempo:**

- Adoptar herramientas de seguimiento de tiempo automatizadas: Integrar herramientas de seguimiento de tiempo automatizadas dentro del entorno de desarrollo para registrar con precisión el tiempo real invertido en el desarrollo del proyecto.



SPRINT 2

1. Sprint Planning

Periodo: 19/02/2024 - 05/03/2024

Participantes: JIAMING SHI, SHENHUI CHENG

Tarea principal: El objetivo principal de este sprint es desarrollar y mejorar los módulos asociados con las funciones de roles de usuario y la gestión de la información de la biblioteca. Nos enfocaremos en el desarrollo de un robusto mecanismo de autenticación de usuarios para garantizar la seguridad y la confiabilidad de las cuentas. Esto será fundamental para una gestión efectiva de las identidades y permisos de los usuarios. Además, implementaremos mejoras en la gestión y visualización de la información de la biblioteca, buscando ofrecer una experiencia de usuario más enriquecedora y conveniente.

Abreviatura:

- ID: ID de la historia de usuario
- T-ID: ID de la tarea
- Est.: esfuerzo estimado
- TR: tiempo real consumido

Historias de usuario involucradas en este sprint:

ID	Título	T-ID	Tarea	Est.	TR
7	Inicio de sesión de usuarios	1007	Implementar función de inicio de sesión con autenticación en el backend	5	9.5h
		1008	Implementar interfaz y función de inicio de sesión en el frontend	3	5h
8	Registro de usuarios	1009	Implementar función de registro de usuario en el backend	5	7.5h
		1010	Implementar interfaz y función de registro de usuario en el frontend	2	4.3h
9	Gestión de usuarios	1011	Implementar gestión de roles de usuario en el backend	5	7.5h
		1012	Implementar interfaz y función de gestión de usuarios en el frontend	5	7.2h
10	Personalización de menús basada en roles	1013	Mostrar menús diferentes según el rol del usuario en el frontend	2	3.5h
11	Gestión de la información de biblioteca	1014	Implementar la gestión de la información de biblioteca en el backend	3	7h
		1015	Implementación interfaz y funcionalidades para gestionar la información de biblioteca en el frontend	2	5.5h
12	Visualización de Información de biblioteca	1016	Implementar la funcionalidad de mostrar la información de biblioteca en el frontend	2	3.5h

2. Sprint Review

Backend

En esta iteración, hemos implementado el módulo de roles de usuario y su funcionalidad de gestión. Esta actualización permite que los usuarios con diferentes

niveles de permisos puedan gestionar los roles de usuario según sus autorizaciones. Este ajuste fino en la configuración de permisos ha traído al sistema un mecanismo de gestión de roles más eficiente y seguro, optimizando la experiencia del usuario.

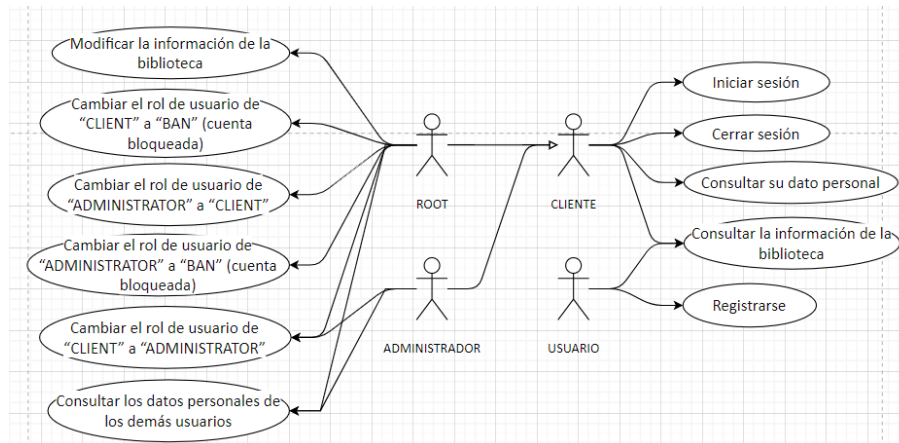


Figura 8. Diagrama de caso de diferentes roles de usuario

Cada rol dentro del sistema tiene permisos funcionales específicos, y cada cuenta está asociada a un único rol. Es importante destacar que el rol ROOT es exclusivo y no se puede transferir.

Para el sprint actual

Usuario "CLIENT" puede realizar las siguientes acciones:

- Iniciar sesión
- Cerrar sesión
- Consultar la información de la biblioteca
- Consultar su dato personal

Los usuarios "ROOT" y "ADMINISTRATOR" tienen una relación de herencia con el usuario "CLIENT", entonces las acciones que pueden realizar por el usuario "CLIENT", también las pueden realizar por los usuarios "ROOT" y "ADMINISTRATOR".

Usuario "ROOT" también puede realizar las siguientes acciones:

- Cambiar el rol de usuario de "CLIENT" a "ADMINISTRATOR"
- Cambiar el rol de usuario de "CLIENT" a "BAN" (cuenta bloqueada)
- Cambiar el rol de usuario de "ADMINISTRATOR" a "CLIENT"
- Cambiar el rol de usuario de "ADMINISTRATOR" a "BAN" (cuenta bloqueada)
- Modificar la información de la biblioteca
- Consultar los datos personales de los demás usuarios

Usuario "ADMINISTRATOR" también puede realizar las siguientes acciones:

- Cambiar el rol de usuario de "CLIENT" a "ADMINISTRATOR"
- Consultar los datos personales de los demás usuarios

Usuario "UNREGISTERED USER" puede realizar las siguientes acciones:

- Registrarse
- Consultar información de la biblioteca



Diagrama de clase

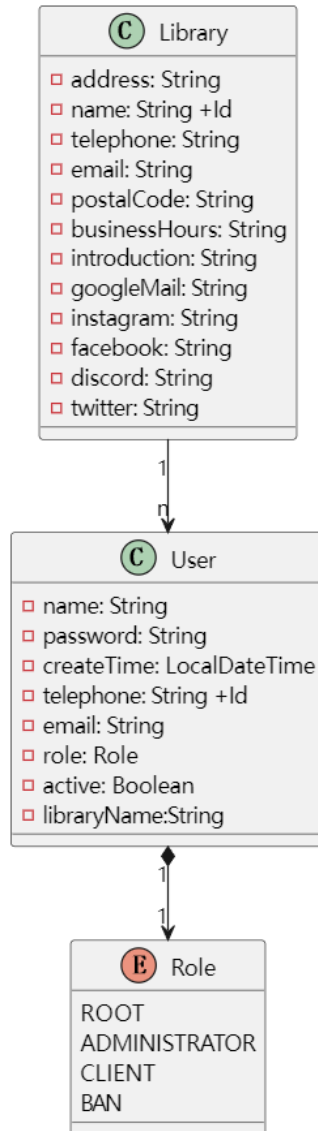
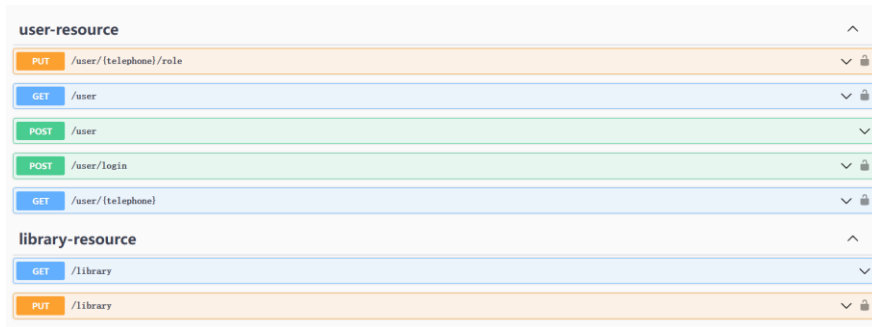


Figura 9. Diagrama de clase biblioteca-usuario

Para alcanzar los objetivos de este sprint, hemos construido con éxito tres clases claves: Usuario (User), Biblioteca (Library) y Rol (Role). La clase Usuario está diseñada para almacenar información detallada de los usuarios, mientras que la clase Rol se encarga de definir los diferentes roles y sus permisos asociados. Se ha establecido una relación uno a uno entre Usuario y Rol, garantizando que cada usuario tenga un rol claramente definido.

Por otro lado, la clase Biblioteca está diseñada para contener la información básica de la biblioteca. De acuerdo con las decisiones de negocio actuales, esta clase ha sido configurada para representar una única instancia de biblioteca, asegurando que en nuestro sistema, la entidad Biblioteca sea única y represente efectivamente la biblioteca a la que proporcionamos servicios.

APIs de Backend



user-resource	
PUT	/user/{telephone}/role
GET	/user
POST	/user
POST	/user/login
GET	/user/{telephone}
library-resource	
GET	/library
PUT	/library

Figura 10. APIs Sprint 2

1. Iniciar sesión de usuario

Endpoint: POST <https://localhost/user/login>

Descripción: Los usuarios registrados pueden verificar su identidad proporcionando un nombre de usuario y contraseña válida. Al verificar con éxito, se retorna un JWT para la autenticación y autorización en solicitudes siguientes.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Cuenta suspendida.

2. Registro de usuario

Endpoint: POST <https://localhost/user>

Descripción: Para el registro de nuevos usuarios, cada número de teléfono puede registrar únicamente una cuenta. Además, el rol predeterminado asignado a cada nueva cuenta es "CLIENT".

Códigos de error:

- 409 Conflict: El número de teléfono ya ha sido registrado.
- 422 Unprocessable Entity: Formato del número de teléfono inválido.

3. Gestión de roles de usuario

Endpoint: PUT <https://localhost/user/{id}/role>

Descripción: Los roles "ADMINISTRATOR" y "ROOT" tienen la capacidad de modificar los roles de los usuarios. Un "ADMINISTRATOR" puede elevar a un usuario de "CLIENT" a "ADMINISTRATOR". Por su parte, un "ROOT" tiene la autoridad para cambiar el rol de cualquier cuenta a "ADMINISTRATOR", "BAN" o "CLIENT".

Códigos de error:

- 400 Bad Request: Nombre de rol inválido.
- 401 Unauthorized: JWT no proporcionado.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: Usuario no existe.



4. Consulta de datos del usuario

Endpoint: GET https://localhost/user/{id}

Descripción: Consulta de datos de usuario por número de teléfono, accesible para todos los usuarios excepto aquellos usuarios "BAN".

Códigos de error:

- 401 Unauthorized: JWT no proporcionado.
- 404 Not Found: Usuario no existe.

5. Consulta de todos los datos de usuario

Endpoint: GET https://localhost/user

Descripción: "ADMINISTRATOR" y "ROOT" pueden consultar la información de todos los usuarios, autenticación requerida mediante JWT.

Códigos de error:

- 401 Unauthorized: JWT no proporcionado.
- 403 Forbidden: Permiso denegado.

6. Consulta de información de la biblioteca

Endpoint: GET https://localhost/library

Descripción: Permite a cualquier usuario consultar la información de la biblioteca.

7. Modificación de información de la biblioteca

Endpoint: PUT https://localhost/library

Descripción: Solo el usuario "ROOT" tiene permiso para modificar la información de la biblioteca, excepto el nombre de la biblioteca.

Códigos de error:

- 401 Unauthorized: JWT no proporcionado.
- 403 Forbidden: Permiso denegado.

Lógica de Backend

Análisis del proceso de inicio de sesión:

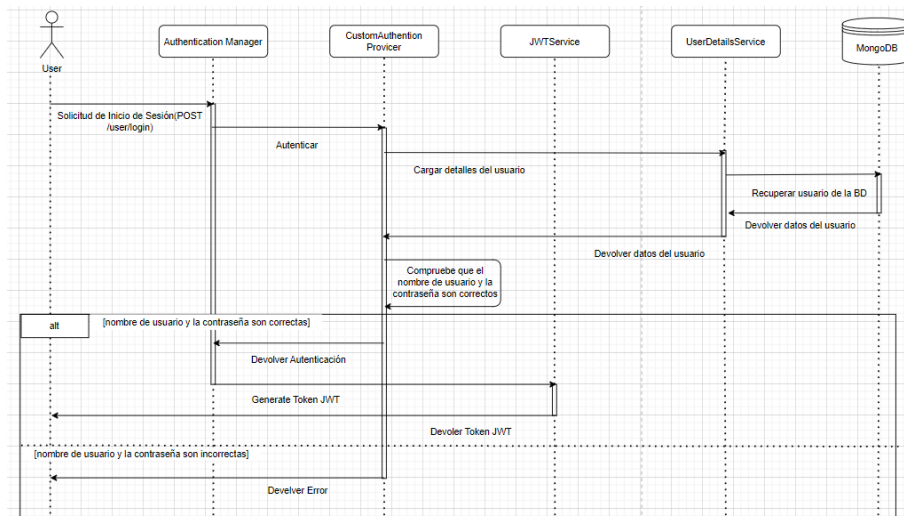


Figura 11. Diagrama de secuencia del proceso de inicio de sesión

1. Recepción de la solicitud de inicio de sesión:

El usuario envía una solicitud de inicio de sesión (generalmente una solicitud POST) al endpoint /login con los datos de inicio de sesión (como nombre de usuario y contraseña).

2. Autenticación:

El AuthenticationManager de Spring Security llama a CustomAuthenticationProvider para realizar la autenticación del usuario.

3. Obtención de datos del usuario:

CustomAuthenticationProvider interactúa con la interfaz UserDetailsService, que se comunica con la base de datos a través de MongoRepository para obtener los datos del usuario.

4. Verificación exitosa y retorno de Authentication:

Una vez que la verificación es exitosa, CustomAuthenticationProvider genera un objeto Authentication y lo devuelve al AuthenticationManager, en este punto el usuario ha completado exitosamente la autenticación. Si la verificación falla, se devuelve un error directamente al usuario.

5. Generación de JWT:

Después de completar la autenticación, el sistema generará un JWT (JSON Web Token) que lleva información de número de teléfono, nombre y rol del usuario.

6. Retorno del JWT:

El JWT generado se devuelve como respuesta al usuario.

El proceso descrito anteriormente aborda el flujo desde el inicio de sesión del usuario hasta la obtención de un token. Después de un inicio de sesión exitoso, el sistema otorga un token al usuario, que sirve como credencial para el acceso posterior a los recursos del sistema. Sin embargo, simplemente obtener el token no es suficiente



para garantizar la seguridad y validez del acceso del usuario, por lo que el siguiente paso importante es el proceso de validación del token.

La validación del token es esencial para asegurar que cada solicitud del usuario esté autorizada. Durante este proceso, el sistema necesita verificar si el token del usuario que realiza la solicitud es válido, lo cual incluye la verificación de la firma del token, si el token ha expirado y la información de los permisos del usuario, entre otros [12]. Solo cuando la validación del token es exitosa, el usuario puede acceder exitosamente al recurso o realizar la acción que solicita. De esta manera, el sistema puede prevenir efectivamente el acceso no autorizado y los potenciales riesgos de seguridad.

Análisis del proceso de secuencia de autenticación:

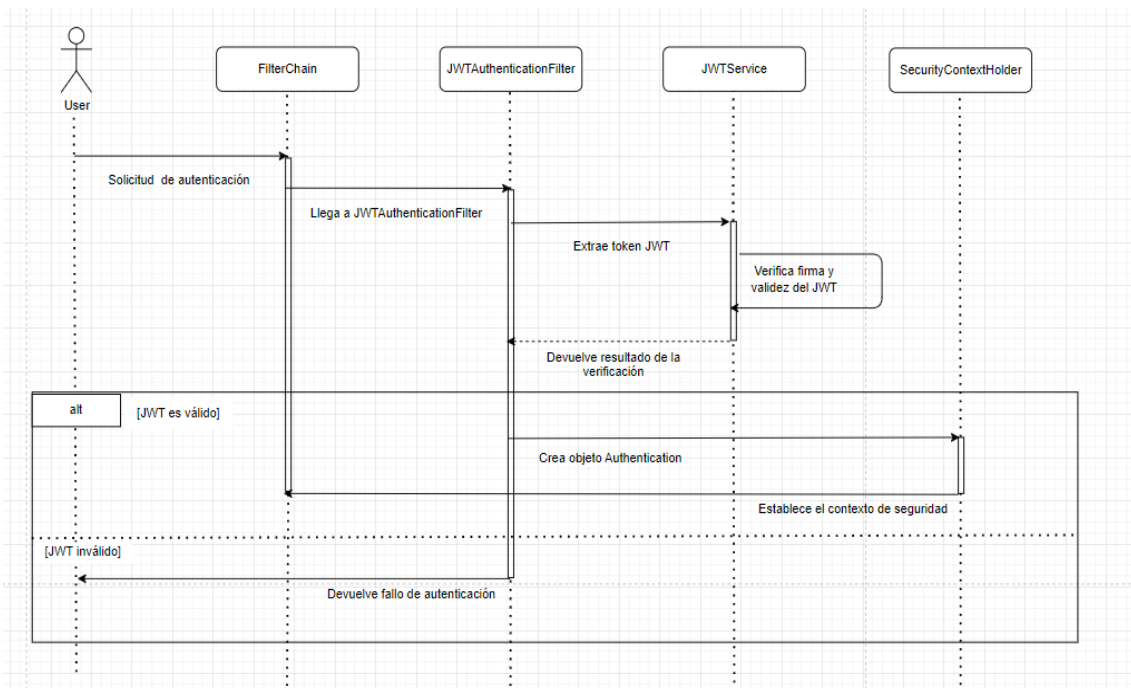


Figura 12. Diagrama de secuencia de autenticación

1. Interceptar solicitudes:

Cuando un usuario realiza una solicitud que requiere autenticación, la cadena de filtros de Spring Security interceptará estas solicitudes. Y mediante JWTFilter verificará si la solicitud contiene un JWT.

2. Extraer el JWT:

Si la solicitud contiene un JWT (JWT en el campo Authorization del encabezado HTTP), el filtro extraerá y analizará este token.

3. Verificar la validez del JWT:

Una vez extraído el JWT, se verificará su validez a través de JWTService. Esto incluye verificar la firma del token, la validez, y la información del emisor, para asegurar que el token no haya sido alterado y provenga de una fuente confiable.

4. Establecer el contexto de seguridad:

Una vez que el JWT pasa la verificación, el sistema establecerá el contexto de seguridad de Spring Security (SecurityContextHolder) según las declaraciones contenidas en el token. En este punto, se creará un objeto Authentication, que contiene la identidad y la información de permisos del usuario, y se almacenará en el contexto de seguridad para su uso posterior.

5. Control de acceso y procesamiento posterior:

Una vez establecido el contexto de seguridad, la solicitud continúa a través de la cadena de filtros hasta llegar al recurso objetivo. En este proceso, hemos utilizado la anotación @PreAuthorize para realizar un control de acceso de grano fino. Si un usuario intenta acceder a un recurso para el cual no tiene permiso, recibirá una respuesta de error correspondiente (403 Forbidden).

Si el JWT es inválido (si ha expirado o la firma no coincide), normalmente el usuario recibirá una respuesta de error, (401 Unauthorized) indicando que la autenticación ha fallado.

Análisis del proceso de todas las solicitudes:

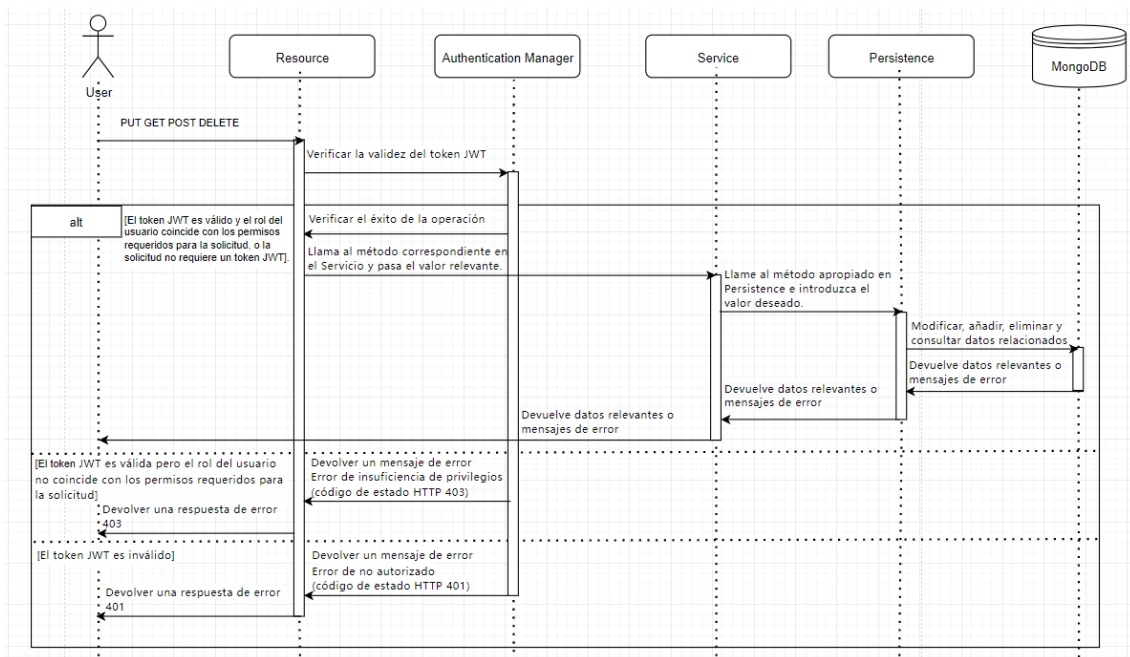


Figura 13. Diagrama de secuencia de todas las solicitudes

1. El usuario envía una solicitud, si la solicitud requiere autenticación JWT, Spring Security intercepta y verifica la validez del JWT, así como si el rol del usuario cumple con los permisos de la solicitud. Si la verificación del JWT es exitosa y el rol del usuario tiene los permisos adecuados, se permite que el Resource continúe procesando la solicitud.
2. En este momento, el Resource llama al método correspondiente en el Service y pasa el objeto relevante.
3. El Service procesa el objeto recibido y llama al método relevante en Persistence, pasando el objeto relevante.



4. Persistence interactúa con la base de datos y obtiene los datos de la base de datos correspondientes a la solicitud o devuelve un mensaje de error.
5. Persistence devuelve los datos relevantes al Service.
6. El Service devuelve los datos relevantes al Resource, los cuales luego se devuelven al usuario.

Errores posibles:

- Si la verificación del JWT falla, se devuelve al usuario el código de error 401 (401 No Autorizado).
- Si el rol del usuario no satisface los permisos necesarios para la solicitud, se devuelve el código de error 403 (403 Prohibido).
- Si el objeto de la solicitud del usuario no existe, se devuelve el código de error 404 (404 No Encontrado).
- Si los datos enviados por el usuario no cumplen con los requisitos, se devuelve el código de error 422 (422 Entidad no procesable).
- Si ocurre una excepción en el servicio, se devuelve el código de error 500 (500 Bad Request).

Frontend

Para realizar la interfaz y la funcionalidad del frontend en esta fase, hemos añadido 4 componentes.

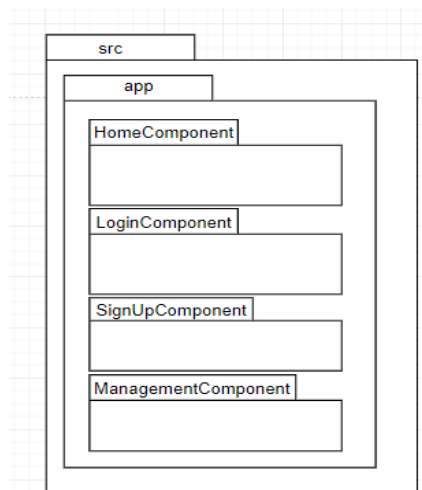


Figura 14. Diagrama de componentes de frontend de sprint 2

- HomeComponent: componente de interfaz para la página principal de la aplicación.
- LoginComponent: componente de interfaz para el inicio de sesión.
- SignUpComponent: componente de interfaz para el registro.
- ManagementComponent: componente de interfaz para la gestión de usuarios.

Análisis del Proceso de Inicio de Sesión de Usuario

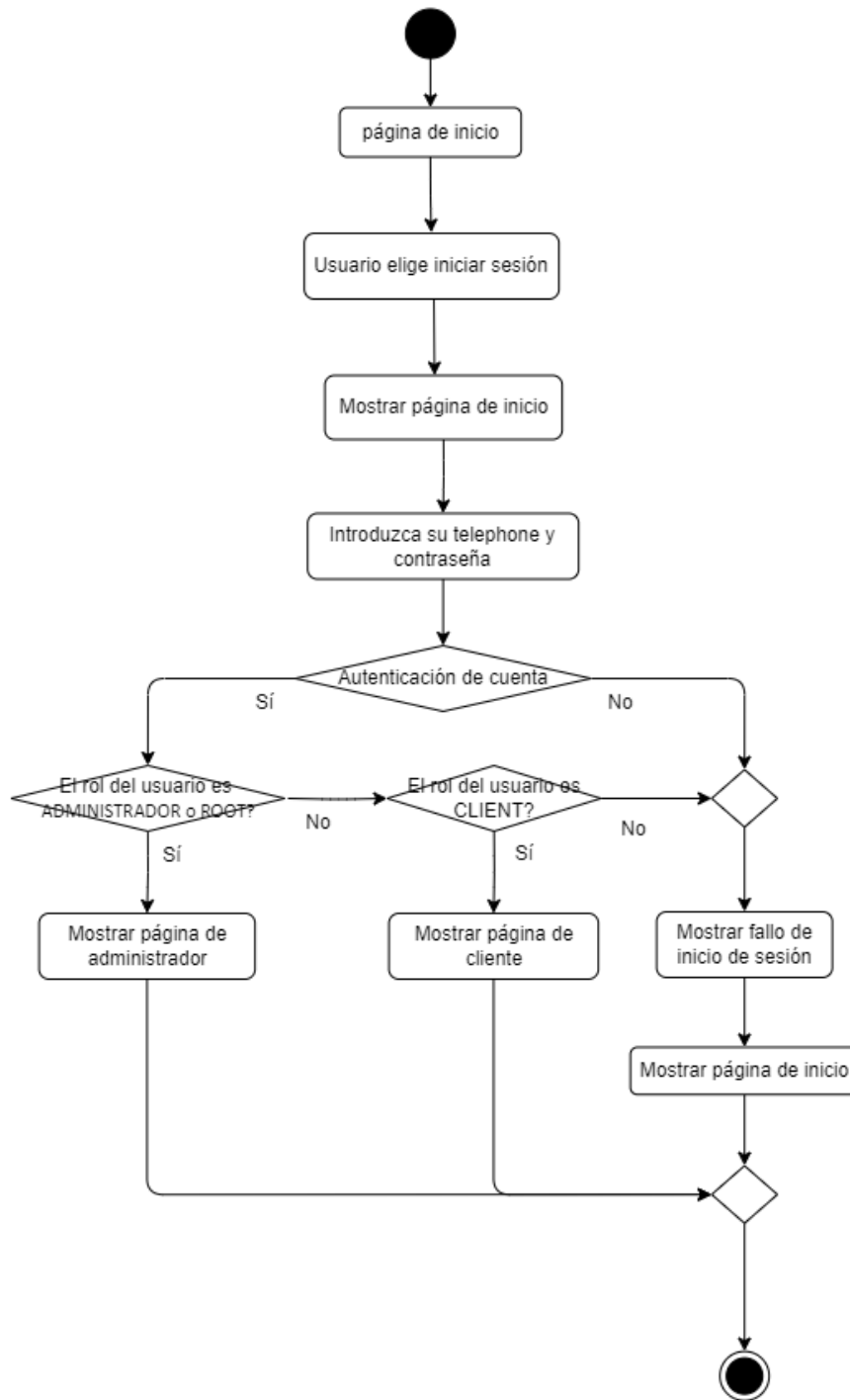


Figura 15. Diagrama de actividad de inicio de sesión de usuario

El objetivo principal del proceso de inicio de sesión es verificar la identidad del usuario y proporcionar una interfaz de usuario adecuada según el rol del usuario. A continuación, se describen los pasos específicos de este proceso:

1. **Inicio del Proceso:** El usuario selecciona “Iniciar sesión” desde el menú de la página principal, esto marca el inicio del proceso de inicio de sesión.
2. **Presentación de la Interfaz:** El sistema de front-end responde la acción del usuario mostrando la página de inicio de sesión. Esta página solicita al usuario que ingrese su número de teléfono y contraseña como credenciales de inicio de sesión.



3. **Ingreso de Información:** El usuario ingresa su número de teléfono y contraseña en la página de inicio de sesión según se indica.
4. **Proceso de Verificación:** Una vez que se envían las credenciales de inicio de sesión, el sistema de back-end recibe esta información para la verificación de la cuenta. Este paso verifica si las credenciales del usuario coinciden con los registros del sistema.
5. **Análisis de Rol y direccionamiento de página:**
 - **Acceso de Administrador:** Si el resultado de la verificación muestra que el rol del usuario es “ADMINISTRATOR” o “ROOT”, el sistema guía al usuario a una interfaz exclusiva para ADMINISTRATOR.
 - **Acceso de Cliente:** Si el usuario es identificado como “CLIENT”, se presenta la interfaz exclusiva para clientes.
 - **Acceso Restringido:** Si el rol del usuario no se puede reconocer o la cuenta está marcada como “BAN”, el sistema muestra una página de error de inicio de sesión y ofrece la opción de regresar a la página principal.
6. **Resultado del Inicio de Sesión:**
 - **Inicio de Sesión Exitoso:** Para los usuarios cuya verificación de identidad es exitosa y el rol es reconocido, el sistema carga y muestra la página principal adecuada según el rol del usuario.
 - **Fallo en el Inicio de Sesión:** En casos donde la verificación de identidad falla o el rol no cumple con los criterios de acceso, el usuario se dirige a una página de error de inicio de sesión, con la oportunidad de regresar a la página principal.
7. **Cancelación y Fin del Proceso:** El usuario puede optar por cancelar el proceso de inicio de sesión en cualquier momento, lo cual interrumpe inmediatamente el proceso y regresa a la página principal. El proceso de inicio de sesión termina cuando el usuario logra iniciar sesión con éxito o decide regresar a la página principal voluntariamente.

Análisis del Proceso de registro de usuario:

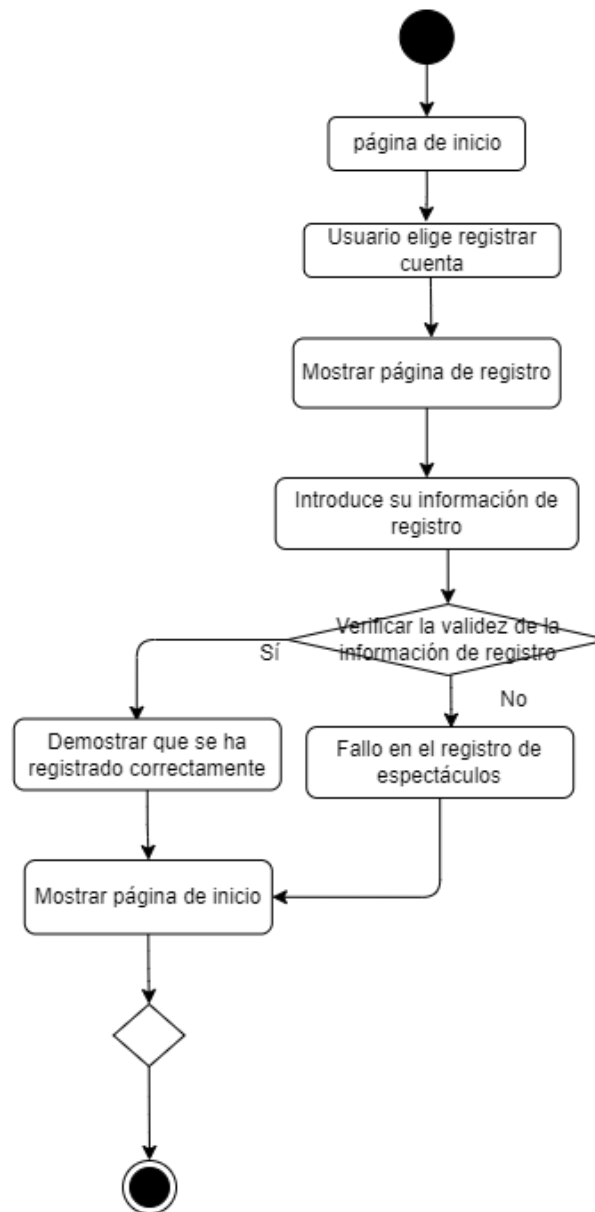


Figura 16. Diagrama de actividad de registro de usuario

El propósito principal del proceso de registro es capturar y validar la información proporcionada por el usuario, asegurando así un ingreso seguro y personalizado al sistema. A continuación, se describen los pasos detallados involucrados en este proceso:

1. **Inicio del Proceso:** El usuario desde la página principal, opta por iniciar el proceso seleccionando la opción "Iniciar sesión".
2. **Opción de Registro:** Una vez presentada la página de inicio de sesión, el usuario procede a seleccionar la opción "Registrar", lo que indica su intención de crear una nueva cuenta.
3. **Ingreso de Información de Registro:** El usuario es redirigido a la página de registro, donde se le solicita llenar los campos necesarios con su información personal, detalles de la cuenta, y cualquier otra información relevante requerida para el registro.



4. **Envío de Información de Registro:** Tras completar todos los campos necesarios, el usuario finaliza el proceso de ingreso de datos seleccionando la opción de enviar o someter su información de registro.
5. **Validación de la Información de Registro:** Al recibir la información, el sistema inicia un proceso de validación para verificar la autenticidad y completitud de los datos proporcionados por el usuario.
6. **Resultado del Registro:**
 - **Registro Exitoso:** Si la información proporcionada es validada exitosamente y cumple con los criterios requeridos, el sistema procede a procesar la solicitud de registro y posteriormente muestra un mensaje confirmando el éxito del registro.
 - **Fallo en el Registro:** En el caso de que la información sea inválida o el proceso de registro falle por cualquier razón, el sistema informará al usuario sobre el fallo del registro, ofreciendo cuando sea posible detalles específicos sobre el error.
7. **Confirmación de Registro Exitoso/Fallido:** Dependiendo del resultado del proceso de validación, el usuario recibirá un mensaje indicando si su registro ha sido exitoso o no.
8. **Opción de Cancelación:** En cualquier momento durante el proceso de registro, el usuario tiene la opción de cancelar su acción. Esta decisión interrumpe el proceso y redirige al usuario de vuelta a la página principal.
9. **Conclusión del Proceso:** El proceso de registro concluye una vez que el usuario ha completado satisfactoriamente su registro o ha elegido voluntariamente regresar a la página principal.

Prueba

Front-end

Prueba de datos de login en la página de inicio de sesión:

tel:666666666 contraseña:abc123 (Usuario que puede iniciar la sesión)

tel:999999999 contraseña:abc123 (Usuario que no existe)

tel:333333333 contraseña:abc123 (Usuario prohibido 'ban')

Test-ID	Teléfono	Contraseña	Resultado
test01	abcd	abc123	El formato del teléfono es incorrecto, no puede iniciar la sesión.
test02	666666666	abc123	Inicio de sesión exitoso.
test03	@!#\$%^&	abc123	El formato del teléfono es incorrecto, no puede iniciar la sesión.
test04	999999999	abc123	Inicio de sesión fallido, error 401
test05	333333333	abc123	Inicio de sesión fallido, error 403
test06	666666666	abc	Inicio de sesión fallido, error 401

test07	666666666		La contraseña es vacía, no puede iniciar la sesión.
--------	-----------	--	---

Prueba de datos de registro en la página de registro:

tel:666666666 (usuario que ya existe)

tel:999999999 (usuario no registrado todavía)

Test-ID	Teléfono	Contraseña	Email	Resultado
test01	abcd	abc123	abc@gmail.com	El formato del teléfono es incorrecto, no puede registrar (El formato sólo deja ser números).
test02	666666666	abc123	abc@gmail.com	Registro de cuenta fallido, 666666666 ya existe.
test03	@!#\$%^&	abc123	abc@gmail.com	El formato del teléfono es incorrecto, no puede registrar (El formato sólo deja ser números).
test04	999999999	abc123	abc	Registro de cuenta fallido, el formato de email es incorrecto.
test05	999999999	123	abc@gmail.com	Registro de cuenta fallido, el formato de contraseña no cumple con los requisitos.
test06	999999999	abc	abc@gmail.com	Registro de cuenta fallido, el formato de contraseña no cumple con los requisitos.
test07	999999999	abc123	abc@gmail.com	Registro de cuenta exitoso.

En estas dos pruebas, nos centramos en probar el formato de entrada, y realizamos pruebas de inicio de sesión con cuentas no registradas y pruebas de registro con cuentas existentes. Los resultados de las pruebas son conforme a lo esperado.

Interfaz de modificación de información de la biblioteca

Realizamos pruebas rompiendo las restricciones de acceso a la página de modificación de información de la biblioteca para el usuario ADMINISTRADOR, mediante la modificación del código en el terminal del navegador.

Test-ID	Rol	Resultado
test01	ROOT	Modificado correctamente.
test02	ADMINISTRADOR	Modificación fallida, permisos insuficientes.
test03	CLIENT	No hay manera de abrir la página.
test04	Visitante casual de la página	No hay manera de abrir la página.



Back-end

Para asegurar la alta calidad del código, hemos decidido escribir casos de prueba exhaustivos para todos los métodos, con el objetivo de lograr una tasa de cobertura del código del mínimo un 80%. Al utilizar el plugin de prueba de cobertura de código integrado en el entorno de desarrollo integrado (IDE), podemos obtener y analizar fácilmente los datos de la cobertura de pruebas, lo que garantiza de manera efectiva la confiabilidad y estabilidad del software [13].



Figura 17. Pruebas de métodos de Sprint 2

Para esta iteración, hemos construido un conjunto de datos de prueba específicos. Durante el proceso de prueba, estos datos se crearán y destruirán automáticamente, asegurando que el entorno de prueba esté completamente aislado de los datos de producción y evitando cualquier contaminación de datos potenciales. Para gestionar y mantener este conjunto de datos de prueba, hemos utilizado la configuración `@Profile("dev")`, asegurando que todas las operaciones de prueba relacionadas se ejecuten en el entorno de desarrollo.

```
@Profile("dev")  
public class SeederDev
```

Figura 18. Prueba de entorno de desarrollo

Resultados de la prueba:

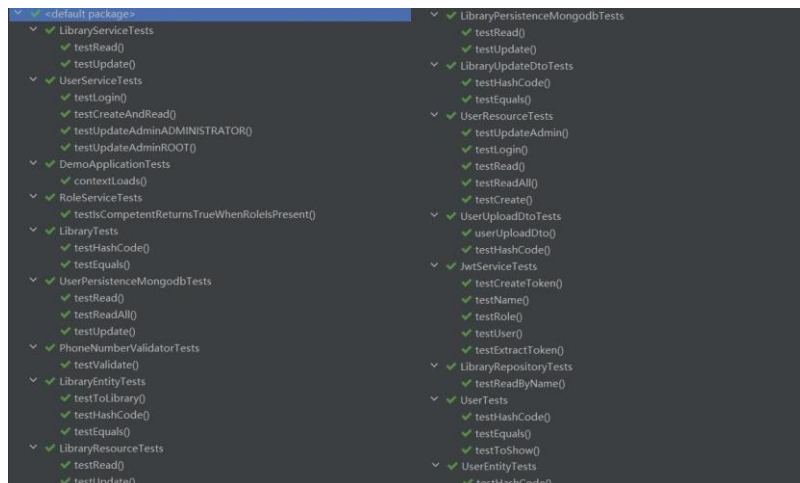


Figura 19. Resultado de las pruebas de Sprint 2

Todas las clases de prueba han pasado con éxito y la cobertura de todos los servicios (service) y métodos es del 100%, con una cobertura de líneas del 97%.

Cobertura de las pruebas:

Element	Class, %	Method, %	Line, %
JwtAuthenticationFilter	100% (1/1)	100% (2/2)	100% (8/8)
OpenApi3Configuration	100% (1/1)	100% (0/0)	100% (1/1)
SecurityConfiguration	100% (1/1)	100% (5/5)	100% (12/12)
WebConfiguration	100% (1/1)	100% (1/1)	100% (2/2)
mongodb	100% (7/7)	100% (45/45)	100% (100/100)
daos	100% (1/1)	100% (4/4)	100% (34/34)
LibraryRepository	100% (0/0)	100% (0/0)	100% (0/0)
SeederDev	100% (1/1)	100% (4/4)	100% (34/34)
UserRepository	100% (0/0)	100% (0/0)	100% (0/0)
entities	100% (4/4)	100% (32/32)	100% (37/37)
LibraryEntity	100% (2/2)	100% (18/18)	100% (20/20)
UserEntity	100% (2/2)	100% (14/14)	100% (17/17)
persistence	100% (2/2)	100% (9/9)	100% (29/29)
LibraryPersistenceMongodb	100% (1/1)	100% (3/3)	100% (12/12)
UserPersistenceMongodb	100% (1/1)	100% (6/6)	100% (17/17)
rest	100% (7/7)	95% (38/40)	93% (62/66)
dto	100% (5/5)	93% (27/29)	93% (27/29)
LibraryUpdateDto	100% (2/2)	100% (16/16)	100% (16/16)
TokenDto	100% (1/1)	50% (2/4)	50% (2/4)
UserUploadDto	100% (2/2)	100% (9/9)	100% (9/9)
LibraryResource	100% (1/1)	100% (4/4)	100% (11/11)
UserResource	100% (1/1)	100% (7/7)	92% (24/26)
domain	94% (17/18)	95% (71/74)	96% (149/155)
exceptions	85% (6/7)	82% (14/17)	81% (18/22)
ApiExceptionHandler	100% (1/1)	71% (5/7)	66% (6/9)
BadRequestException	0% (0/1)	0% (0/1)	0% (0/1)
ConflictException	100% (1/1)	100% (1/1)	100% (1/1)
ErrorMessage	100% (1/1)	100% (5/5)	100% (8/8)
ForbiddenException	100% (1/1)	100% (1/1)	100% (1/1)
NotFoundException	100% (1/1)	100% (1/1)	100% (1/1)
UnprocessableEntityException	100% (1/1)	100% (1/1)	100% (1/1)
models	100% (5/5)	100% (34/34)	100% (41/41)
Library	100% (2/2)	100% (17/17)	100% (17/17)
Role	100% (1/1)	100% (4/4)	100% (8/8)
User	100% (2/2)	100% (13/13)	100% (16/16)
persistence	100% (0/0)	100% (0/0)	100% (0/0)
LibraryPersistence	100% (0/0)	100% (0/0)	100% (0/0)
UserPersistence	100% (0/0)	100% (0/0)	100% (0/0)
service	100% (6/6)	100% (23/23)	97% (90/92)
JwtService	100% (1/1)	100% (7/7)	93% (28/30)
LibraryService	100% (1/1)	100% (3/3)	100% (7/7)
PhoneNumberValidator	100% (1/1)	100% (1/1)	100% (7/7)
RoleService	100% (1/1)	100% (1/1)	100% (6/6)
UserDetailsService	100% (1/1)	100% (3/3)	100% (9/9)
UserService	100% (1/1)	100% (8/8)	100% (33/33)

Figura 20. Cobertura de las pruebas



Burn-down chart

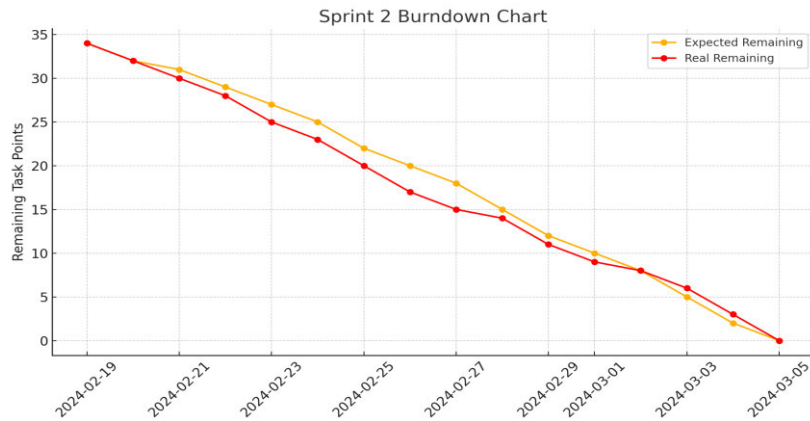


Figura 21. Burn down chart de Sprint 2

La eficiencia de ejecución de este sprint fue muy alta, el equipo completó el plan con anticipación y finalizó todas las tareas a tiempo. Las pequeñas desviaciones indican que gestionamos y asignamos muy bien el progreso de las tareas. Este es un resultado muy bueno, fue un sprint perfecto.

3. Retrospective

Lo que fue bien:

1. Uso de VPN como proxy para GitHub:

- Al emplear una VPN como proxy, se logró aumentar la eficiencia del desarrollo al mejorar la velocidad de subida y descarga del código en GitHub.

2. Implementación de herramientas de seguimiento del tiempo:

- El uso de herramientas automáticas de seguimiento del tiempo permitió registrar con precisión el tiempo de desarrollo del proyecto hasta el minuto, facilitando una mejor gestión del tiempo y recursos.

Lo que se puede mejorar:

1. Alcance de las historias demasiado amplio:

- El alcance de las historias definidas en el plan de sprint era excesivamente amplio, lo que resultó en una discrepancia significativa entre el tiempo de finalización real y el tiempo estimado inicialmente. Esto afectó la capacidad del equipo para cumplir con los compromisos del sprint de manera efectiva.

Acciones a tomar:

1. División más detallada de las historias en futuros Sprints:

- Para abordar el problema del alcance amplio de las historias, implementaremos una estrategia de división más detallada en el próximo sprint. Esto incluirá una revisión cuidadosa de las historias propuestas para asegurar que su alcance sea manejable y que las estimaciones de tiempo sean realistas. Esperamos que este enfoque mejore significativamente la precisión en la planificación y ejecución de los sprints, permitiendo al equipo alcanzar los objetivos establecidos de manera más efectiva.

SPRINT 3

1. Sprint Planning

Fecha: 07/03/2024 - 30/03/2024

Participantes: JIAMING SHI, SHENHUI CHENG

Tarea principal: Este sprint del proyecto se enfoca principalmente en fortalecer las funcionalidades de la página principal del usuario, mejorando los servicios personalizados y la experiencia del usuario. Expandimos la gestión de información de usuarios en el backend para incluir configuraciones personalizadas y funcionalidades de avatares. Paralelamente, implementamos operaciones básicas del monedero electrónico, como la simulación de recargas y retiros, y la consulta de registros de transacciones. Esto sentará las bases para el desarrollo futuro de funcionalidades relacionadas con el préstamo de libros.

Abreviatura:

- ID: ID de la historia de usuario
- T-ID: ID de la tarea
- Est.: esfuerzo estimado
- TR: tiempo real consumido

Historias de usuario involucradas en este sprint:

ID	Título	T-ID	Tarea	Est.	TR
13	Modificación de datos personales del usuario	1016	Añadir más atributos en la clase de usuario del backend	2	3.5h
		1017	Implementar la funcionalidad de modificación de información personal del usuario en el backend	3	4.5h
		1018	Implementar la interfaz y funcionalidad de modificación de información personal del usuario en el frontend	3	3.2h
14	Consulta de avatar de usuario	1019	Implementar la funcionalidad básica del avatar del usuario (consulta) en el backend	2	2.2h
		1020	Implementar la interfaz para mostrar el avatar del usuario en el frontend	2	1.8h
15	Modificación del avatar de usuario	1021	Implementar la funcionalidad básica del avatar del usuario (modificación) en el backend	3	2.2h
		1022	Implementar la interfaz y funcionalidad para modificar el avatar del usuario en el frontend	3	4.2h
16	Actualización de contraseña de usuario	1023	Implementar la funcionalidad de cambio y reseteo de contraseña del usuario en el backend	3	1.2h
		1024	Implementar la interfaz y funcionalidad de cambio y reseteo de contraseña del usuario en el frontend	3	3.4h
17	Autenticación de rol de usuarios	1025	Implementar la funcionalidad de autenticación en el backend.	3	1.6h
18	Consulta de saldo de monedero personal	1026	Implementar la funcionalidad básica del monedero personal (consulta) en el backend	3	2.4h
		1027	Implementar la interfaz y funcionalidad para la mostrar el saldo del monedero personal en el frontend	3	1.2h
19	Simulación de recarga de monedero electrónico	1028	Implementar la funcionalidad de recarga simulada del monedero electrónico en el backend	3	6.5h
		1029	Implementar la interfaz y funcionalidad para la recarga simulada del monedero electrónico en el frontend	2	4.3h
20	Simulación de retiro de monedero electrónico	1030	Implementar la funcionalidad de retiro simulado del monedero electrónico en el backend	3	5.4h
		1031	Implementar la interfaz y funcionalidad para el retiro simulado del monedero electrónico en el frontend	2	4.4h



21	Consulta de transacciones de monedero electrónico	1032	Implementar la funcionalidad de consulta de registros de transacciones en el backend	3	4.3h
		1033	Implementar la interfaz y funcionalidad para consultar los registros de transacciones en el frontend	3	4.2h
22	Envío de correo de notificación tras transacción de monedero electrónico	1034	Implementar la funcionalidad de enviar un correo electrónico de notificación tras una transacción exitosa en el backend	3	3.5h
23	Configuración personalizada de notificaciones de correo para transacciones de monedero electrónico	1034	Implementar la funcionalidad de configuración personalizada para recibir correos electrónicos sobre transacciones exitosas en el backend	2	2.2h
		1036	Implementar la interfaz y funcionalidad de configuraciones personalizadas del usuario en el frontend	2	1.2h
24	Desarrollo de página personalizada de usuario	1027	Implementar la interfaz y funcionalidad de la página principal del usuario en el frontend	5	5h

2. Sprint Review

Backend

En este sprint se han involucrado las clases de Usuario, Avatar, Cartera y Registro de Transacciones. Se implementaron funciones que permiten al usuario modificar su información personal, gestionar su avatar personalizado, administrar su cartera electrónica (incluyendo recargas y retiros), y consultar los registros de transacciones.

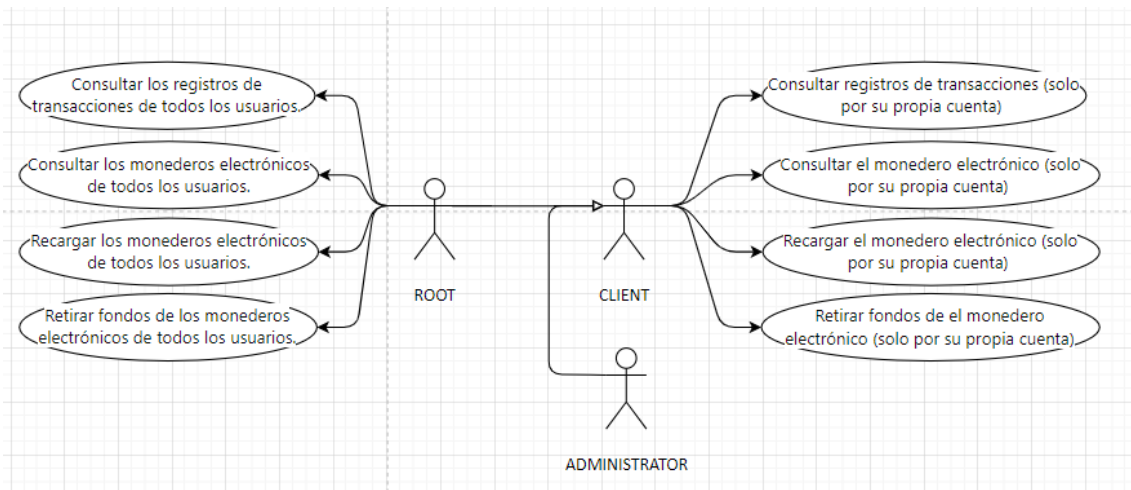


Figura 22. Diagrama de caso de uso monedero-registro de transacciones de sprint 3

Usuario “CLIENT” puede realizar las siguientes acciones:

- Consultar registros de transacciones (solo por su propia cuenta)
- Consultar el monedero electrónico (solo por su propia cuenta)
- Recargar el monedero electrónico (solo por su propia cuenta)
- Retirar fondos del monedero electrónico (solo por su propia cuenta)

Como usuario “ADMINISTRATOR” y usuario “ROOT” tiene una relación de herencia con usuario “CLIENT”, entonces se pueden realizar las mismas acciones que usuario “CLIENT”.

Además, usuario “ROOT” puede realizar las siguientes acciones:

- Consultar los registros de transacciones de todos los usuarios.
- Consultar los monederos electrónicos de todos los usuarios.
- Recargar los monederos electrónicos de todos los usuarios.
- Retirar fondos de los monederos electrónicos de todos los usuarios.

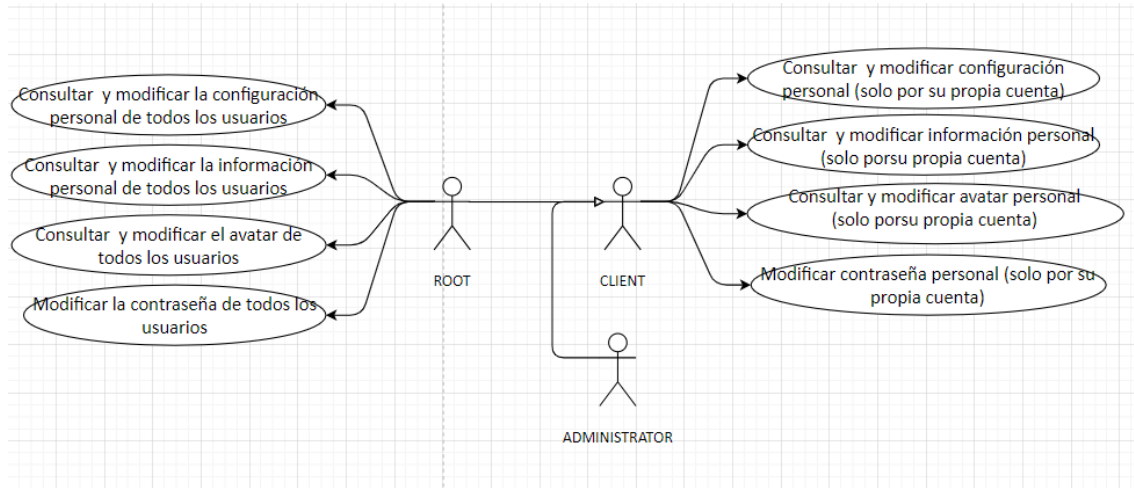


Figura 23. Diagrama de caso de uso información personal-avataar de sprint 3

Usuario “CLIENT” puede realizar las siguientes acciones:

- Consultar y modificar configuración personal (solo por su propia cuenta)
- Consultar y modificar información personal (solo por su propia cuenta)
- Consultar y modificar avatar personal (solo por su propia cuenta)
- Modificar contraseña personal (solo por su propia cuenta)

Como usuario “ADMINISTRATOR” y usuario “ROOT” tiene una relación de herencia con usuario “CLIENT”, entonces se pueden realizar las mismas acciones que usuario “CLIENT”.

Además, usuario “ROOT” puede realizar las siguientes acciones:

- Consultar y modificar la configuración personal de todos los usuarios
- Consultar y modificar la información personal de todos los usuarios
- Consultar y modificar el avatar de todos los usuarios
- Modificar la contraseña de todos los usuarios



Diagrama de clase

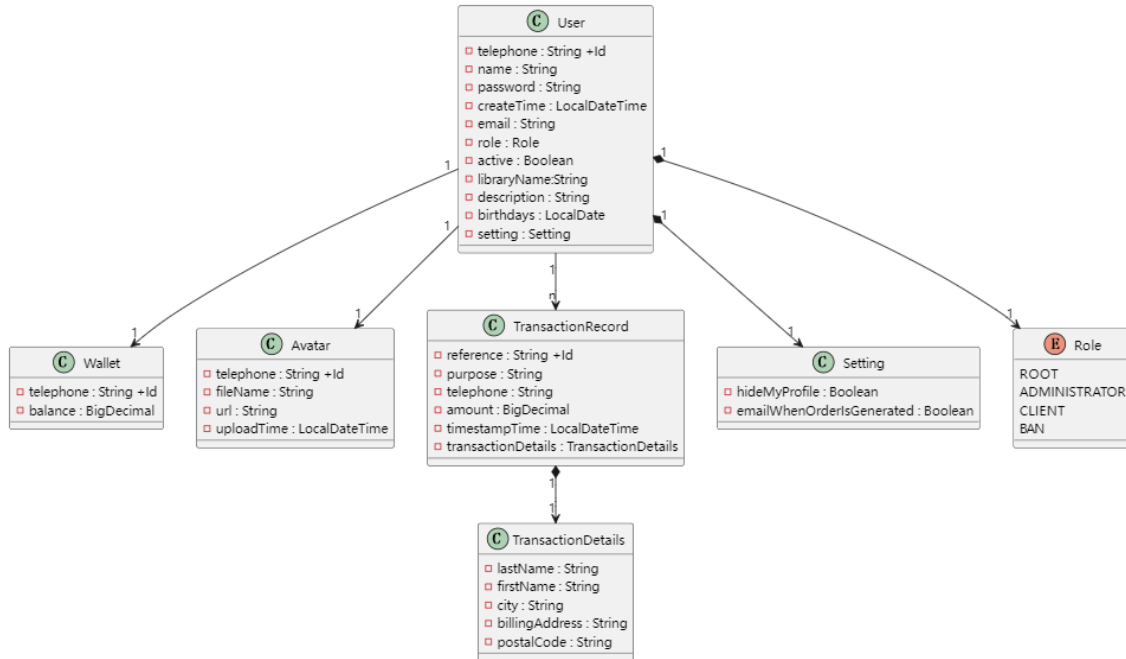


Figura 24. Diagrama de clase de Sprint 3

Para alcanzar los objetivos de este sprint, construimos cinco clases claves: Wallet (cartera), Avatar, TransactionRecord (registro de transacciones), Setting (configuración) y TransactionDetails (detalles de transacción).

La clase Wallet se utiliza para almacenar el saldo de la cartera del usuario y tiene una relación uno a uno con la clase User (usuario). Este diseño asegura que cada usuario tenga solo una cartera.

La clase Avatar se utiliza para almacenar la información del avatar del usuario y también tiene una relación uno a uno con la clase User, lo que garantiza que cada usuario tenga solo un avatar.

La clase TransactionRecord se utiliza para registrar los registros de transacciones de los usuarios. Dado que un usuario puede tener varios registros de transacciones, entonces su relación es muchos a uno con la clase User.

La clase TransactionDetails se utiliza para almacenar datos de pedidos relacionados con los registros de transacciones y tiene una relación uno a uno con la clase TransactionRecord, asegurando que cada registro de transacción tenga solo un detalle de pedido relacionado.

En esta iteración, para implementar el almacenamiento de la información personal del usuario, expandimos el contenido de la clase User, agregando campos como description (descripción personal), birthdays(cumpleaños) y setting (configuraciones).

La clase Setting se utiliza para almacenar las configuraciones personalizadas del usuario y tiene una relación uno a uno con la clase User.

APIs de Backend

PUT	/user/{telephone}	🔒
PUT	/user/{telephone}/setting	🔒
PUT	/user/{telephone}/resetPassword	🔒
PUT	/user/{telephone}/password	🔒
avatar-resource		
GET	/avatar/{telephone}	🔒
PUT	/avatar/{telephone}	🔒
wallet-resource		
POST	/wallet/withdrawal	🔒
POST	/wallet/recharge	🔒
GET	/wallet/{telephone}	🔒
transaction-record-resource		
GET	/transaction/{reference}	🔒
GET	/transaction/search	🔒

Figura 25. APIs de Sprint 3

1. Modificar la información personal de un usuario

Endpoint: PUT <https://localhost/user/{id}>

Descripción: Este puerto está destinado a la actualización de la información personal de los usuarios. Solo el propietario de la cuenta o un usuario con privilegios de 'ROOT' pueden realizar solicitudes a través de él. Permite modificar datos como la fecha de nacimiento, descripción, el nombre y la dirección de correo electrónico del usuario.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe

2. Modificar el avatar de usuario

Endpoint: PUT <https://localhost/avatar/{id}>

Descripción: Este puerto se utiliza para cambiar el avatar del usuario. Solo el propietario de la cuenta o un usuario con privilegios 'ROOT' pueden realizar solicitudes a través de este puerto. Los usuarios pueden subir archivos de imagen para el perfil en formatos JPG, PNG o GIF, con un tamaño máximo de 5 MB.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe
- 413 Max Upload Size Exceeded: Archivos cargados de más de 5 MB
- 422 Unprocessable Entity: El usuario ha cargado un archivo que no es de imagen



3. Modificar la configuración personalizada de usuario

Endpoint: PUT <https://localhost/user/{id}/setting>

Descripción: Este puerto está diseñado para modificar la configuración personalizada de un usuario. Solo el propietario de la cuenta o un usuario con privilegios 'ROOT' pueden realizar solicitudes a través de este puerto. Los usuarios pueden utilizarlo para decidir si desean recibir notificaciones por correo electrónico sobre pedidos de transacciones. Se espera que en el futuro se añadan más opciones de configuración personalizada.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe.

4. Cambiar la contraseña de la cuenta

Endpoint: PUT <https://localhost/user/{id}/password>

Descripción: Este puerto está destinado al cambio de contraseña de la cuenta. Solo el propietario de la cuenta o un usuario con privilegios de 'ROOT' pueden realizar solicitudes a través de este puerto. Permite a los usuarios cambiar la contraseña de inicio de sesión de su cuenta.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe.

5. Restablecer la contraseña de la cuenta

Endpoint: PUT <https://localhost/user/{id}/resetPassword>

Descripción: Este puerto se utiliza para que los usuarios puedan resetear la contraseña de su cuenta. Solo el usuario 'ROOT' puede realizar la solicitud. Los usuarios que hayan olvidado su contraseña anterior pueden pedir al usuario ROOT que restablezca su contraseña al valor inicial de 123456.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe.

6. Consultar el avatar

Endpoint: GET <https://localhost/avatar/{id}>

Descripción: Este puerto se utiliza para obtener el avatar del usuario, y cualquier usuario puede realizar la solicitud. Para obtener el archivo del avatar, el usuario debe proporcionar un ID correspondiente al usuario cuyo avatar desea obtener.

Códigos de error:

- 404 Not Found: El ID solicitado por el usuario no existe.

7. Consultar el historial de transacciones de un usuario

Endpoint: GET <https://localhost/transactionRecord/search?telephone={telephone}>

Descripción: Este puerto está destinado a obtener el historial de transacciones del usuario. Solo el propietario de la cuenta o un usuario con privilegios de 'ROOT' pueden realizar solicitudes a través de él. Permite a los usuarios consultar todo el historial de transacciones asociado al monedero electrónico de su cuenta individual.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.

8. Consulta de los datos de la transacción

Endpoint: GET <https://localhost/transactionRecord/{id}>

Descripción: Este puerto se utiliza para consultar los detalles de una transacción. Solo el propietario de la cuenta o un usuario con privilegios de 'ROOT' pueden realizar solicitudes a través de él. Los usuarios pueden usarlo para visualizar detalles específicos de una transacción, como el importe, la dirección de la orden, la hora, el propósito de la transacción y el nombre del comerciante, a partir de un ID determinado.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe.

9. Recargar

Endpoint: POST <https://localhost/recharge>

Descripción: Este puerto se utiliza para recargar el monedero electrónico personal. Solo el propietario de la cuenta o un usuario con privilegios de 'ROOT' pueden realizar solicitudes a través de él. Actualmente, este servicio sólo ofrece una simulación de recarga.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe.
- 422Unprocessable Entity: La cantidad de recarga del usuario no puede ser un número negativo.



10. Retirar

Endpoint: POST <https://localhost/withdrawal>

Descripción: Este puerto está destinado a las operaciones de retiro desde un monedero electrónico personal. Solo el propietario de la cuenta o un usuario con privilegios de 'ROOT' pueden realizar solicitudes. Actualmente, los usuarios pueden efectuar operaciones de retiro en modo de simulación a través de este puerto.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe
- 422Unprocessable Entity: La cantidad de retiro del usuario no puede ser un número negativo.
- 422Unprocessable Entity: El saldo del monedero del usuario es insuficiente

11. Consultar el saldo de monedero electrónico

Endpoint: POST <https://localhost/wallet/{id}>

Descripción: Este puerto es utilizado por los usuarios para consultar el saldo de su monedero electrónico. Solo el propietario de la cuenta o un usuario con privilegios de 'ROOT' pueden realizar solicitudes a través de él.

Códigos de error:

- 401 Unauthorized: Credenciales inválidas o cuenta no registrada.
- 403 Forbidden: Permiso denegado.
- 404 Not Found: El ID solicitado por el usuario no existe

Lógica de Backend

Proceso de recarga por usuario:

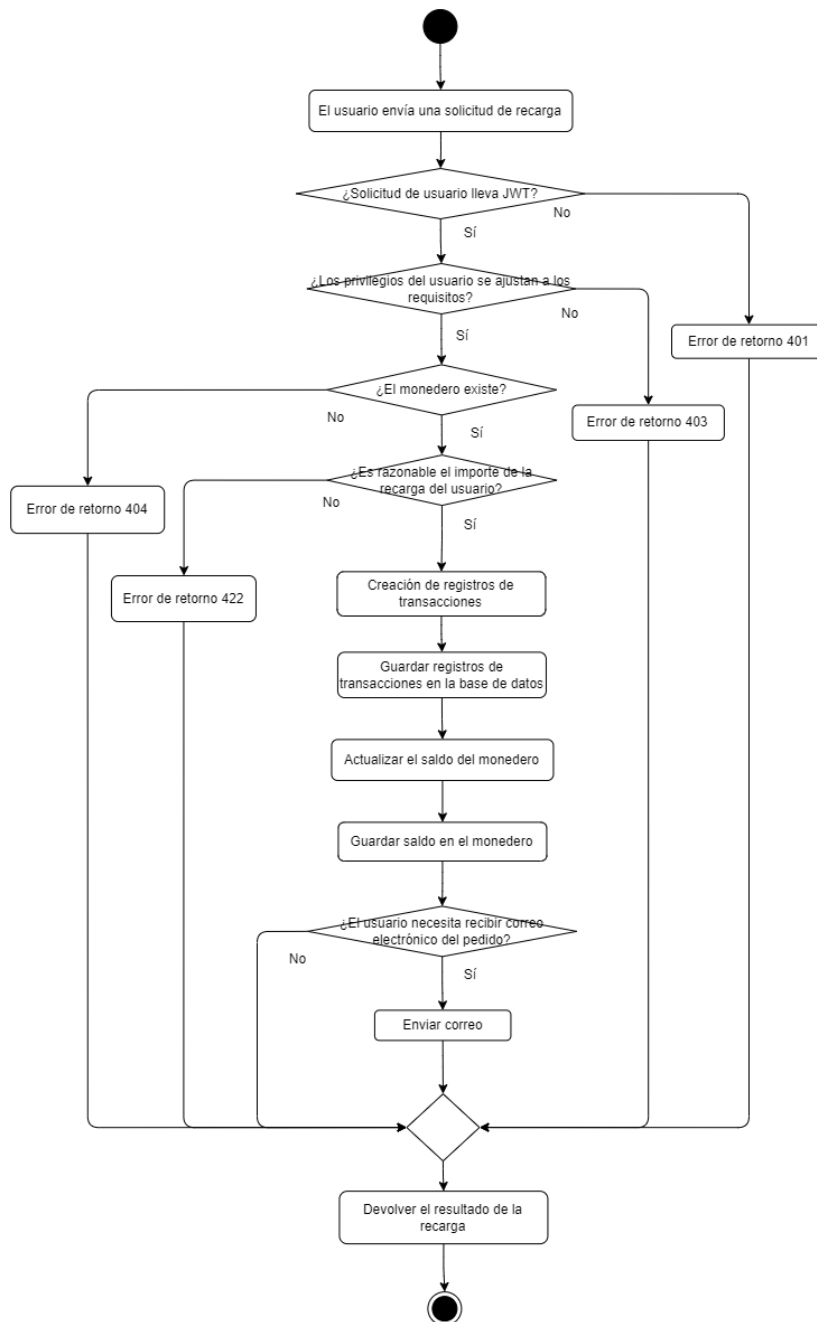


Figura 26. Diagrama de actividad de recarga de Sprint 3

1. El usuario inicia una solicitud de recarga.
2. El sistema verifica primero si el JWT en el encabezado de la solicitud enviado por el usuario es válido.
 - Si no es válido, el sistema devolverá una respuesta de error 401 Unauthorized.
3. El sistema verifica si el usuario tiene permiso para realizar recargas (si es el titular o si el rol del usuario es ROOT).
 - Si la verificación de permiso no cumple con los requisitos, el sistema devolverá un error 403 Prohibido.
4. Si se supera la verificación de autoridad, el sistema comprueba si existe el número de teléfono del usuario asociado al monedero.



- Si no existe, el sistema devolverá un error 404 Not Found.
5. Tras verificar que el número de teléfono del usuario asociado al monedero existe, el sistema evalúa si el importe de la recarga se encuentra dentro de un rango razonable.
 - Si el importe no es razonable, el sistema devolverá un error 422 Entidad no procesable.
 6. Tras confirmar que el importe es razonable, el sistema crea un nuevo registro de transacción.
 7. Una vez creado el registro de la transacción, el sistema realiza la operación de actualización del saldo del monedero.
 8. Si el usuario ha optado por recibir notificaciones por correo electrónico, el sistema enviará una notificación por correo electrónico, confirmando la recarga una vez que se haya guardado el registro de la transacción y se haya actualizado el saldo del monedero.
 9. Una vez finalizada la operación, el sistema devuelve al usuario el resultado de la operación de recarga.

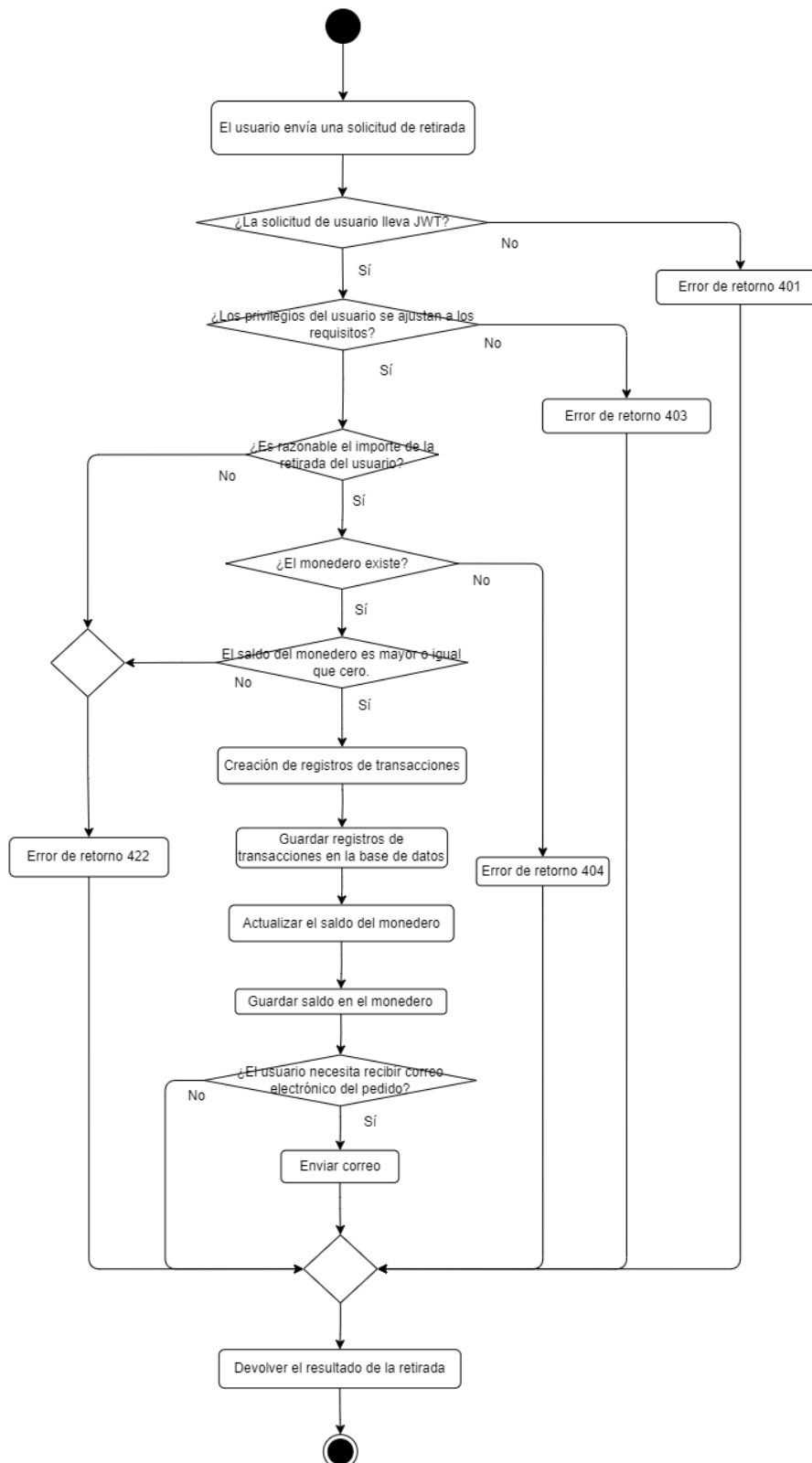
Proceso de retiro por usuario:


Figura 27. Diagrama de actividad de retiro de Sprint 3

1. El usuario presenta una solicitud de retiro de fondo.
2. El sistema verifica primero si el JWT en el encabezado de la solicitud enviado por el usuario es válido.
 - Si no es válido, el sistema devolverá una respuesta de error 401 Unauthorized.



3. El sistema verifica si el usuario tiene permiso para realizar retiros (si es el titular o si el rol del usuario es ROOT).
 - Si la verificación de permiso no cumple con los requisitos, el sistema devolverá un error 403 Prohibido.
4. El sistema evalúa si el importe de retiro de fondo solicitado por el usuario es válido.
 - Si el importe no es adecuado, el sistema devolverá una respuesta de error 422 Entidad no procesable.
5. El sistema confirma si existe el número de teléfono del usuario asociado al monedero.
 - Si no existe, el sistema devolverá una respuesta de error 404 No encontrado.
6. El sistema comprueba si el saldo del monedero del usuario cumple los requisitos de retiro.
 - Si el saldo es insuficiente, el sistema devolverá una respuesta de error 422 Entidad no procesable.
7. Una vez confirmado que el saldo es suficiente, el sistema crea un registro de transacción y lo guarda.
8. Una vez que el registro de la transacción se ha guardado correctamente, el sistema actualiza y guarda el saldo del monedero.
9. Si el usuario ha optado por recibir notificaciones por correo electrónico, el sistema envía un correo electrónico de confirmación de retirada al usuario después de que se hayan realizado todas las actualizaciones.
10. Finalmente, el sistema devuelve al usuario el resultado satisfactorio de la operación de retirada.

Actualmente en fase de desarrollo, nuestro equipo ha decidido utilizar el servicio SMTP de Google para la función de envío de correo electrónico. Esta elección se basa en la doble autenticación y encriptación SSL/TLS que proporciona, garantizando una alta fiabilidad y seguridad en la transmisión del correo. Gracias a la integración directa con el sistema de correo, podemos lograr una comunicación eficaz y segura con nuestros usuarios. Sin embargo, el servicio SMTP de Google tiene limitaciones en la frecuencia de envío cuando su uso es gratuito. Teniendo en cuenta que el crecimiento del número de usuarios en el entorno de producción puede disparar estas limitaciones en el futuro, tenemos previsto migrar a otros servidores SMTP de pago similares cuando sea necesario para garantizar la estabilidad y continuidad del servicio de correo [14].

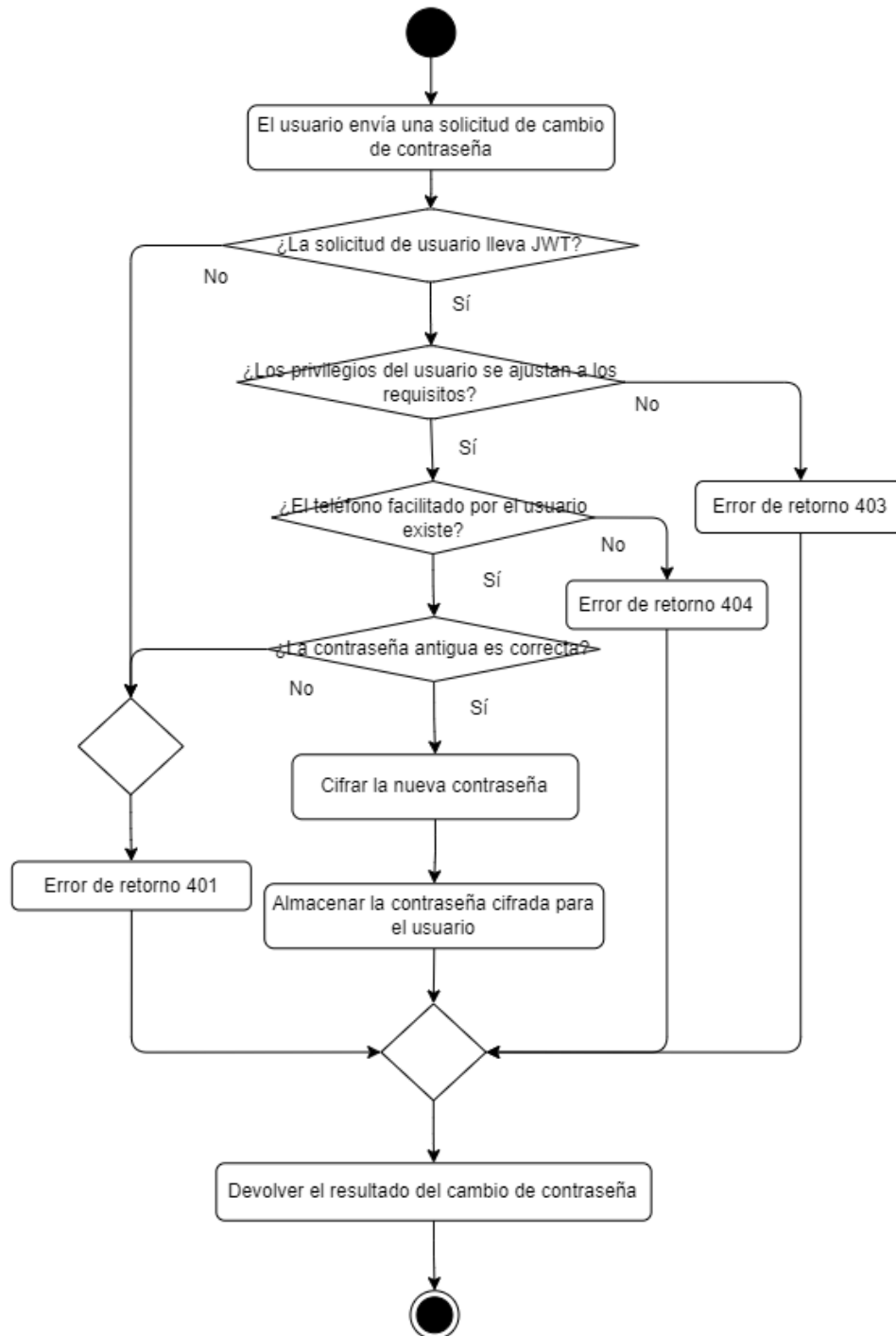
Proceso de cambio de contraseña:


Figura 28. Diagrama de actividad de cambio de contraseña de Sprint 3

1. El usuario envía una solicitud de cambio de contraseña.
2. El sistema verifica primero si el JWT en el encabezado de la solicitud enviado por el usuario es válido.
 - Si no es válido, el sistema devolverá un error 401 Unauthorized.
3. El sistema verifica si el usuario tiene permiso de modificación (si es el titular o si el rol del usuario es 'ROOT').
 - Si la verificación de permiso no cumple con los requisitos, el sistema devolverá un error 403 Prohibido.



4. Después de la verificación de autoridad, el sistema confirma si el número de teléfono móvil proporcionado por el usuario ha sido registrado en el sistema.
 - Si el número de teléfono móvil no está registrado, el sistema devolverá un error 404 Not Found.
5. Tras confirmar que el número de teléfono móvil es válido, el sistema verifica si la contraseña antigua introducida por el usuario es correcta.
 - Si la contraseña antigua es incorrecta, el sistema devolverá un error 401 No autorizado.
6. Después de verificar que la contraseña antigua es correcta, el sistema encripta la nueva contraseña introducida por el usuario.
7. Una vez finalizada la encriptación, el sistema actualizará la contraseña encriptada en la información de la cuenta del usuario.
8. Una vez que la actualización se ha realizado correctamente, el sistema devuelve el resultado al usuario y finaliza el proceso de cambio de contraseña.

Cuando los usuarios olvidan sus antiguas contraseñas pueden buscar la ayuda de ROOT, que permite a los usuarios de ROOT restablecer la contraseña del usuario a la contraseña por defecto.

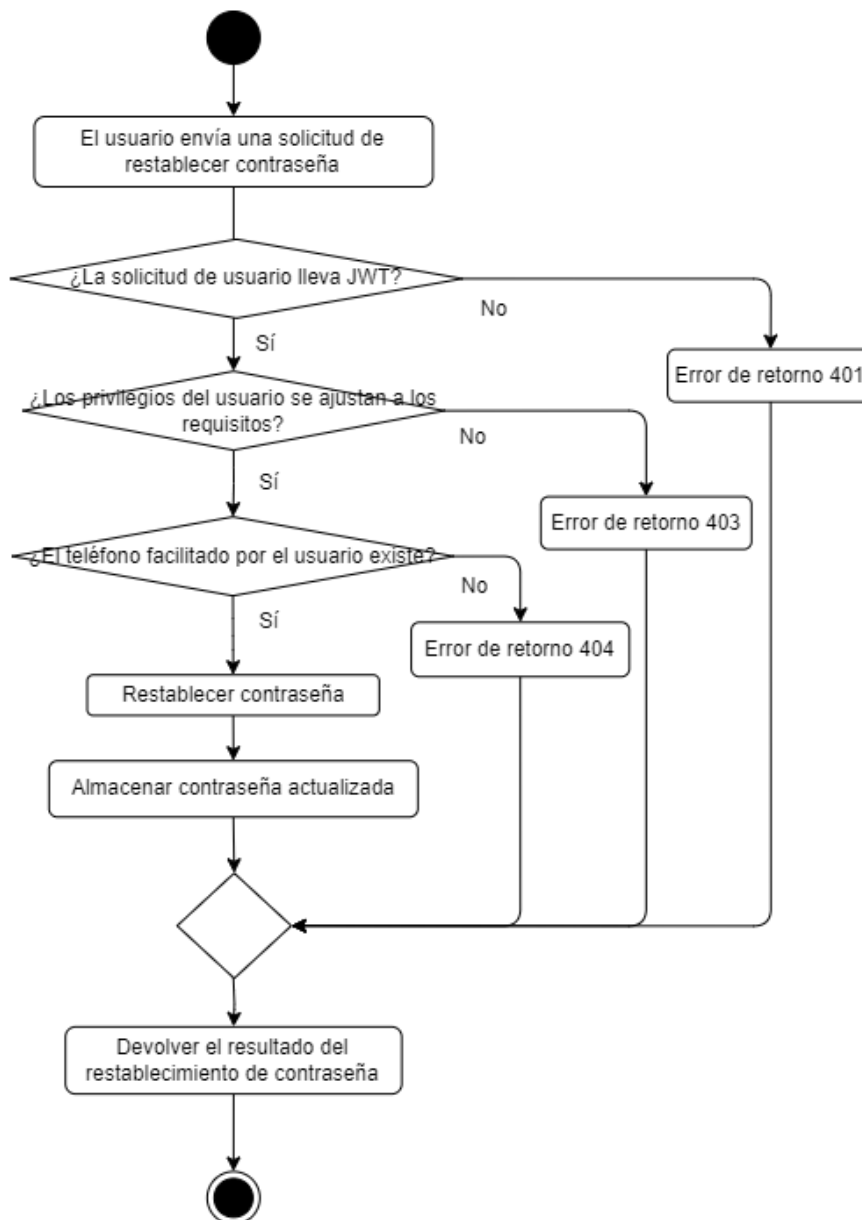
Proceso de restablecer la contraseña:


Figura 29. Diagrama de actividad de restablecer contraseña de Sprint 3

1. El usuario envía una solicitud para restablecer la contraseña.
2. El sistema verifica primero si el JWT en el encabezado de la solicitud enviado por el usuario es válido.
 - Si no es válido, el sistema devolverá un error 401 Unauthorized.
3. El sistema comprueba en primer lugar si el usuario tiene autoridad (si el rol del usuario es 'ROOT') para realizar un restablecimiento de contraseña.
 - Si el usuario no tiene suficiente permiso, el sistema devolverá un error 403 Forbidden.
4. Una vez superada la verificación de autoridad, el sistema comprueba si el número de teléfono móvil proporcionado por el usuario está registrado en el sistema.
 - Si el número de teléfono móvil no existe, el sistema devolverá un error 404 Not Found.
5. Tras confirmar que el número de teléfono móvil es válido y que la cuenta existe, el sistema restablecerá la contraseña de la cuenta.



6. Una vez restablecida correctamente, el sistema guardará de forma segura la contraseña actualizada en la información de la cuenta del usuario.
7. Una vez completados todos los pasos, el sistema devuelve al usuario el resultado del restablecimiento de la contraseña.

Frontend

Para realizar la interfaz y la funcionalidad del frontend en esta fase, hemos añadido 4 componentes.

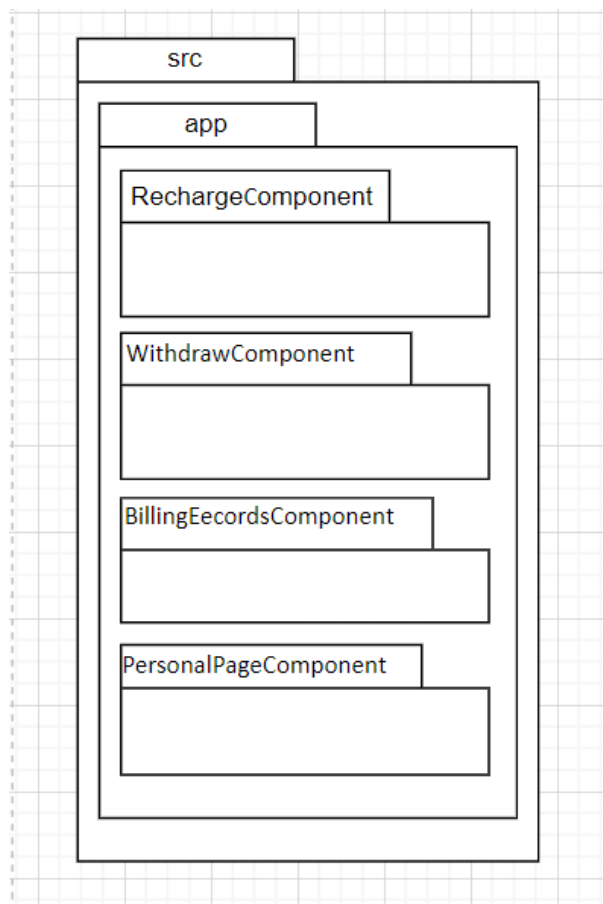


Figura 30. Diagrama de componentes de Sprint 3

RechargeComponent: componente de página para la recarga del monedero electrónico

WithdrawComponent: componente de página para retirar dinero del monedero.

BillingRecordsComponent: componente de página para mostrar los registros de transacciones del usuario.

PersonalPageComponent: componente para mostrar la interfaz personal del usuario.

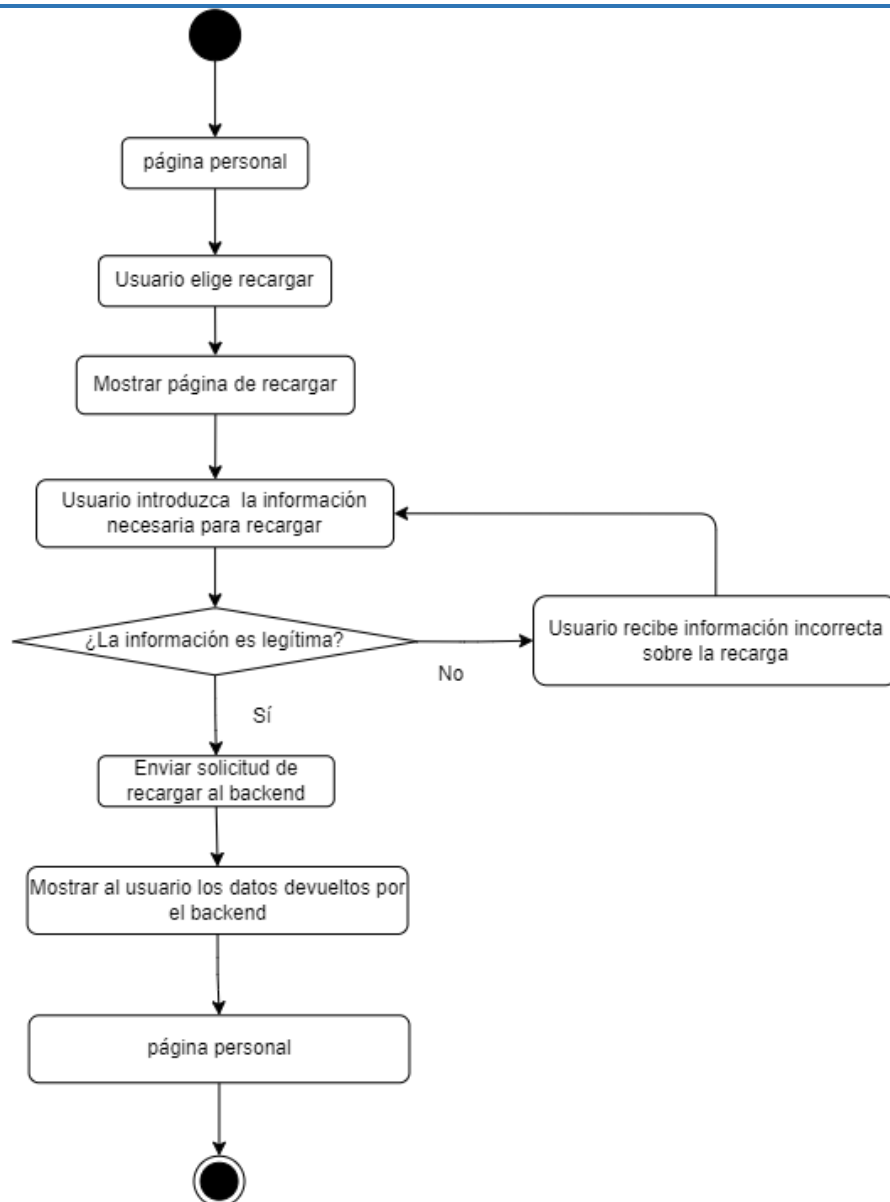


Figura 31. Diagrama de actividad de recarga saldo en frontend

Tras pulsar el botón de recargar, el usuario será dirigido a la pantalla de recarga. En esta pantalla, deberá introducir la información necesaria para la recarga, incluyendo el número de tarjeta, el CVV (valor de verificación de la tarjeta), la fecha de caducidad de la tarjeta, el importe de la recarga y cualquier información relevante del pedido, si es necesario. Una vez que el usuario haya completado la información y enviado el formulario, el sistema primero validará la legalidad del formato de los datos para asegurarse de que todos los campos obligatorios están correctamente rellenos. Tras la verificación, el sistema enviará una solicitud de recarga al servidor. Una vez que el procesamiento del servidor esté completo, el frontend recibirá y mostrará los datos devueltos por el servidor para que el usuario pueda comprender los resultados de la operación de recarga.

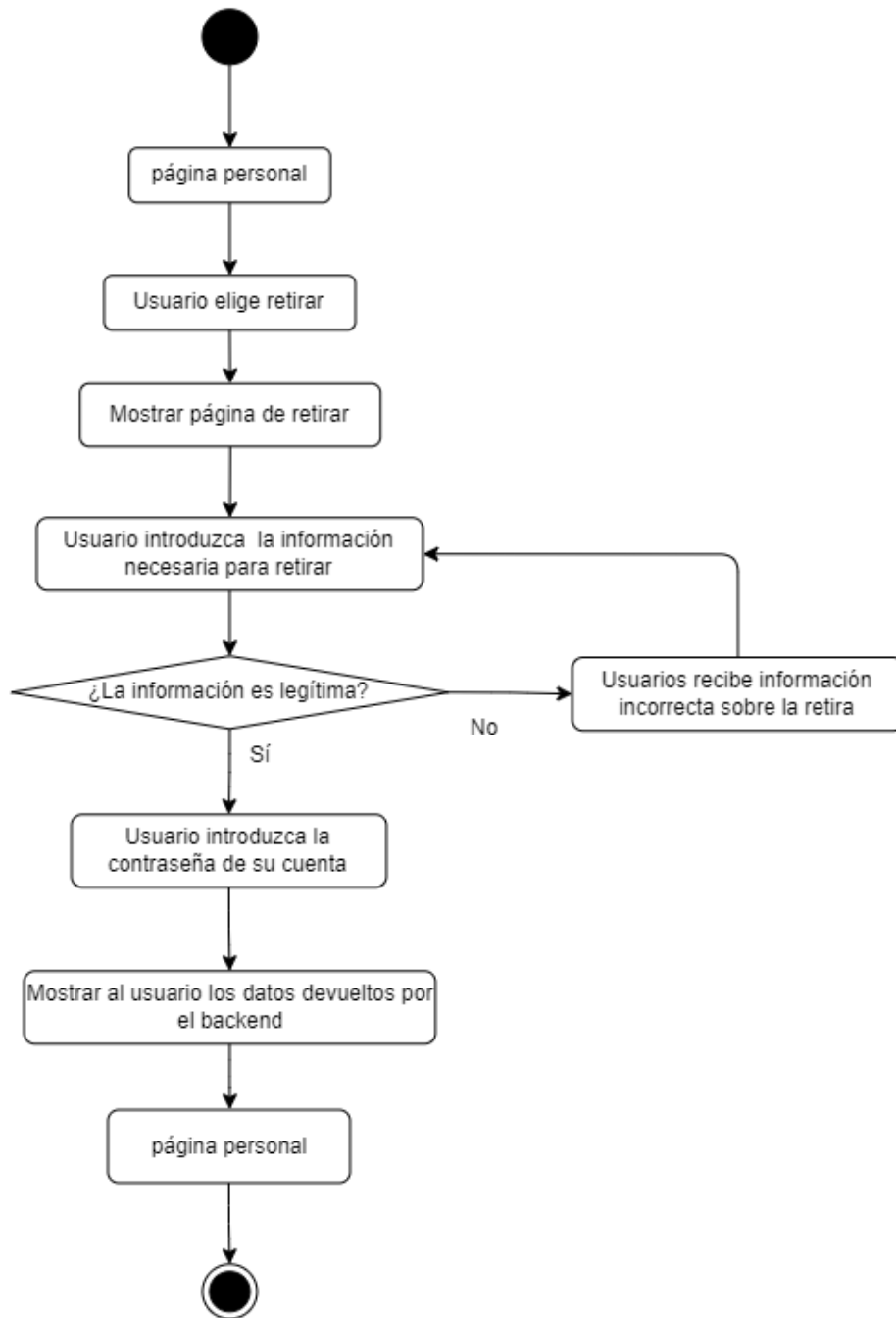


Figura 32. Diagrama de actividad de retirar saldo en frontend

Tras pulsar el botón de retirar, el usuario será dirigido a la interfaz correspondiente. En esta pantalla, debe introducir la información necesaria para la retirada, incluyendo el número de tarjeta, el importe a retirar y cualquier detalle relevante del pedido (si es necesario). Una vez completados los datos, el usuario pulsa el botón "Enviar" y el sistema le redirige a una página de autenticación segura donde se le solicita que introduzca la contraseña de su cuenta para confirmar su identidad.

Después de que el usuario introduzca correctamente la contraseña y la envíe, el sistema frontend realizará una estricta verificación de los datos proporcionados para garantizar que toda la información clave está en el formato adecuado. Una vez superada

la verificación, el sistema iniciará una solicitud de retirada de fondos al servidor. Una vez completado el procesamiento por parte del servidor, el frontend recibirá y mostrará los datos devueltos para que el usuario pueda comprender los resultados de la operación de retirada.

Prueba

Durante la última fase de pruebas de esta iteración, hemos realizado pruebas exhaustivas manuales de la función de carga de avatares. Para garantizar la estabilidad y la compatibilidad, prestamos especial atención al límite de tamaño de los archivos y a los tipos de archivos compatibles. Utilizando la herramienta Postman, simulamos varios escenarios de usuario y registramos los resultados de las pruebas.

AMT: es el que ha sido modificado

MT: modificador de avatar

El rol de **+346666666666** es ROOT

El rol de **+34666000001** es ADMINISTRATOR

El rol de **+34666000002** es CLIENT

Test No.	AMT	MT	File Name	Size	Type	Resultado
test01	+34666000001	+346666666666	File1.jpg	6.12MB	jpg	error 413
test02	+34666000001	+346666666666	File2.jpg	2.2MB	jpg	Cargado correctamente
test03	+34666000001	+346666666666	File3.png	3.32MB	png	Cargado correctamente
test04	+34666000001	+346666666666	File4.gif	4.76MB	gif	Cargado correctamente
test05	+34666000001	+346666666666	File5.avi	4.33MB	avi	error 422
test06	+34666000001	+346666666666	File6.mp4	3.85MB	mp4	error 422
test07	+34666000001	+346666666666	File7.mp3	3.78MB	mp3	error 422
test08	+34666000001	+346666666666	File8.docx	2.47MB	docx	error 413
test09	Null	+346666666666	File1.jpg	6.12MB	jpg	error 404
test10	Null	Null	File1.jpg	6.12MB	jpg	error 401
test11	Null	Null	File5.avi	4.33MB	avi	error 401
test12	+346666666666	+34666000001	File1.jpg	6.12MB	jpg	error 403
test13	+34666000001	+346666666666	File5.jpg	4.33MB	avi	error 422
test14	+34666000001	+34666000002	File2.jpg	2.2MB	jpg	error 403
test15	+346666666666	+346666666666	Null	Null	Null	error 422

Prueba de existencia: Realizamos modificaciones de avatares en usuarios existentes y los resultados mostraron que los avatares podían actualizarse con éxito. Para los usuarios no existentes, el sistema correctamente devolvió un error 404, indicando que el recurso no se encontraba.

Pruebas de permisos: Simulamos situaciones donde usuarios con diferentes roles intentaban modificar sus avatares. Los resultados demostraron que los usuarios con los roles de CLIENTE y ADMINISTRADOR pudieron cambiar sus propios avatares exitosamente, mientras que los intentos de cambiar avatares de otros usuarios generaron correctamente un error 403, indicando que los permisos estaban denegados.

Prueba de tamaño de archivos: Cargamos archivos de diversos tamaños para probar los límites del sistema. Los archivos de avatar de menos de 5 MB se modificaron sin



problemas, mientras que los archivos de más de 5 MB generaron un código de error 412, indicando que el tamaño máximo permitido había sido superado.

Pruebas de tipo de archivo: También probamos subir diversos tipos de archivo, incluidos formatos de imagen admitidos y tipos de archivo no admitidos. El sistema solo aceptó archivos de imagen para las actualizaciones de avatares. En los casos de archivos que no eran de imagen, el sistema devolvió un error 422, indicando que el tipo de archivo cargado no estaba permitido.

Específicamente, en la prueba número test14, cambiamos deliberadamente la extensión de un archivo File5.avi a jpg e intentamos subirlo. A pesar de haber cambiado la extensión, el sistema identificó correctamente el verdadero tipo de archivo y devolvió un error 422, demostrando la efectividad del sistema para prevenir que los usuarios suban archivos inapropiados simplemente cambiando la extensión del archivo.

Otros métodos para los que también hemos escrito casos de prueba:



Figura 33. Pruebas (1) de métodos de Sprint 3

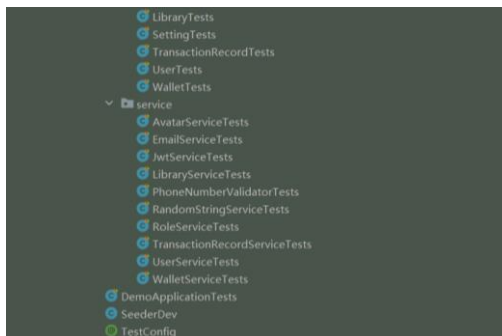


Figura 34. Pruebas (2) de métodos de Sprint 3

Resultados de las pruebas

Esta iteración terminó con una cobertura del 100% para todas las clases de servicio y métodos, y una cobertura del 98% para las filas de código. Todas las demás clases tuvieron una cobertura superior al 85%.

Package/Class	100% (x/y)	100% (x/y)	100% (x/y)
com.miw	100% (2/2)	100% (2/2)	100% (2/2)
JwtAuthenticationFilter	100% (1/1)	100% (2/2)	100% (8/8)
OpenApi3Configuration	100% (1/1)	100% (0/0)	100% (1/1)
SecurityConfiguration	100% (1/1)	100% (5/5)	100% (12/12)
WebConfiguration	100% (1/1)	100% (2/2)	100% (4/4)
mongodb	100% (15/15)	100% (86/86)	100% (149/149)
daos	100% (0/0)	100% (0/0)	100% (0/0)
AvatarRepository	100% (0/0)	100% (0/0)	100% (0/0)
LibraryRepository	100% (0/0)	100% (0/0)	100% (0/0)
TransactionRecordRepository	100% (0/0)	100% (0/0)	100% (0/0)
UserRepository	100% (0/0)	100% (0/0)	100% (0/0)
WalletRepository	100% (0/0)	100% (0/0)	100% (0/0)
entities	100% (10/10)	100% (67/67)	100% (80/80)
AvatarEntity	100% (2/2)	100% (11/11)	100% (14/14)
LibraryEntity	100% (2/2)	100% (18/18)	100% (20/20)
TransactionRecordEntity	100% (2/2)	100% (13/13)	100% (16/16)
UserEntity	100% (2/2)	100% (17/17)	100% (20/20)
WalletEntity	100% (2/2)	100% (8/8)	100% (10/10)
persistence	100% (5/5)	100% (19/19)	100% (69/69)
AvatarPersistenceMongodb	100% (1/1)	100% (3/3)	100% (9/9)
LibraryPersistenceMongodb	100% (1/1)	100% (3/3)	100% (12/12)
TransactionRecordPersistenceMongodb	100% (1/1)	100% (4/4)	100% (21/21)
UserPersistenceMongodb	100% (1/1)	100% (7/7)	100% (22/22)
WalletPersistenceMongodb	100% (1/1)	100% (2/2)	100% (5/5)

Figura 35. Cobertura de pruebas (1) de Sprint 3

Package/Class	100% (x/y)	100% (x/y)	100% (x/y)
Avatar	100% (2/2)	100% (9/9)	100% (9/9)
Library	100% (2/2)	100% (17/17)	100% (17/17)
Role	100% (1/1)	100% (5/5)	100% (13/13)
Setting	100% (2/2)	100% (7/7)	100% (7/7)
TransactionDetails	100% (2/2)	100% (10/10)	100% (10/10)
TransactionRecord	100% (2/2)	100% (11/11)	100% (11/11)
User	100% (2/2)	100% (16/16)	100% (20/20)
Wallet	100% (2/2)	100% (7/7)	100% (7/7)
persistence	100% (0/0)	100% (0/0)	100% (0/0)
AvatarPersistence	100% (0/0)	100% (0/0)	100% (0/0)
LibraryPersistence	100% (0/0)	100% (0/0)	100% (0/0)
TransactionRecordPersistence	100% (0/0)	100% (0/0)	100% (0/0)
UserPersistence	100% (0/0)	100% (0/0)	100% (0/0)
WalletPersistence	100% (0/0)	100% (0/0)	100% (0/0)
service	100% (11/11)	100% (42/42)	98% (156/159)
AvatarService	100% (1/1)	100% (3/3)	100% (4/4)
EmailService	100% (1/1)	100% (2/2)	90% (9/10)
JwtService	100% (1/1)	100% (7/7)	93% (28/30)
LibraryService	100% (1/1)	100% (3/3)	100% (7/7)
PhoneNumberValidator	100% (1/1)	100% (1/1)	100% (7/7)
RandomStringService	100% (1/1)	100% (2/2)	100% (7/7)
RoleService	100% (1/1)	100% (1/1)	100% (6/6)
TransactionRecordService	100% (1/1)	100% (5/5)	100% (25/25)
UserDetailsService	100% (1/1)	100% (3/3)	100% (9/9)
UserService	100% (1/1)	100% (13/13)	100% (51/51)
WalletService	100% (1/1)	100% (2/2)	100% (3/3)

Figura 36. Cobertura de pruebas (2) de Sprint 3

Prueba superada

<default package>	SettingUpdateDtoTests
LibraryServiceTests	SettingTests
RandomStringServiceTests	TransactionRecordTests
UserServiceTests	WalletEntityTests
WalletPersistenceMongodbTests	WalletResourceTests
LibraryTests	DemoApplicationTests
EmailServiceTests	RoleServiceTests
UserPersistenceMongodbTests	TransactionRecordResourceTests
AvatarServiceTests	WalletTests
PhoneNumberValidatorTests	WalletServiceTests
LibraryEntityTests	WalletRepositoryTests
LibraryPersistenceMongodbTests	TransactionRecordDtoTests
LibraryUpdateDtoTests	LibraryResourceTests
AvatarResourceTests	UserResourceTests
UserUpdateDtoTests	UserUploadDtoTests
JwtServiceTests	AvatarEntityTests
TransactionRecordServiceTests	PasswordUpdateDtoTests
AvatarPersistenceMongodbTests	LibraryRepositoryTests
AvatarTests	UserTests
TransactionRecordRepositoryTests	AvatarRepositoryTests
TransactionRecordEntityTests	TransactionRecordPersistenceTests
UserRepositoryTests	UserEntityTests

Figura 37. Pruebas superadas de Sprint 3



Burn-down chart

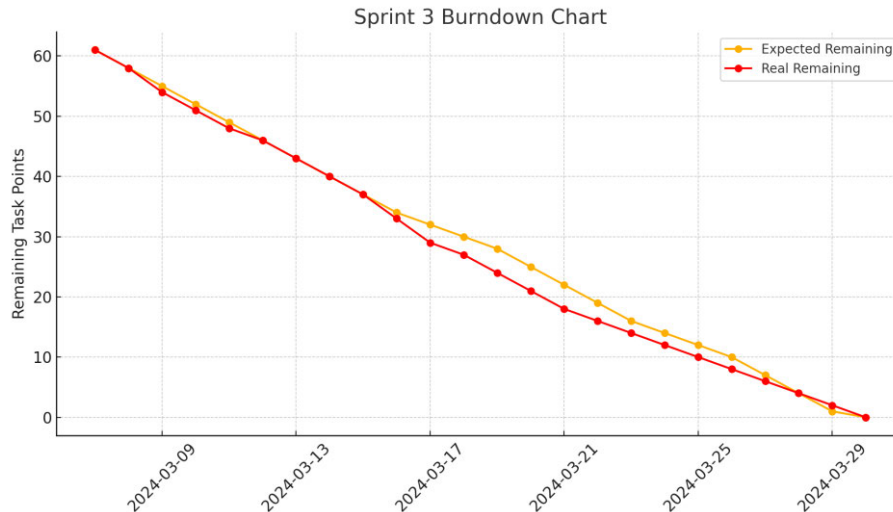


Figura 38. Burn down chart de Sprint 3

Este sprint fue exitoso, no ocurrieron incidentes inesperados, y el control del progreso de las tareas por parte del equipo fue muy preciso, con una coincidencia casi exacta entre el progreso real y las previsiones.

3. Retrospective

Lo que fue bien:

1. Refinamiento del tamaño de las historias:

- En este sprint, ajustamos el tamaño de las historias para hacer las tareas de cada historia más claras y manejables, lo cual ayudó a mejorar la organización y ejecución de las tareas.

2. Uso de herramientas automáticas de seguimiento del tiempo:

- La implementación de herramientas de seguimiento del tiempo automatizadas permitió registrar con precisión el tiempo de desarrollo del proyecto hasta el minuto, lo que facilitó una mejor gestión del tiempo y recursos.

Lo que se puede mejorar:

1. Ineficiencias en las pruebas:

- Se identificaron casos en los que parte del código se sometía a pruebas repetidas veces, lo que provocaba ineficiencias y desperdicio de recursos en las pruebas.

Acciones a tomar:

1. Modificación del orden de las pruebas:

- Para abordar las ineficiencias encontradas en el proceso de pruebas, modificaremos el orden en que se prueba el código en los próximos sprints. Esto ayudará a optimizar el uso de recursos y reducir la redundancia en las pruebas, asegurando que cada parte del código sea probada de manera eficiente y solo cuando sea necesario.

SPRINT 4

1. Sprint Planning

Fecha: 2/04/2024 - 30/04/2024

Participantes: JIAMING SHI, SHENHUI CHENG

Tarea principal: El objetivo principal de este sprint del proyecto es desarrollar y perfeccionar los módulos relacionados con las funciones de roles de usuario y la gestión de la información de la biblioteca. Nos enfocamos en desarrollar un mecanismo de autenticación de usuarios para garantizar la seguridad y fiabilidad de las cuentas, asegurando una gestión efectiva de las identidades y permisos. Además, con la implementación de sistemas para la gestión y visualización de la información de la biblioteca, buscamos ofrecer una experiencia de usuario más rica y conveniente.

Abreviatura:

- ID: ID de la historia de usuario
- T-ID: ID de la tarea
- Est.: esfuerzo estimado
- TR: tiempo real consumido

Historias de usuario involucradas en este sprint:

ID	Título	T-ID	Tarea	Est.	TR
25	Gestión de libros	1028	Implementar la creación y consulta de libros en el backend	3	3.5h
		1029	Implementar la modificación y eliminación de libros en el backend	3	3.5h
		1030	Implementar la búsqueda de libros en el backend	3	2.5h
		1031	Implementar la interfaz y funcionalidad de consulta de libros en el frontend	2	4.2h
		1032	Implementar la interfaz y funcionalidad de creación de libros en el frontend	3	5.2h
		1033	Implementar la interfaz y funcionalidad de eliminación y modificación de libros en el frontend	2	6.2h
		1034	Implementar la interfaz y funcionalidad de búsqueda de libros en el frontend	3	2h
26	Gestión de autores de libros	1035	Implementar la creación y consulta de autores de libros en el backend	2	2.2h
		1036	Implementar la eliminación y modificación de autores de libros en el backend	3	1.6h
		1037	Implementar la interfaz y funcionalidad de gestión de información de autores de libros en el frontend	3	1.8h
27	Gestión de etiquetas de categorías de libros	1038	Implementar la creación y consulta de etiquetas de categorías de libros en el backend	2	2h
		1039	Implementar la modificación y eliminación de etiquetas de categorías de libros en el backend	2	3.5h
		1040	Implementar la interfaz y funcionalidad de gestión de etiquetas de categorías de libros en el frontend	3	4.2h
28	Libro del día	1041	Implementar la funcionalidad de "Libro del Día" en el backend	1	1.5h
		1042	Implementar la interfaz y funcionalidad del "Libro del Día" en el frontend	1	1.2h
29	Préstamo de libros y reducción de saldo en monedero electrónico	1043	Implementar la funcionalidad de préstamo de libros (creación de órdenes de préstamo, deducción de saldo) en el backend	3	4.2h
		1044	Implementar la interfaz y funcionalidad de préstamo de libros (creación de órdenes de préstamo) en el frontend	2	4h



30	Configuraciones personalizadas de notificaciones de préstamo	1045	Implementar configuraciones personalizadas para recibir notificaciones por correo de órdenes de préstamo generadas en el backend	3	1.6h
		1046	Implementar la interfaz y funcionalidad del usuario para recibir la personalización del correo en el frontend	3	3.5h
31	Devolución de libros	1047	Implementar la funcionalidad de devolución de libros en el backend	2	4.5h
		1048	Implementar la interfaz y funcionalidad de devolución de libros (creación de órdenes de devolución) en el frontend	2	3.2h
32	Gestión de cajas de devolución de libros	1049	Implementar la gestión de cajas de devolución de libros en el backend	3	2.2h
		1050	Implementar la interfaz y funcionalidad de gestión de cajas de devolución de libros en el frontend	3	2.2h
33	Feedback sobre situación de devolución de libros	1051	Implementar la retroalimentación del usuario sobre la situación de devolución de libros en el backend	2	2.5h
		1052	Implementar la interfaz y funcionalidad para que el usuario proporcione retroalimentación sobre la situación de devolución de libros en el frontend	3	3.6h
34	Consulta de órdenes de préstamo y devolución	1053	Implementar la consulta de órdenes de préstamo y devolución por parte del usuario en el backend	3	3.6h
		1054	Implementar la interfaz y funcionalidad para que el usuario consulte órdenes de préstamo y devolución en el frontend	2	2h
35	Consulta de resumen de información de órdenes	1055	Implementar la consulta de resumen de información de órdenes por parte del usuario en el backend	3	2h
		1056	Implementar la interfaz y funcionalidad para que el usuario consulte resumen de los órdenes en el frontend	2	2.2h
36	Agregación y eliminación de libros a la lista de favoritos	1057	Implementar la funcionalidad para que el usuario agregue libros a su colección personal en el backend	3	2h
		1058	Implementar la funcionalidad para que el usuario elimine libros de su colección personal en el backend	2	1.5h
		1059	Implementar la funcionalidad y la interfaz para que el usuario gestione su colección de libros y la muestre en el frontend	2	3.6h
37	Página principal del autor del libro	1060	Implementar la interfaz y funcionalidad de la página principal del autor del libro (muestra la información personal del autor y sus otras obras)	3	4.4h

2. Sprint Review

Backend

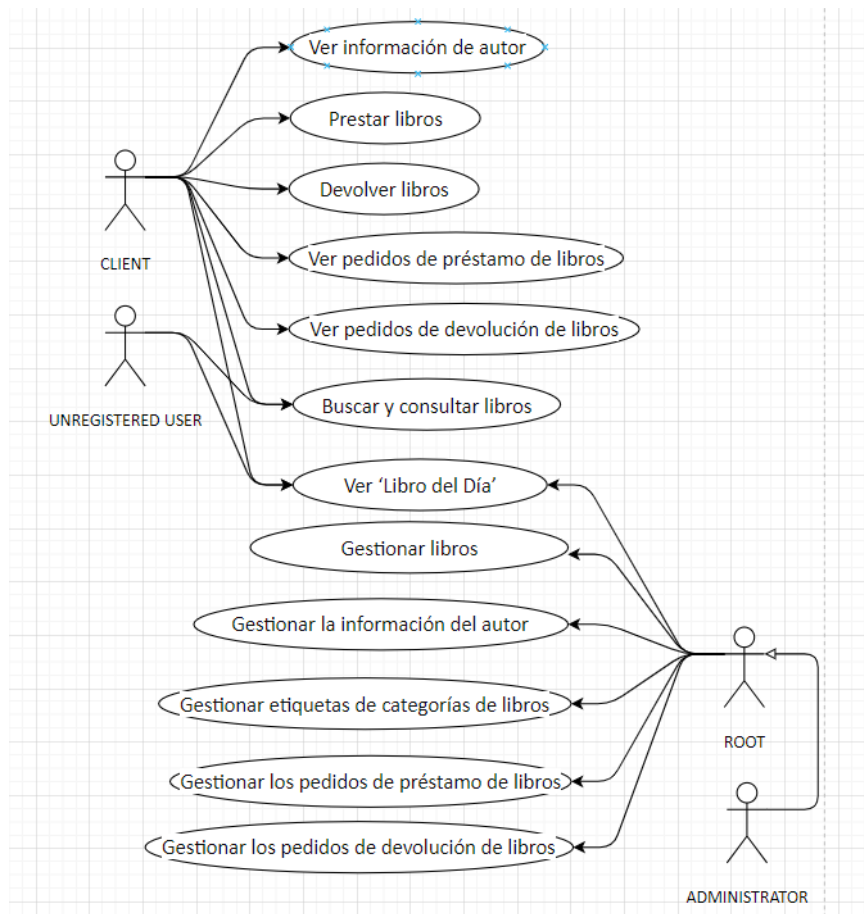


Figura 39. Caso de uso de gestión de libros de backend

Usuario "CLIENT" puede realizar las siguientes acciones:

- Prestar libros
- Devolver libros
- Buscar y consultar libros
- Ver pedidos de préstamo de libros
- Ver pedidos de devolución de libros
- Ver información de autor
- Ver 'Libro del Día'

Usuario "ROOT" puede realizar las siguientes acciones:

- Gestionar libros (añadir, eliminar, modificar, consultar, buscar)
- Gestionar los pedidos de préstamo de libros
- Gestionar los pedidos de devolución de libros
- Gestionar la información del autor (añadir, eliminar, modificar, consultar)
- Gestionar etiquetas de categorías de libros (añadir, eliminar, modificar, consultar)
- Ver el 'Libro del Día'



Como usuario "ADMINISTRATOR" hereda de usuario "ROOT", entonces se puede realizar las mismas acciones que usuario "ROOT".

Los usuarios no registrados pueden realizar las siguientes acciones:

- Buscar y consultar libros
- Ver el 'Libro del Día'

Diagrama de clase

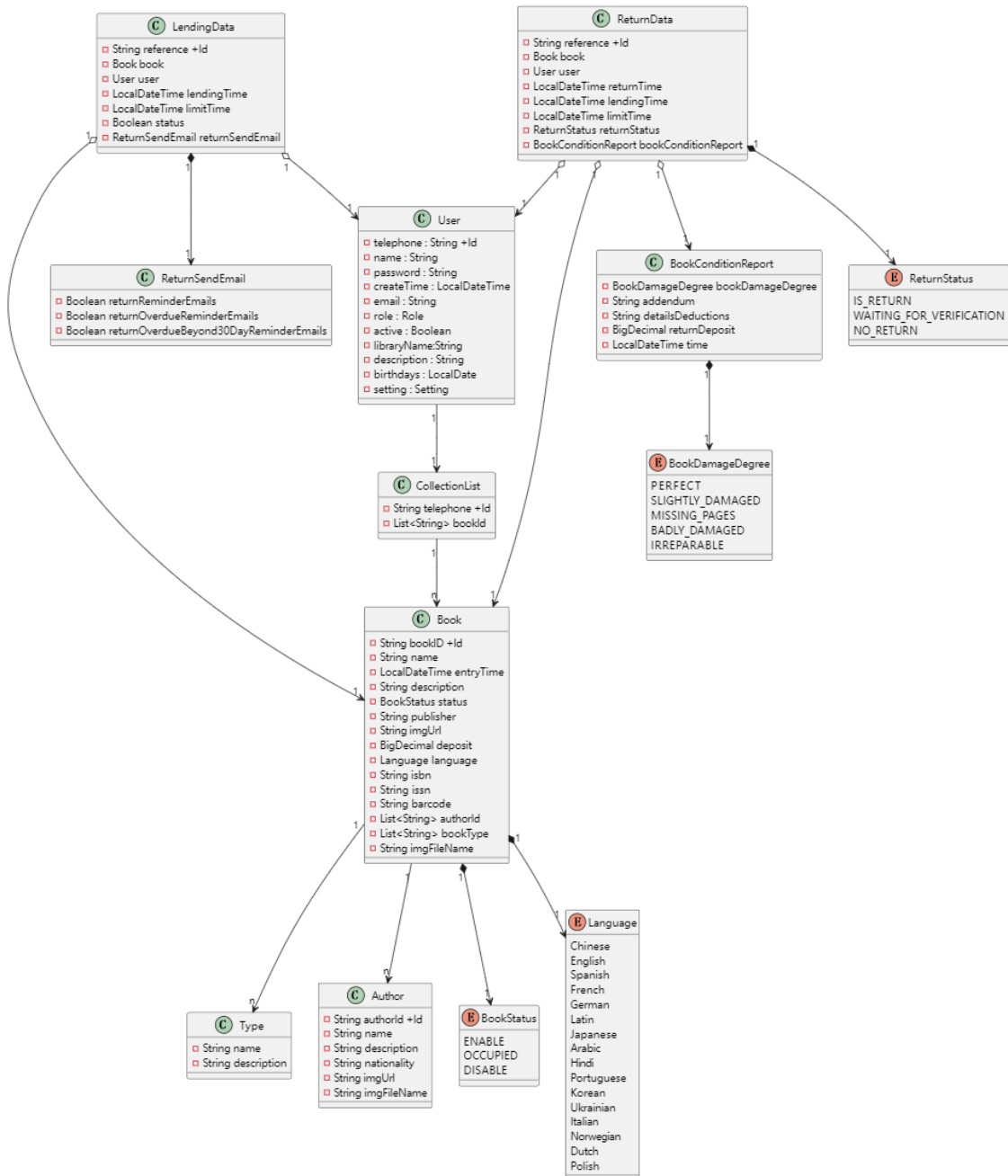


Figura 40. Diagrama de clase de Sprint 4

Para alcanzar los objetivos de este sprint, hemos construido exitosamente doce clases claves en el backend: Book, Type, Author, BookStatus, Language, LendingData,

ReturnSendEmail, BookDamageDegree, BookConditionReport, ReturnData, ReturnStatus, CollectionList.

La clase Book se utiliza para almacenar información detallada sobre los libros. Type es etiquetas de categorías de libro, Book y Type tienen una relación de uno a muchos, donde un libro puede tener uno o más etiquetas de categorías. Author es el autor del libro. Book y Author tienen una relación de uno a muchos, donde un libro puede tener uno o más autores. Language es el idioma del libro. Book y Language tienen una relación de uno a uno, donde un libro solo tiene un idioma. BookStatus es el estado del libro, Book y BookStatus tienen una relación de uno a uno, donde un libro solo tiene un estado.

La clase LendingData se utiliza para almacenar información detallada sobre las órdenes de préstamo. ReturnSendEmail es el registro de recordatorio por correo electrónico de las órdenes de préstamo. LendingData y ReturnSendEmail tienen una relación de uno a uno, donde una orden de préstamo solo tiene un registro de recordatorio por correo electrónico.

La clase ReturnData se utiliza para almacenar información detallada sobre las órdenes de devolución. ReturnStatus es la información del estado de orden de devolución. ReturnData y ReturnStatus tienen una relación de uno a uno, donde una orden de devolución solo tiene un estado. BookConditionReport es la retroalimentación de información de devolución del libro. ReturnData y BookConditionReport tienen una relación de uno a uno, donde una orden de devolución solo tiene una información de devolución del libro. BookDamageDegree es el estado de daño del libro. BookConditionReport y BookDamageDegree tienen una relación de uno a uno, donde una información de devolución del libro solo tiene un estado de daño.

La clase CollectionList se utiliza para almacenar la información de los libros favoritos del usuario.

APIs de Backend

type-resource	
GET	/type/{name}
PUT	/type/{name}
DELETE	/type/{name}
GET	/type
POST	/type

Figura 41. APIs (1) de backend de Sprint 4

return-data-resource	
PUT	/return_data/{reference}/noReturn
PUT	/return_data/{reference}/isReturn
POST	/return_data
GET	/return_data/{reference}

Figura 42. APIs (2) de backend de Sprint 4



collection-list-resource	
PUT	/collection_list/{telephone}/remove_book
PUT	/collection_list/{telephone}/add_book
GET	/collection_list/{telephone}
GET	/collection_list/{telephone}/book

Figura 43. APIs (3) de backend de Sprint 4

book-resource	
GET	/book/{bookId}
PUT	/book/{bookId}
DELETE	/book/{bookId}
PUT	/book/{bookId}/status
PUT	/book/{bookId}/image
GET	/book
POST	/book
GET	/book/search
GET	/book/random
GET	/book/all_language

Figura 44. APIs (4) de backend de Sprint 4

author-resource	
GET	/author/{authorId}
PUT	/author/{authorId}
DELETE	/author/{authorId}
PUT	/author/{authorId}/image
GET	/author
POST	/author

Figura 45. APIs (5) de backend de Sprint 4

lending-data-resource	
POST	/lending_data
POST	/lending_data/send_email_to_user_by_order_overdue
POST	/lending_data/send_email_to_user_by_approaching_date
POST	/lending_data/ban_user_by_overdue_max_30day
GET	/lending_data/{reference}
GET	/lending_data/yearly_counts
GET	/lending_data/weekly_counts
GET	/lending_data/read_lending_data_by_overdue_max_30day
GET	/lending_data/no_return/search
GET	/lending_data/monthly_counts
GET	/lending_data/lending_statistics
GET	/lending_data/client_return_and_lending/search
GET	/lending_data/admin_return_and_lending

Figura 46. APIs (6) de backend de Sprint 4

1. Crear libro

Puerto: POST https://localhost/book

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden crear nuevos libros a través de este puerto. Tras la creación exitosa, se devolverá la información del libro. Por defecto, el libro estará en estado "deshabilitado". (disable).

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 422: El formato del idioma del libro proporcionado por el usuario es incorrecto.

2. Consultar libro por BookId

Puerto: GET https://localhost/book/{bookId}

Descripción: Todos los usuarios tienen la capacidad de consultar la información detallada de un libro utilizando un bookId específico. Si el bookId se encuentra en la base de datos, se proporcionará la información completa del libro.

Códigos de error:

- 404: El bookId proporcionado por el usuario no existe.

3. Consultar todos los libros

Puerto: GET https://localhost/book

Descripción: Todos los usuarios pueden obtener información resumida de todos los libros en la biblioteca a través de este puerto.

4. Obtener todos los idiomas

Puerto: GET https://localhost/book/all_language

Descripción: Todos los usuarios pueden obtener información sobre todos los idiomas de los libros en la biblioteca a través de este puerto.

5. Modificar información del libro

Puerto: PUT https://localhost/book/{bookId}

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT tienen la capacidad de modificar la información de un libro utilizando un bookId específico. Tras una modificación exitosa, se proporcionará la información del libro actualizado.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El bookId proporcionado por el usuario no existe.
- 422: El formato del idioma del libro proporcionado por el usuario es incorrecto.

6. Modificar estado del libro

Puerto: PUT https://localhost/book/{bookId}/status

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden modificar el estado de un libro utilizando un bookId específico. Tras una modificación exitosa, se proporcionará la información actualizada del libro.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El bookId proporcionado por el usuario no existe.

7. Modificar imagen del libro

Puerto: PUT https://localhost/book/{bookId}/image

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden modificar la



imagen de presentación de un libro mediante un bookId específico. Una vez realizada la modificación con éxito, se proporcionará la información actualizada del libro.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El bookId proporcionado por el usuario no existe.
- 413: El archivo enviado por el usuario es demasiado grande.
- 422: El usuario subió un archivo que no es una imagen.

8. Eliminar libro

Puerto: DELETE <https://localhost/book/{bookId}>

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden eliminar la información de un libro utilizando un bookId específico. Una vez que la eliminación sea exitosa, se proporcionará la información del libro eliminado. Es importante destacar que el libro a eliminar no debe estar en estado "ocupado" (occupied).

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El bookId proporcionado por el usuario no existe.

9. Libro del día

Puerto: GET <https://localhost/book/random>

Descripción: Todos los usuarios pueden obtener información de un libro seleccionado al azar a través de este puerto.

10. Buscar libros

Puerto: GET

<https://localhost/book/search?name={name}&publisher={publisher}&authorName={authorName}&language={language}&type={type}&authorId={authorId}&barcode={barcode}&issn={issn}&isbn={isbn}>

Descripción: Todos los usuarios pueden obtener información de los libros que coincidan con los criterios de búsqueda proporcionados. Una solicitud exitosa devolverá una lista con información resumida sobre los libros.

11. Crear autor

Puerto: POST <https://localhost/author>

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden crear información de autores utilizando este puerto. Una vez completada la creación con éxito, se proporcionará la información del autor creado.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

12. Consultar autor por AuthorId

Puerto: GET https://localhost/author/{authorId}

Descripción: Todos los usuarios pueden acceder a información detallada de un autor utilizando un authorId específico. Si el authorId existe en la base de datos, se proporcionará la información detallada del autor.

Códigos de error:

- 404: El authorId proporcionado por el usuario no existe.

13. Consultar todos los autores

Puerto: GET https://localhost/author

Descripción: Todos los usuarios tienen la capacidad de acceder a información resumida de todos los autores de la biblioteca mediante este puerto.

14. Modificar información del autor

Puerto: PUT https://localhost/author/{authorId}

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden modificar la información de un autor utilizando un authorId específico. Tras una modificación exitosa, se proporcionará la información del autor modificado.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El authorId proporcionado por el usuario no existe.

15. Eliminar autor

Puerto: DELETE https://localhost/author/{authorId}

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden eliminar la información de un autor asociada a un authorId específico. Una vez que la eliminación sea exitosa, se proporcionará la información del autor eliminado.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El authorId proporcionado por el usuario no existe.

16. Modificar imagen del autor

Puerto: PUT https://localhost/author/{authorId}/image

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden modificar la imagen de presentación de un autor usando un authorId específico. Tras la modificación exitosa, se proporcionará la información del autor modificado.



Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El authorId proporcionado por el usuario no existe.
- 413: El archivo enviado por el usuario es demasiado grande.
- 422: El usuario subió un archivo que no es una imagen.

17. Crear etiquetas de categorías de libro

Puerto: POST <https://localhost/type>

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden crear nuevas etiquetas de categorías de libros a través de este puerto. Tras la creación exitosa, se proporcionará la información de las etiquetas de categorías creadas.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 409: Las etiquetas de categorías proporcionado por el usuario ya ha sido creado.

18. Consultar etiquetas de categorías por nombre

Puerto: GET <https://localhost/type/{name}>

Descripción: Todos los usuarios pueden obtener información detallada de una etiqueta de categoría de libros correspondiente a un nombre. Si el nombre existe en la base de datos, se proporcionará la información detallada de la etiqueta de categoría de libro.

Códigos de error:

- 404: El nombre proporcionado por el usuario no existe.

19. Modificar etiquetas de categorías de libro

Puerto: PUT <https://localhost/type/{name}>

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden modificar la información de una etiqueta de categoría de libros correspondiente a un nombre específico. Tras la modificación exitosa, se proporcionará la información de la etiqueta de categoría de libro modificada.

20. Crear orden de préstamo

Puerto: POST https://localhost/lending_data

Descripción: Los usuarios con rol de CLIENT pueden crear nuevas órdenes de préstamo a través de este puerto. Tras la creación exitosa, se proporcionará la información de la orden de préstamo creada.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 403: El saldo del usuario no es suficiente.
- 409: El libro solicitado no está disponible.
- 409: El libro solicitado ya ha sido prestado.

21. Consultar orden de préstamo por referencia

Puerto: GET https://localhost/lending_data/{reference}

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT, o CLIENT correspondiente pueden obtener información detallada de una orden de préstamo asociada a una referencia específica. Si la referencia existe en la base de datos, se proporcionará la información detallada de la orden de préstamo.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: La referencia proporcionada por el usuario no existe.

22. Consultar todas las órdenes de préstamo y devolución por teléfono

Puerto: GET

https://localhost/lending_data/client_return_and_lending/search?telephone={telephone}

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT o CLIENT correspondiente pueden obtener información resumida de todas las órdenes de préstamo y devolución asociadas a un número de teléfono específico.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El número de teléfono proporcionado por el usuario no existe.

23. Consultar todas las órdenes de préstamo, devolución y en espera de revisión

Puerto: GET https://localhost/lending_data/admin_return_and_lending

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden acceder a información resumida de todas las órdenes de préstamo, devolución y aquellas en espera de revisión mediante este puerto.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

24. Consultar todas las órdenes de préstamo no devueltas por teléfono

Puerto: GET https://localhost/lending_data/no_return/search?telephone={telephone}

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT, o CLIENT



correspondiente pueden obtener información resumida de todas las órdenes de préstamo no devueltas asociadas a un número de teléfono específico.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El número de teléfono proporcionado por el usuario no existe.

25. Consultar estadísticas de préstamo de la biblioteca

Puerto: GET https://localhost/lending_data/lending_statistics

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden acceder a estadísticas de préstamos de la biblioteca, incluyendo el total de préstamos, así como los préstamos realizados durante el año, mes, semana y día.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

26. Consultar análisis mensual de préstamos del año

Puerto: GET https://localhost/lending_data/monthly_counts

Descripción: ADMINISTRADOR y ROOT pueden obtener estadísticas de préstamo mensual del año en curso.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

27. Consultar análisis semanal de préstamos de la semana

Puerto: GET https://localhost/lending_data/week_counts

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden acceder a las estadísticas diarias de préstamos de la semana en curso.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

28. Consultar análisis anual de préstamos de todos los años

Puerto: GET https://localhost/lending_data/year_counts

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden acceder a estadísticas de préstamos anuales de todos los años registrados.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

29. Deshabilitar cuentas con préstamos vencidos por más de 30 días

Puerto: POST https://localhost/lending_data/ban_user_by_overdue_max_30day

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden deshabilitar cuentas de usuarios cuyos préstamos hayan vencido por más de 30 días. Una solicitud exitosa devolverá una lista con la información de los préstamos vencidos.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

30. Consultar órdenes de préstamo vencidas por más de 30 días

Puerto: GET

https://localhost/lending_data/read_lending_data_by_overdue_max_30day

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden acceder a la información de todas las órdenes de préstamo que han vencido por más de 30 días mediante este puerto.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

31. Enviar recordatorio de vencimiento en tres días por email

Puerto: POST

https://localhost/lending_data/send_email_to_user_by_approaching_date

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden enviar recordatorios por email a los usuarios con préstamos que están próximos a vencer en tres días. Una solicitud exitosa proporcionará una lista con información resumida de estos préstamos.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

32. Enviar recordatorio de préstamo vencido por email

Puerto: POST https://localhost/lending_data/send_email_to_user_by_order_overdue

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden enviar recordatorios por email a los usuarios con préstamos vencidos. Una solicitud exitosa proporcionará una lista con información resumida de estos préstamos.



Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.

33. Crear orden de devolución

Puerto: POST https://localhost/return_data

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT, o CLIENT correspondiente pueden crear nuevas órdenes de devolución mediante este puerto. Tras la creación exitosa, se proporcionará la información de la orden de devolución creada.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El usuario proporcionó una referencia inexistente.

34. Consultar orden de devolución por referencia

Puerto: GET https://localhost/return_data/{reference}

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT, o CLIENT correspondiente pueden obtener información detallada de una orden de devolución asociada a una referencia específica. Si la referencia existe en la base de datos, se proporcionará la información detallada de la orden de devolución.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: La referencia proporcionada por el usuario no existe.

35. Confirmar libro devuelto

Puerto: PUT https://localhost/return_data/{reference}/is_return

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden cambiar el estado de una orden de devolución asociada a una referencia específica a "devuelto". Tras una modificación exitosa, se proporcionará la información actualizada de la orden de devolución.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: La referencia proporcionada por el usuario no existe.

36. Confirmar libro no devuelto

Puerto: PUT https://localhost/return_data/{reference}/no_return

Descripción: Los usuarios con roles de ADMINISTRADOR y ROOT pueden cambiar el estado de una orden de devolución asociada a una referencia específica a "no devuelto". Tras una modificación exitosa, se proporcionará la información actualizada de la orden de devolución.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: La referencia proporcionada por el usuario no existe.

37. Consultar información de libros en la lista de colección por teléfono

Puerto: GET https://localhost/collection_list/{telephone}/book

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT, o CLIENT correspondiente pueden obtener información resumida de los libros en la lista de colecciones asociada a un número de teléfono específico.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El número de teléfono proporcionado por el usuario no existe.

38. Consultar lista de colección por teléfono

Puerto: GET https://localhost/collection_list/{telephone}

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT, o CLIENT correspondiente pueden obtener información de la lista de colecciones asociada a un número de teléfono específico.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El número de teléfono proporcionado por el usuario no existe.

39. Añadir libro a la lista de colección

Puerto: GET https://localhost/collection_list/{telephone}/add_book

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT, o CLIENT correspondiente pueden añadir libros a la lista de colecciones asociada a un número de teléfono específico.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud.
- 404: El número de teléfono proporcionado por el usuario no existe.



40. Eliminar libro de la lista de colección

Puerto: GET https://localhost/collection_list/{telephone}/remove_book

Descripción: Los usuarios con roles de ADMINISTRADOR, ROOT, o el CLIENTE correspondiente pueden eliminar libros de la lista de colecciones asociada a un número de teléfono específico.

Códigos de error:

- 401: Usuario no proporcionó un JWT válido.
- 403: El rol del usuario no tiene los permisos necesarios para la solicitud
- 404: El número de teléfono proporcionado por el usuario no existe.
- 409: BookId suministrado por el usuario que no existe en collection_list

Lógica de Backend

Análisis del proceso del préstamo de un libro

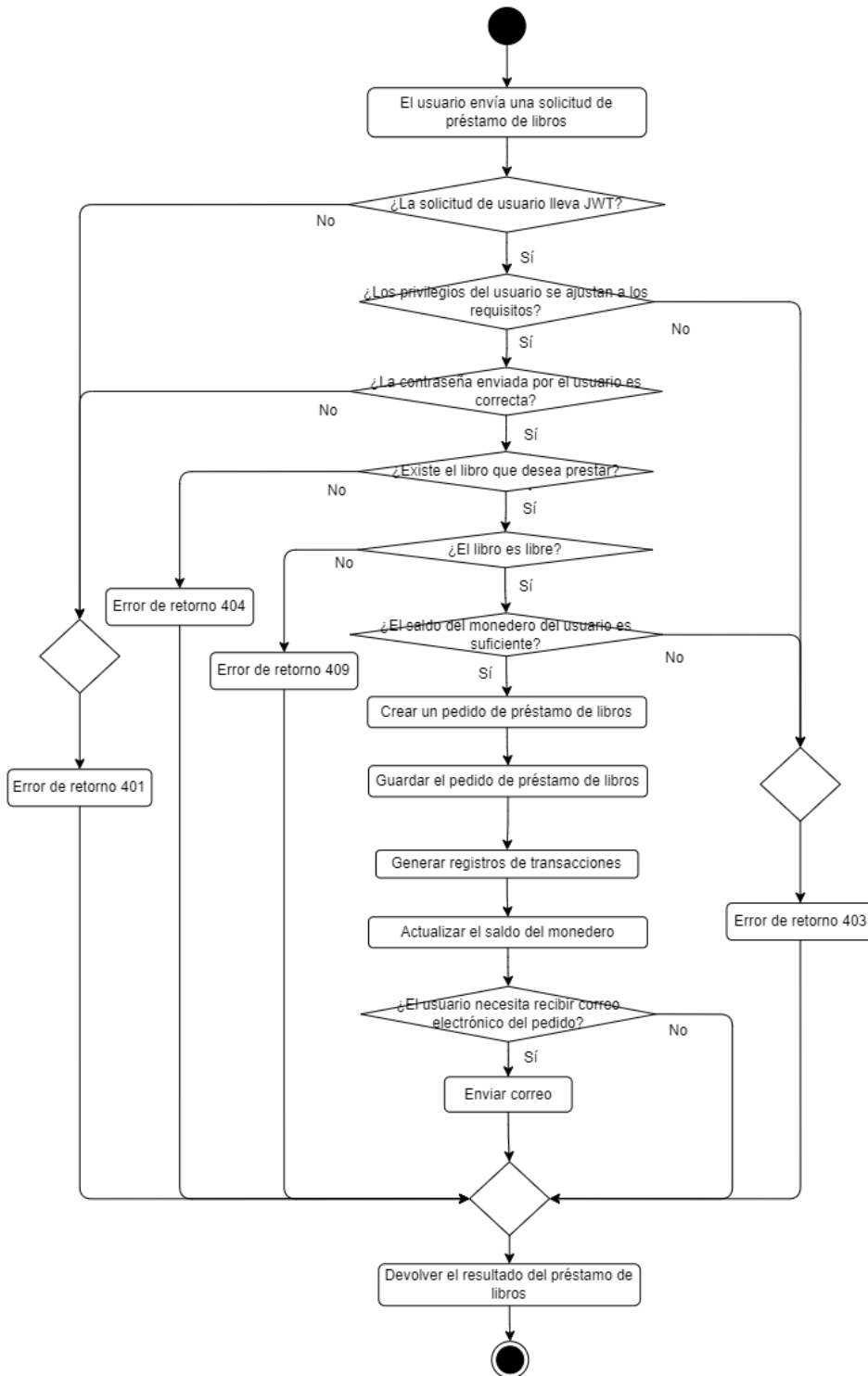


Figura 47. Diagrama de actividad del préstamo de un libro de backend de Sprint 4

1. El usuario inicia una solicitud de préstamo de libros.
2. El sistema verifica primero si el JWT en el encabezado de la solicitud enviado por el usuario es válido.
 - Si no es válido, el sistema devolverá una respuesta de error 401 Unauthorized.



3. El sistema verifica si el usuario tiene permiso para prestar el libro (si es el titular de orden).
 - Si la verificación de permiso no cumple con los requisitos, el sistema devolverá un error 403 Prohibido.
4. El sistema determina si la contraseña de la cuenta enviada por el usuario es correcta.
 - Si la contraseña de la cuenta es incorrecta, el sistema devuelve un error 401 No autorizado.
5. Verifica si el libro existe o está libre.
 - Si el libro no existe, el sistema devuelve un error 404 NotFound.
 - Si el libro está ocupado, el sistema devuelve un error 409 LockedResource
6. Verificar que si el saldo del monedero es suficiente
 - Si el saldo del monedero de la cuenta es insuficiente, el sistema devuelve un error 403 Forbidden.
7. Creación de una orden de préstamo de libros
8. El monedero reduce el depósito correspondiente y genera un registro de transacciones.
9. Se envía un correo electrónico al usuario sobre la creación satisfactoria del pedido de libros de acuerdo con la configuración del usuario.
10. Una vez finalizada la operación, el sistema devuelve la información del pedido de libros al usuario.

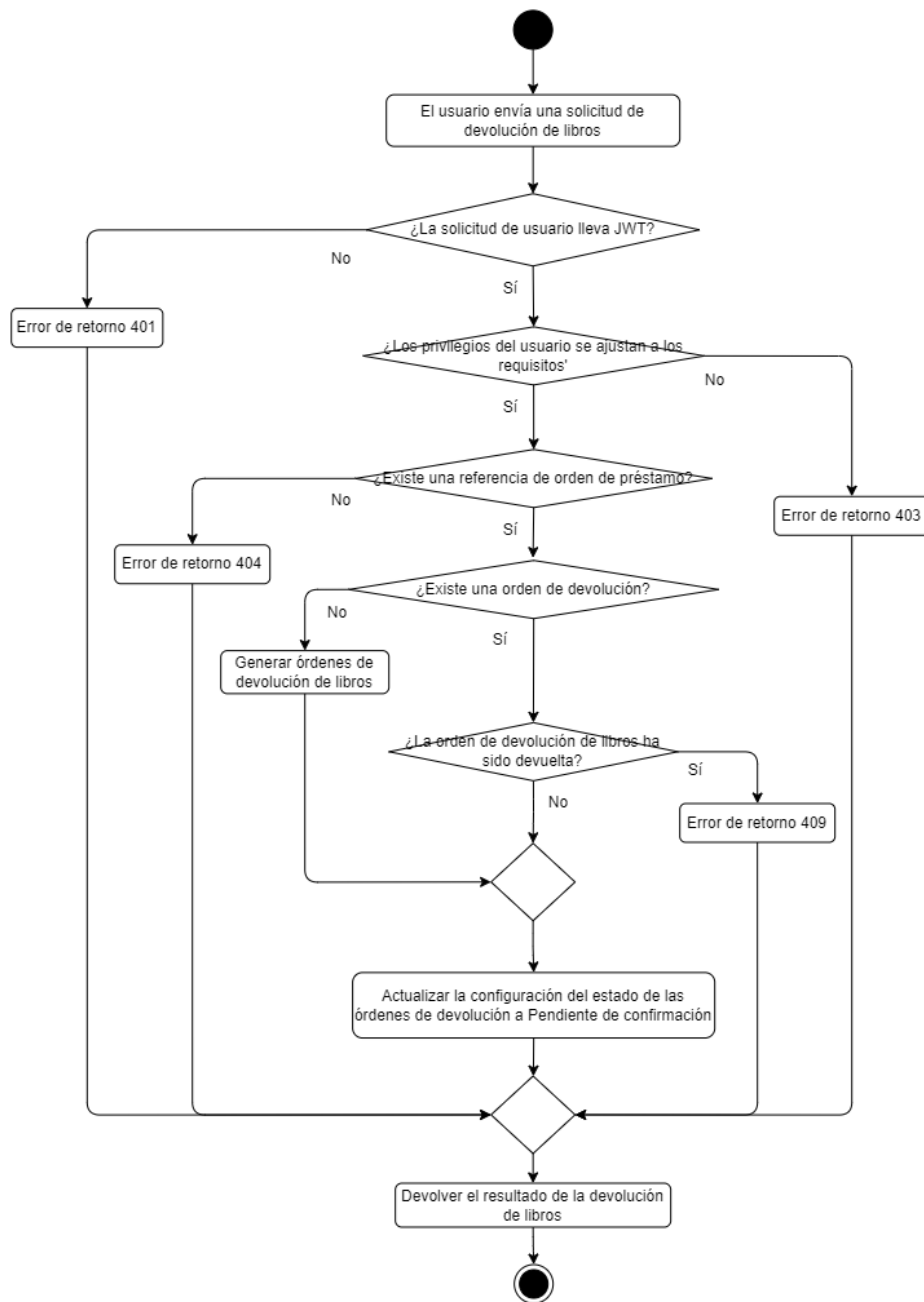
Análisis del proceso de la devolución de libros


Figura 48. Diagrama de actividad de la devolución de libros de backend de Sprint 4

1. El usuario envía una solicitud de devolución de un libro.
2. El sistema verifica primero si el JWT en el encabezado de la solicitud enviado por el usuario es válido.
 - Si no es válido, el sistema devolverá una respuesta de error 401 Unauthorized.
3. El sistema verifica si el usuario tiene permiso para devolver el libro (si es el titular de orden o si el rol del usuario es ROOT o ADMINISTRATOR).
 - Si la verificación de permiso no cumple con los requisitos, el sistema devolverá un error 403 Prohibido.
4. El sistema verifica si la referencia proporcionada por el usuario existe.
 - Si la referencia de la orden de préstamo no existe, el sistema devolverá un error 404 NotFound.



5. El sistema verifica si existe una orden de devolución, en caso negativo, genera una orden de devolución correspondiente, en caso afirmativo, determina si la orden ha sido devuelta.
 - Si la orden de préstamo de libros es devuelta, el sistema devolverá 409 LockedResource error.
6. El sistema establece el estado de la orden de devolución de libros en pendiente de confirmación.
7. Una vez finalizada la operación, el sistema devuelve el resultado de la devolución de libros.

Análisis del proceso para la confirmación del estado de devolución de libros por el administrador

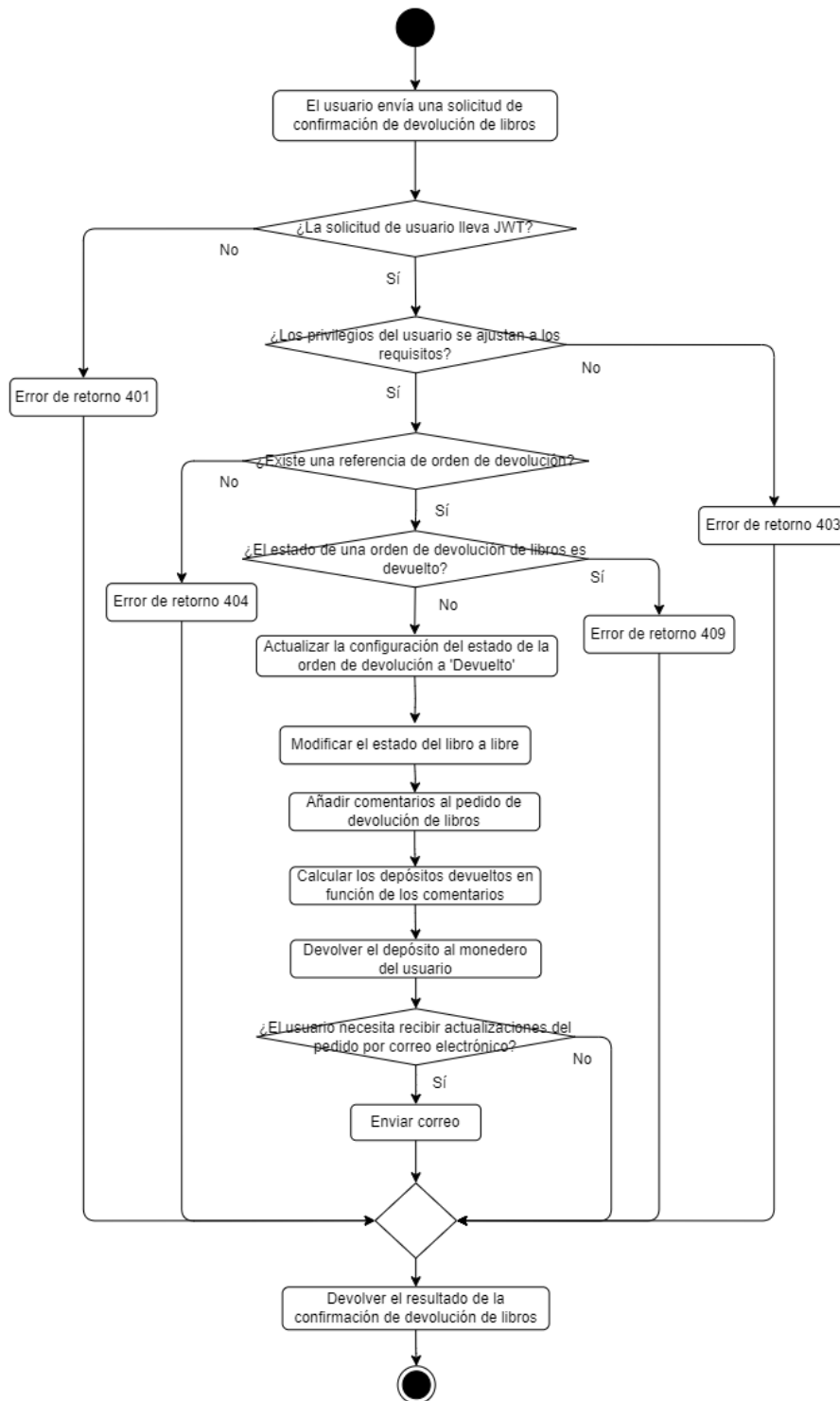


Figura 49. Diagrama de actividad de confirmación del estado de devolución de libros de backend de Sprint 4

1. El usuario envía una solicitud para confirmar el estado de devolución del libro.
2. El sistema verifica primero si el JWT en el encabezado de la solicitud enviado por el usuario es válido.
 - Si no es válido, el sistema devolverá una respuesta de error 401 Unauthorized.



3. El sistema verifica si el usuario tiene permiso para confirmar el estado (si el rol del usuario es ROOT o ADMINISTRATOR).
 - Si la verificación de permiso no cumple con los requisitos, el sistema devolverá un error 403 Prohibido.
4. El sistema verifica si la referencia proporcionada por el usuario para devolver el libro existe.
 - Si la referencia proporcionada para la orden de devolución no existe, el sistema devuelve un error 404 NotFound.
5. El sistema determina si el pedido de devolución es devuelto.
 - Si el estado de la orden de devolución es devuelto, el sistema devolverá un error 409 LockedResource.
6. El sistema cambia el estado del libro que corresponde al pedido de devolución a "Disponible para prestar".
7. El sistema cambia el estado del pedido de devolución a "devuelto".
8. El sistema descuenta el depósito según el estado del libro enviado por el usuario y el estado de atrasado.
9. El sistema devuelve al usuario el depósito calculado.
10. El sistema envía un correo electrónico de alerta de que el libro ha sido devuelto según la configuración del usuario.
11. Una vez finalizada la operación, el sistema devuelve el resultado de la confirmación de devolución de libros.

Análisis del proceso del administrador que deniega el estado de devolución de libros

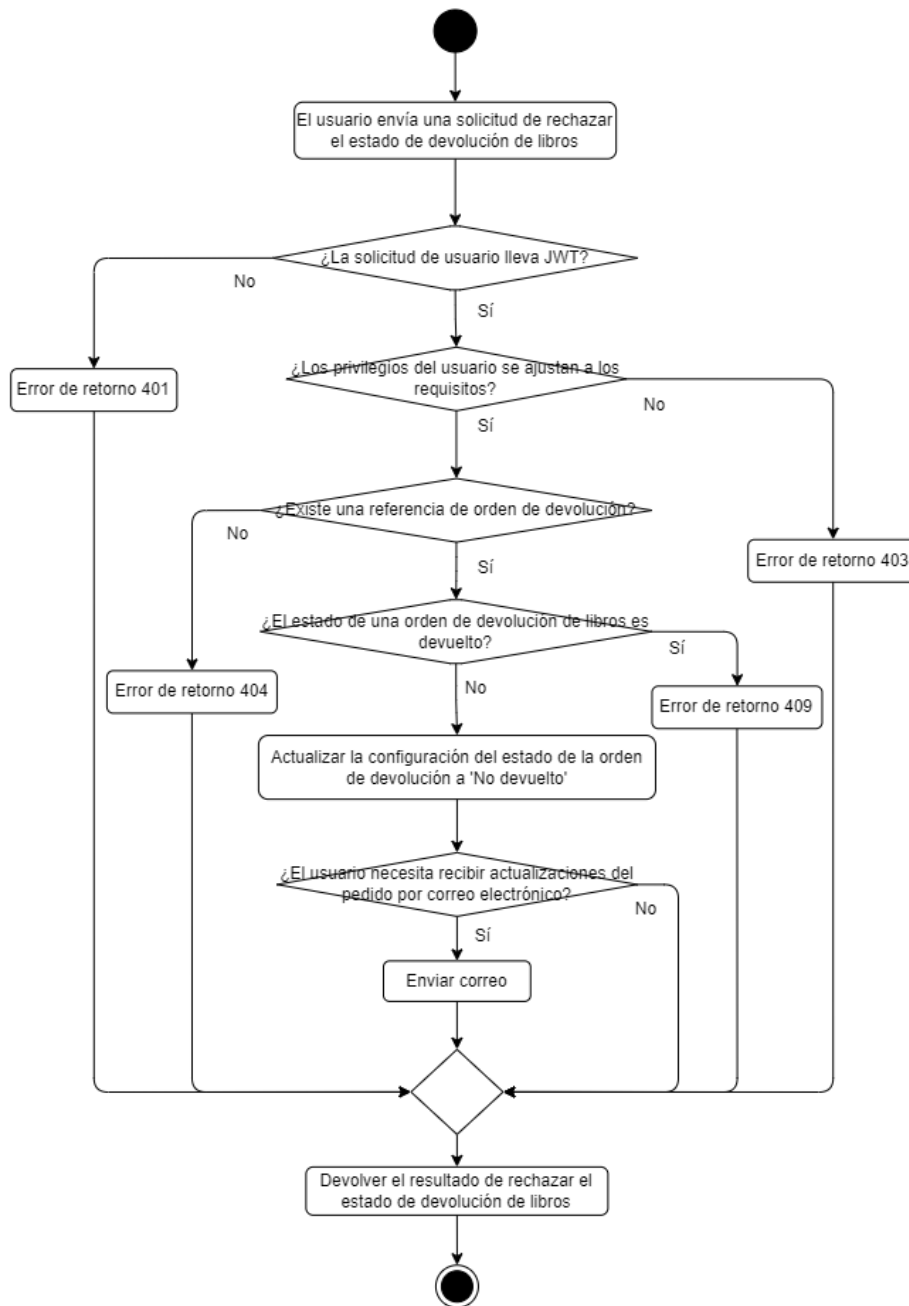


Figura 50. Diagrama de actividad de denegación del estado de devolución de libros de backend de Sprint 4

1. El usuario envía una solicitud para rechazar el estado de devolución del libro.
2. El sistema verifica primero si el JWT en el encabezado de la solicitud enviado por el usuario es válido.
 - Si no es válido, el sistema devolverá una respuesta de error 401 Unauthorized.
3. El sistema verifica si el usuario tiene permiso para rechazar el estado (si el rol del usuario es ROOT o ADMINISTRATOR).
 - Si la verificación de permiso no cumple con los requisitos, el sistema devolverá un error 403 Prohibido.



4. El sistema verifica si la referencia proporcionada por el usuario para devolver el libro existe.
 - Si la referencia proporcionada para la orden de devolución no existe, el sistema devuelve un error 404 NotFound.
5. El sistema determina si el pedido de devolución es devuelto.
 - Si el estado de la orden de devolución es devuelto, el sistema devolverá un error 409 LockedResource.
6. El sistema cambia el estado del pedido de devolución a “No Devuelto”.
7. El sistema envía un correo electrónico de alerta según la configuración del usuario.
8. Una vez finalizada la operación, el sistema devuelve el resultado de rechazar el estado de devolución de libros.

Frontend

En este sprint añadimos 9 componentes al frontend para completar los servicios relacionados.

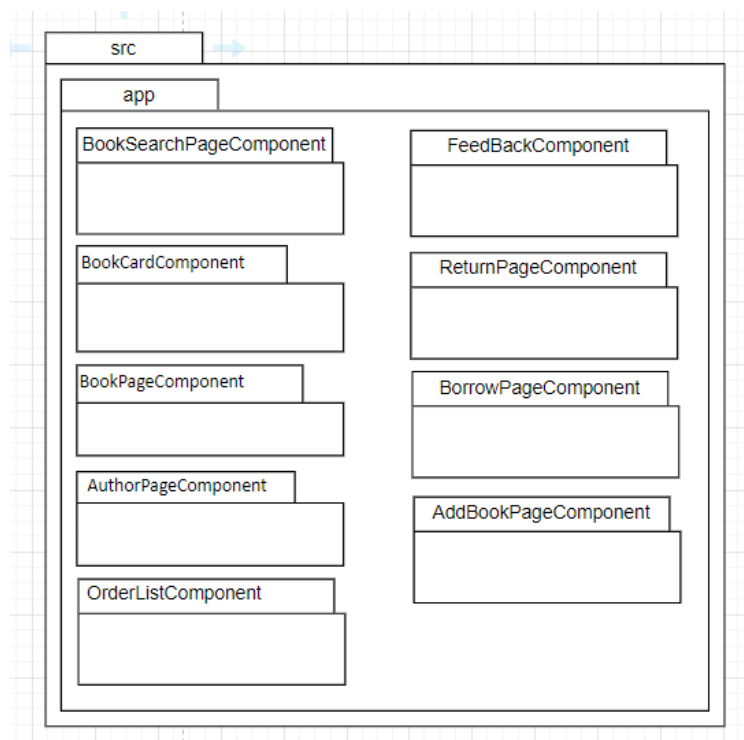


Figura 51. Diagrama de componentes de Sprint 4 frontend

BookSearchPageComponent: Componente de la página de búsqueda de libros.

FeedBackComponent: Componente de la página de comentarios para devoluciones de libros.

BookCardComponent: Componente de la página de mostrar una breve información sobre el libro.

ReturnPageComponent: Componente de la página de mostrar los detalles de pedido préstamo.

BookPageComponent: Componente de la página de mostrar los detalles del libro.

BorrowPageComponent: Componente de la página de mostrar los detalles del pedido devuelto.

AddBookPageComponent: Componente de la página de añadir un libro.

AuthorPageComponent: Componente de la página de mostrar los detalles del autor.

OrderListComponent: Componente de la página de mostrar lista de los pedidos de préstamo y devolución.

Análisis del proceso del frontend para el préstamo de libros:

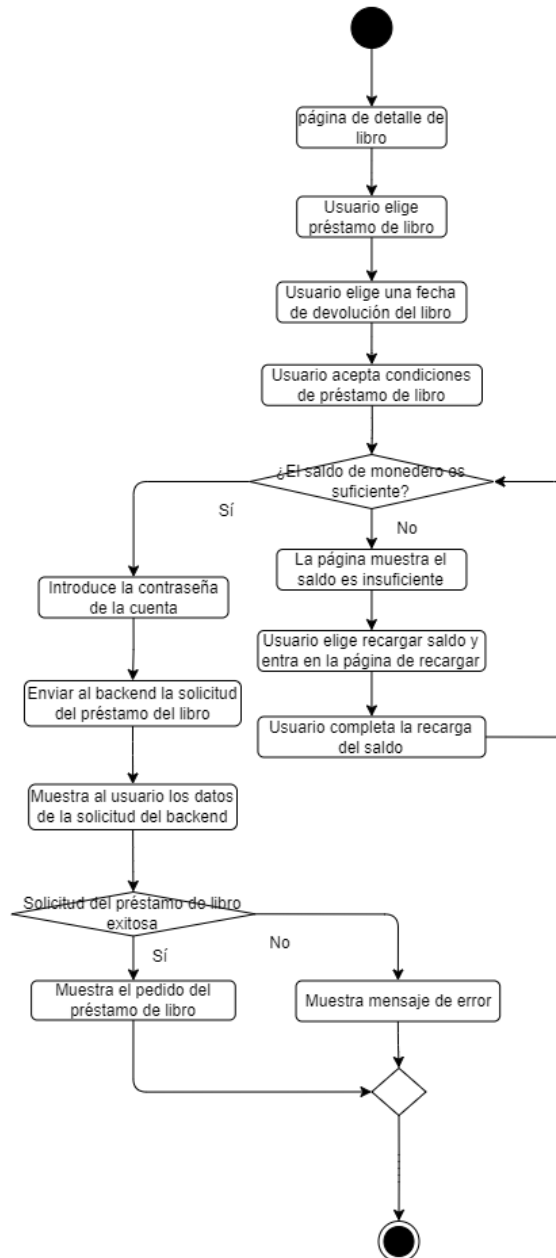


Figura 52. Diagrama de actividad de préstamo de libros de frontend de Sprint 4

En la página del libro, el usuario hace clic para abrir la página de préstamo, iniciando así el proceso de préstamo de libros. En esta interfaz, el usuario debe seleccionar una fecha límite para la devolución del libro. La página muestra el saldo del monedero, si el saldo es insuficiente para pagar el depósito del libro, se pedirá al usuario que recargue el monedero. La página incluye un botón de redirección. Si se necesita recargar, el usuario debe hacer clic en "He leído y acepto los términos y condiciones del



servicio de préstamo y me comprometo a devolver el libro antes de la fecha límite". Al hacer clic en "OK", aparecerá una ventana emergente para la verificación de la cuenta, donde el usuario debe ingresar su contraseña para confirmar que es el titular de la cuenta. Al hacer clic en "Confirmar", se enviará una solicitud al backend y se mostrará la respuesta del servidor. Si el backend confirma que el préstamo fue exitoso, se mostrará la página de la información del pedido de préstamo al usuario.

Análisis del proceso del front-end de devolución de libros.

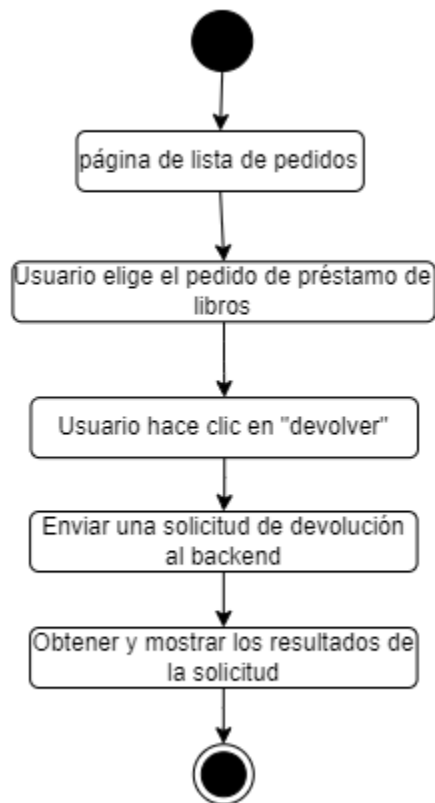


Figura 53. Diagrama de actividad de devolución de libros de frontend de Sprint 4

El usuario selecciona el libro que desea devolver y hace clic en 'Devolver' en la página de la lista de pedidos de libros. En este punto, el front-end envía una solicitud al back-end para devolver el libro y alimenta los datos del back-end.

Análisis del proceso del front-end ROOT y ADMINISTRADOR confirmando la devolución del libro.

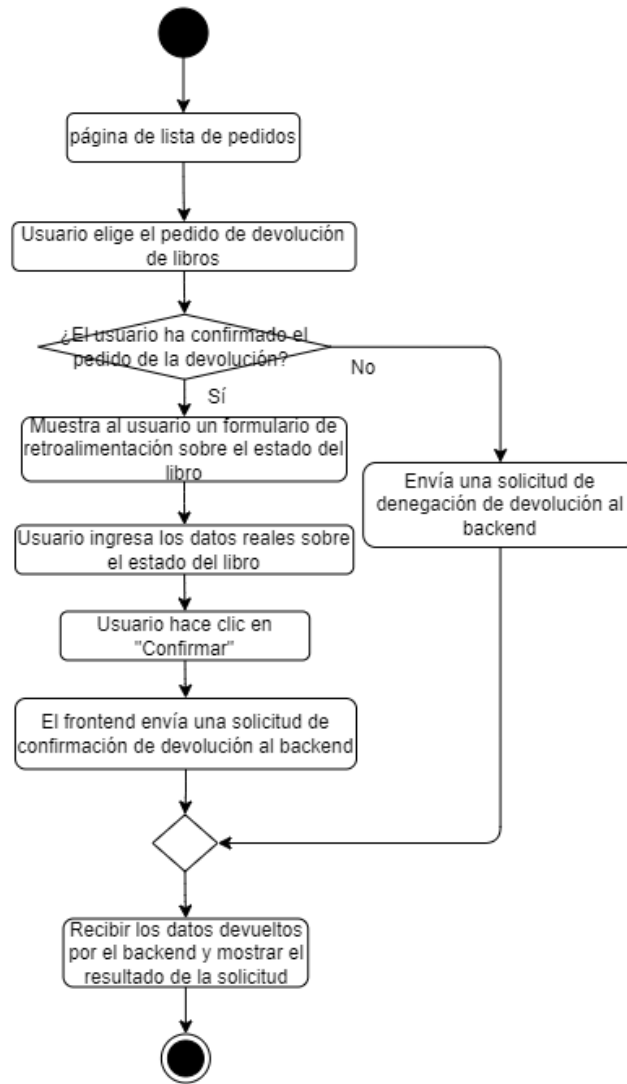


Figura 54. Diagrama de actividad de confirmación de la devolución de libros de frontend de Sprint 4

En la página de la lista de cajas de devolución, el usuario selecciona el libro que necesita confirmar y hace clic en "Confirmar devolución". Aparecerá un formulario de retroalimentación sobre el estado de devolución del libro, donde el usuario deberá ingresar los datos reales sobre el estado del libro. Al hacer clic en "Confirmar", el frontend enviará una solicitud de confirmación de devolución al backend y mostrará la respuesta del servidor. Si el usuario deniega la devolución, el frontend enviará una solicitud de denegación de devolución al backend y espera la respuesta del servidor.

Prueba

Front-end

Prueba de Añadir Etiquetas de Categoría de Libro

Nombre de etiqueta de categoría: ficción (ya existe en el sistema).

TEST-ID	Nombre	Descripción	Resultado
test01	null	test	Añadido fallido (nombre vacío)



test02	ficción	test	Añadido fallido ('ficción' ya existe)
test03	test	test	Añadido exitoso

En esta prueba, nos centramos en probar el nombre de etiqueta de categoría del libro, Los resultados de las pruebas son conforme a lo esperado.

Prueba de Añadir Libro

TEST-ID	Nombre	ISBN	ISSN	Idioma	Resultado
test01	null	test1	null	Spanish	Añadido fallido (el nombre no puede estar vacío)
test02	BOOK1	null	null	Spanish	Añadido fallido (ISBN o ISSN no pueden estar vacío)
test03	BOOK2	test3	null	null	Añadido fallido (el idioma no puede estar vacío)
test04	BOOK3	test4	null	Spanish	Añadido exitoso
test05	BOOK4	null	test5	Spanish	Añadido exitoso
test06	BOOK5	test6	test6	Spanish	Añadido exitoso
test07	BOOK6	test7	test7	test	Añadido fallido (el idioma no es válido)

En esta prueba, nos centramos en probar el nombre, ISBN, ISSN y el idioma del libro. Todos estos parámetros no pueden estar vacíos, y el idioma debe ser un idioma oficial. Los resultados de las pruebas son conforme a lo esperado.

Prueba de Préstamo de Libro

Usuario con teléfono 666666666, su contraseña es 6.

SM: Saldo del monedero del usuario

DL: Depósito del libro

FL: Fecha límite de devolución del libro

CSA: Condiciones de servicios aceptadas

TEST-ID	SM	DL	FL	CSA	Contraseña	Resultado
test01	0	12	2024/8/21	si	6	Préstamo fallido (saldo insuficiente)
test02	20	12	null	si	6	Préstamo fallido (fecha de devolución no especificada)
test03	20	12	2024/8/21	no	6	Préstamo fallido (términos de servicio no marcados)
test04	20	12	2024/8/21	si	null	Préstamo fallido (contraseña incorrecta)
test05	20	12	2024/8/21	si	6	Préstamo exitoso

En esta prueba, nos centramos en probar el saldo del monedero del usuario, fecha límite de devolución del libro, condiciones de servicios aceptadas y la contraseña. De los que el saldo de monedero debe tener un importe superior al depósito del libro, los usuarios deben seleccionar una fecha límite de devolución del libro, deben aceptar las

condiciones de servicios y la contraseña debe estar correcta. Los resultados de las pruebas son conforme a lo esperado.

Prueba de añadir autor

NA: nombre del autor

TEST-ID	NA	Nacionalidad	Descripción	Resultado
test01	null	España	test	Añadido fallido (nombre vacío)
test02	null	null	null	Añadido fallido (nombre vacío)
test03	Pedro González	null	null	Añadido exitoso
test04	Pedro González	null	Más de 5000 caracteres	Añadido fallido (se ha superado el número máximo de caracteres permitidos)
test05	null	null	test	Añadido fallido (nombre vacío)
test06	null	España	null	Añadido fallido (nombre vacío)
test07	Pedro González	España	Es un autor de novela	Añadido exitoso
test08	Pedro González	España	null	Añadido exitoso
Test09	Pedro González	null	test	Añadido exitoso

En esta prueba, nos centramos en probar añadir un autor en el sistema. De los que probamos el nombre, la nacionalidad y su descripción. El nombre no puede estar vacío y que el número de caracteres que introduce por la descripción no puede superar a 5000. Los resultados de las pruebas son conforme a lo esperado.

Prueba de modificar los datos del libro

NL: nombre del libro

EL: Editor del libro

TEST-ID	NL	ISBN	ISSN	EL	Depósito	idioma	Resultado
test01	test	1213215	null	null	-1	Inglés	Añadido fallido (El depósito debe ser mayor o igual que 0)
test02	null	1213215	null	null	5	Inglés	Añadido fallido (nombre del libro vacío)
test03	test	null	12315	test	10	test	Añadido fallido (el idioma no es válido)
test04	test	null	null	test	10	Inglés	Añadido fallido (el ISBN o ISSN vacío)
test05	null	null	12354	null	5	Inglés	Añadido fallido (nombre del libro vacío)
test06	null	1213215	null	test	5	Inglés	Añadido fallido (nombre del libro vacío)
test07	test	null	12315	test	10	Inglés	Añadido fallido (nombre del libro vacío)
test08	test	12312	null	test	5	Inglés	Añadido exitoso
test09	test	null	12312	test	5	Inglés	Añadido exitoso
Test10	test	15462	12312	test	5	Inglés	Añadido exitoso

En esta prueba, nos centramos en probar modificar los datos del libro en el sistema. De los que probamos el nombre, ISBN, ISSN, editor, depósito y el idioma. El nombre no puede estar vacío. Y debe rellenar entre ISBN o ISSN. Por último, el depósito del libro debe ser mayor o igual que 0. Los resultados de las pruebas son conforme a lo esperado.



Back-end

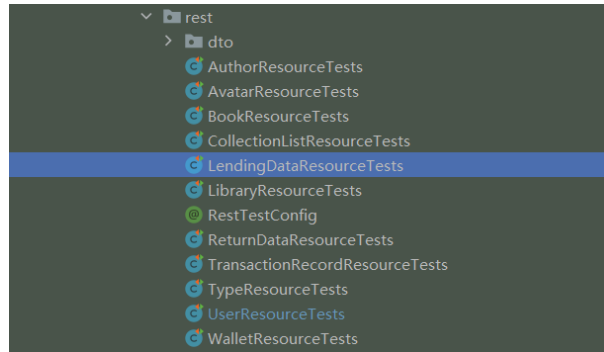


Figura 55. Pruebas (1) de Sprint 4

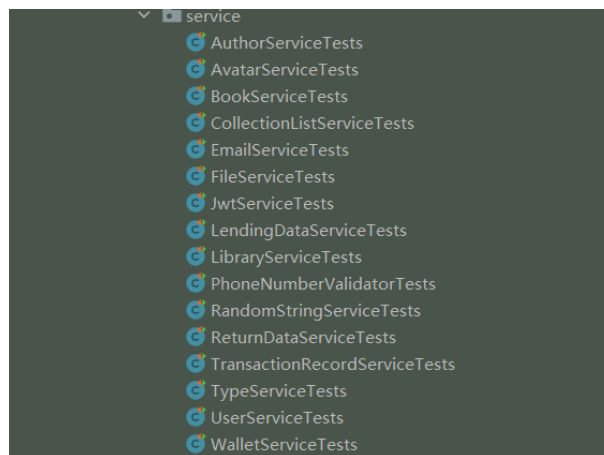


Figura 56. Pruebas (2) de Sprint 4

Resultados de las pruebas

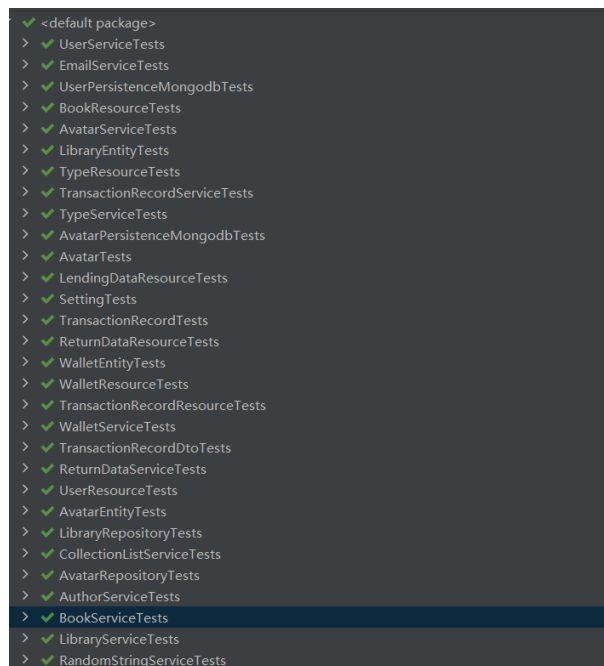


Figura 57. Resultado de pruebas (1) de Sprint 4

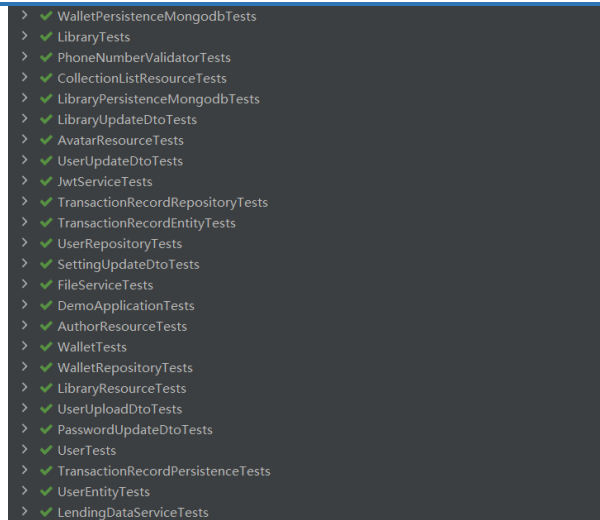


Figura 58. Resultado de pruebas (2) de Sprint 4

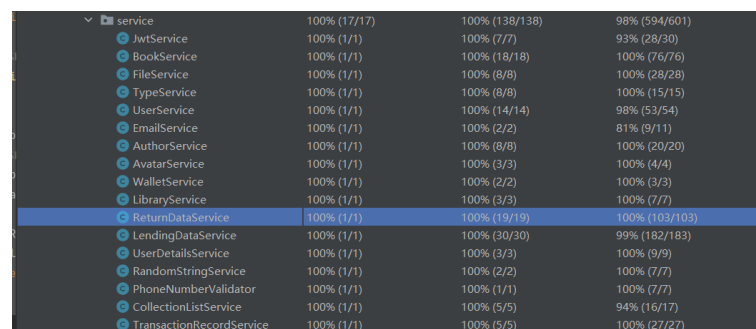


Figura 59. Resultado de pruebas (3) de Sprint 4

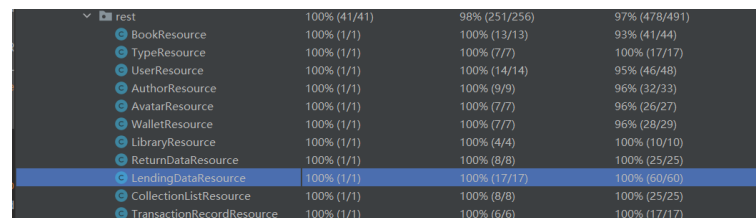


Figura 60. Resultado de pruebas (4) de Sprint 4

La prueba final de esta iteración terminó con una cobertura del 100% para todas las clases de servicio y del 98% para las líneas de código.

La clase de puerto (Resource) tiene una cobertura del 100% y del 97% para las líneas de código.

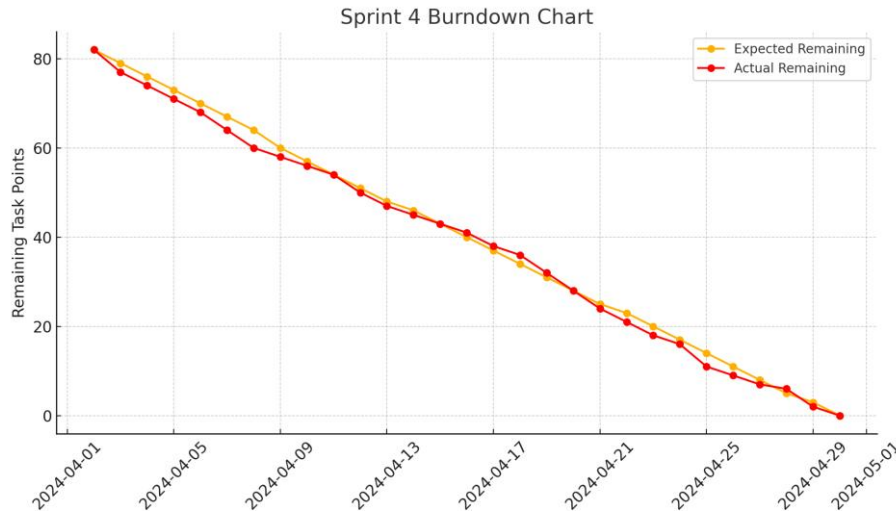


Figura 61. Burn down chart de Sprint 4

El tiempo del último sprint fue bastante ajustado, y la gran cantidad de tareas nos obligó a acelerar el ritmo. Gracias a la excelente capacidad de cada miembro de nuestro equipo, logramos completar todas las tareas del sprint antes de la fecha límite.

3. Retrospective

Lo que fue bien:

1. Ajuste en el orden de escritura de las clases de prueba:

- Se ajustó el orden en que se escriben las clases de prueba para reducir las pruebas duplicadas en el código, lo que resultó en una mayor eficiencia y menos desperdicio de recursos.

Lo que se puede mejorar:

1. Ineficiencias en las pruebas de sprints anteriores:

- A pesar de los ajustes realizados, todavía se identificaron algunos casos en los que el código se sometía a pruebas repetidas de sprints anteriores, lo que podría seguir provocando ineficiencias.

Acciones a tomar:

1. Continua optimización del proceso de pruebas:

- Seguir ajustando y refinando el proceso de pruebas para eliminar por completo las redundancias y asegurar que cada parte del código sea probada de manera eficiente. Implementar revisiones periódicas del proceso de pruebas para identificar y corregir cualquier ineficiencia residual.

PRESENTACIÓN DEL SISTEMA RESULTANTE

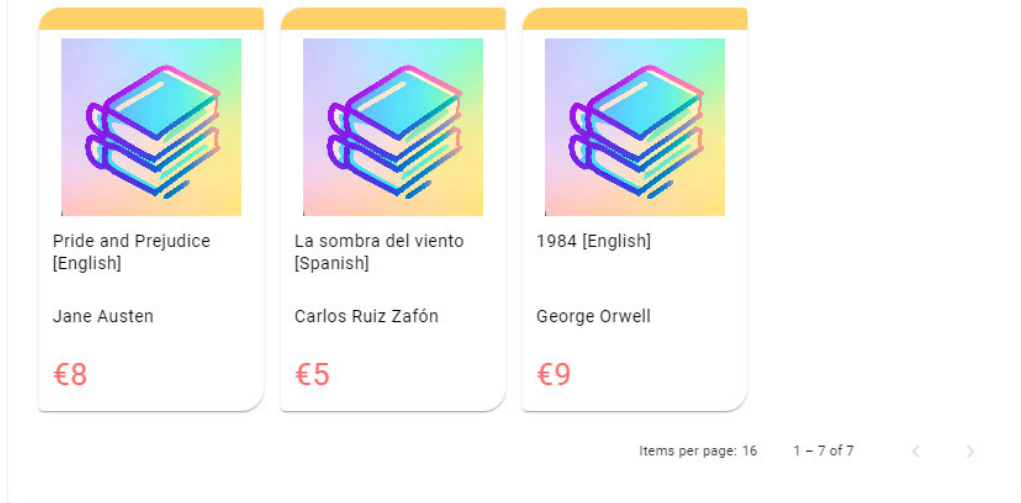
En este capítulo se presenta el resultado final del sistema web desarrollado, proporcionando una visión general de la aplicación que servirá como un manual de uso y presentación de las interfaces de la web desarrollada. El sistema web de biblioteca desarrollada se distingue por cinco tipos principales de usuarios: visitante casuales, clientes registrados, administrador, administrador superior y usuario prohibido.

1. PÁGINA DE INICIO

The screenshot shows the home page of a library website. At the top, there is a teal navigation bar with the text "LIBRARY" on the left, and "Home", "Sign up", and "Login" on the right. Below the navigation bar, there are three main sections: a date and book cover for "The Great Gatsby" (13/Jun), a weather forecast for Madrid (Clear, 17° MIN:15° MAX:19°, 10:27:32 Madrid), and "Login" and "Sign up" buttons. Below these sections is a search bar with the text "Search" and a search icon. The search bar has filters for "Language" (All), "Type" (All), and "Search name" (name). Below the search bar are two checkboxes: "Only show ENABLE" and "Show Borrow Count". At the bottom, there are four book cards, each with a colorful book icon, title, author, and price:

Book Title	Author	Price
Cien años de soledad [Spanish]	Gabriel García Márquez	€10
The Great Gatsby [English]	F. Scott Fitzgerald	€12
Ficciones [Spanish]	Jorge Luis Borges	€10
Don Quijote de la Mancha [Spanish]	Miguel de Cervantes Saavedra	€10

Figura 62. Pantalla de inicio (1)



Phone: 6669996669

Email: bibliotecaMadrid@email.com

Address: X.º XX XXXXXXX, XX, 28001, XXXXXX, Madrid

Postal Code: 28001

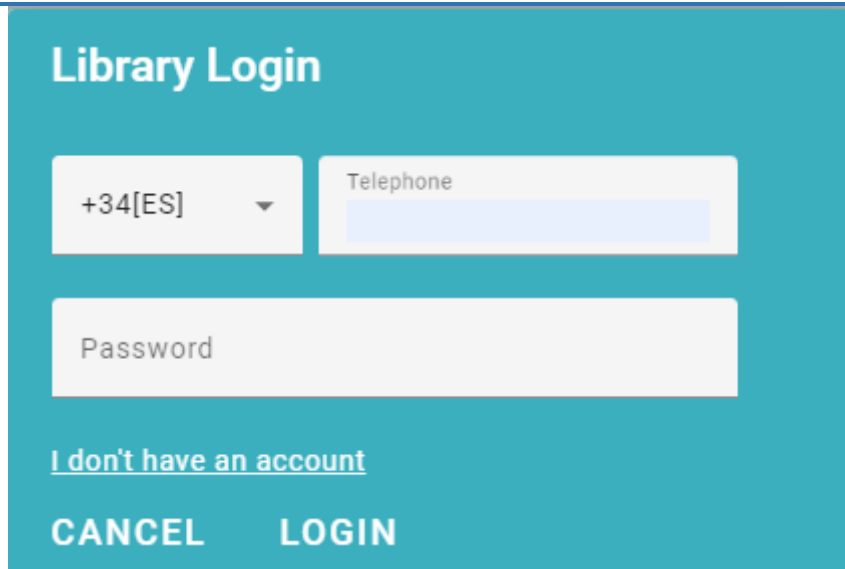
Business Hours: Monday to Friday 10:00 to 22:00 Saturday and Sunday 10:00 to 16:00

--Contact Us--



Figura 63. Pantalla de inicio (2)

Al entrar al sistema web de la biblioteca, los visitantes reciben una visión general de la página de inicio de la web de biblioteca. En el que se presenta las funciones básicas de la biblioteca, como hacer un login, permite un usuario no registrado registrarse, buscar libros que está interesado y mirar si está activo para prestar, incluso para conocer cuántas veces ha sido prestado este libro, conocer el tiempo y la temperatura que hace por Madrid, ver cuál es el libro recomendado diario y finalmente, se presenta la información básica de la biblioteca como forma de contacto, dirección de ubicación y los horarios de apertura.



Library Login

+34[ES] Telephone

Password

[I don't have an account](#)

CANCEL LOGIN

Figura 64. Pantalla de login

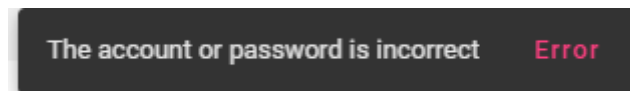


Figura 65. Pantalla de error de login

La opción de 'login' proporciona a los usuarios que poseen cuenta la capacidad de iniciar sesión en la biblioteca, introduciendo su número de teléfono y contraseña, le permite seleccionar el prefijo de número de teléfono si se ha registrado con un número de teléfono de otro país. Si alguno de estos datos no coincide con los que tienen registrado en la base de datos, se muestra un mensaje de error al usuario para que se corrija.



Sign up

Figura 66. Pantalla de sign up (1)

Sign up

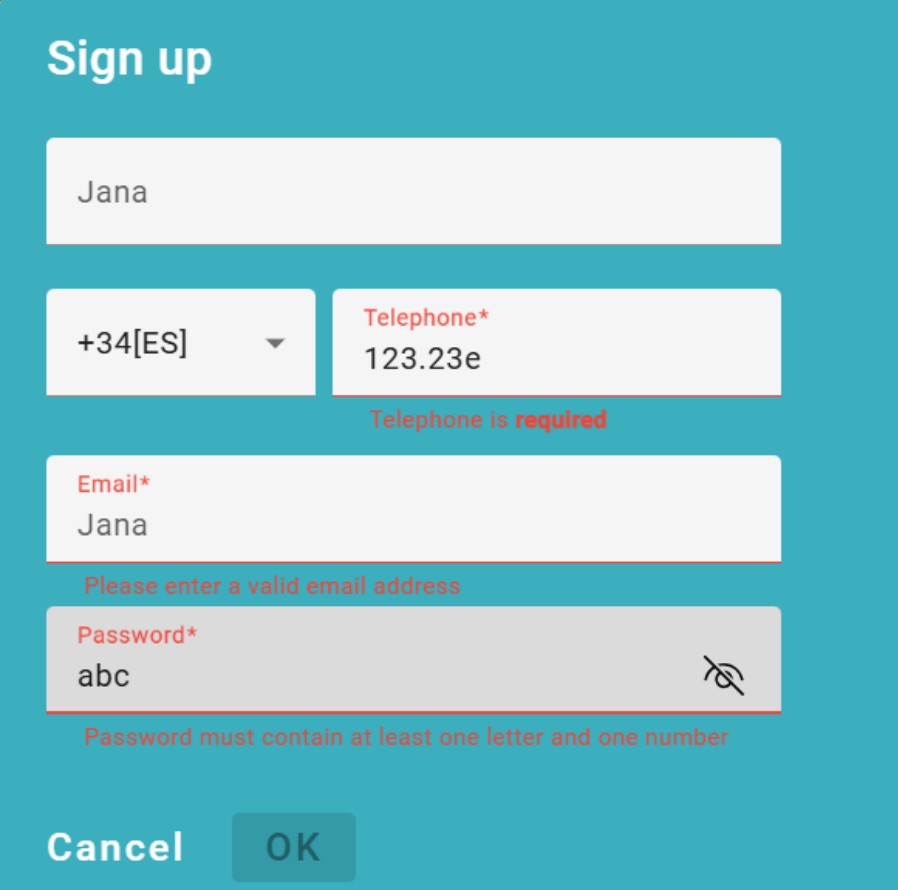
Name is required

Telephone is required

Email is required

Password is required

Figura 67. Pantalla de sign up (2)



The image shows a 'Sign up' form with the following fields and errors:

- Name:** Jana
- Country:** +34[ES]
- Telephone:** 123.23e. Error: Telephone is **required**.
- Email:** Jana. Error: Please enter a valid email address.
- Password:** abc. Error: Password must contain at least one letter and one number.

Buttons: Cancel, OK

Figura 68. Pantalla de sign up (3)

La opción de 'Sign up' permite a los usuarios crear una cuenta de cliente en la biblioteca llenando un formulario con sus datos personales como nombre, número de teléfono, email y contraseña. En caso de que no se completen todos los campos o si el formato del correo electrónico, teléfono o contraseña no se ajuste a los requisitos especificados, se mostrarán mensajes de error correspondientes a cada campo. Es importante mencionar que el sistema verifica si el número de teléfono a registrar ya existe en la base de datos.

2. VISTA DE CLIENTE REGISTRADO



13/Jun
Ficciones [Language:Spanish]
Jorge Luis Borges

Madrid
MIN:15°
MAX:19°
25:33 Madrid

Hello
Jana!!
Balance
18 €

Search

Language: All | Type: All | Search name: name

Only show ENABLE Show Borrow Count

 Cien años de soledad [Spanish] Gabriel García Márquez €10	 The Great Gatsby [English] F. Scott Fitzgerald €12	 Ficciones [Spanish] Jorge Luis Borges €10	 Don Quijote de la Mancha [Spanish] Miguel de Cervantes Saavedra €10
---	--	---	---

Figura 69. Pantalla de inicio (1) de cliente registrado

 Pride and Prejudice [English] Jane Austen €8	 La sombra del viento [Spanish] Carlos Ruiz Zafón €5	 1984 [English] George Orwell €9
--	---	---

Items per page: 16 1 - 7 of 7

Phone: 6669996669
Email: bibliotecaMadrid@email.com
Address: X.º XX XXXXXXX, XX, 28001, XXXXXX, Madrid
Postal Code: 28001
Business Hours: Monday to Friday 10:00 to 22:00 Saturday and Sunday 10:00 to 16:00

--Contact Us--

Figura 70. Pantalla de inicio (2) de cliente registrado

13 / Jun

Cien años de soledad
[Language: Spanish]
Gabriel García Márquez

Lending Data		
Reference	Book	LimitTime
JwDVYQRQWQVZ	The Great Gatsby [ID:L0osyn5iMN5o]	2024-08-15 23:59:59

Hello
Jana!!
Balance
18 €

Figura 71. Pantalla de encabezado de cliente registrado

Si el cliente registrado inicia su cuenta en la web de biblioteca, se le muestra la página inicial similar que un visitante, pero la diferencia está que ahora permite que el cliente registrado haga peticiones por la web. Como se puede ver por la captura de arriba, el cliente registrado tiene un monedero electrónico donde puede hacer recargas y retiros, y en la página de inicio se le muestra el saldo, además se le muestra un modal más con su información de préstamo de libros.

Language
All

Type
All

Search name

Only show ENABLE Show Borrow Count

BookId: FKfD5WNfSafv

Barcode: 222222222

Book Name: Cien años de soledad

ISBN: 978-84-376-0494-7

Language: Spanish

Publisher: Editorial Sudamericana

Deposit: €10

Author: Gabriel García Márquez[Colombia]

Book Type:

♥

Close

Pride and Prejudice
[English]

Jane Austen

€8

La sombra del viento
[Spanish]

Carlos Ruiz Zafón

€5

1984 [English]

George Orwell

€9

Figura 72. Pantalla de libro de cliente registrado



Y si el cliente registrado hace clic encima de un libro que le interesa, le permite hacer préstamo de este libro y le muestra la información detallada del libro. Le permite guardar el libro a la lista de favoritos también para facilitar la búsqueda desde luego.

Reference	Book	BorrowTime	LimitTime	User	Return
xK3zbGZ4e0rg	Don Quijote de la Mancha[ID:WeDqA0qxIKXx]	2024-06-13 00:05:19	2024-06-26 23:59:59	+34654321000	
JwDVYQRQWQVZ	The Great Gatsby[ID:L0osyn5iMN5o]	2024-06-13 10:25:02	2024-08-15 23:59:59	+34654321000	Return

Figura 73. Pantalla de lista de préstamo de libro

Borrowing Information

Reference: JwDVYQRQWQVZ
User name: Jana
Telephone: +34654321000
Book ID: L0osyn5iMN5o
Barcode: 5555555
Book Name: The Great Gatsby
Book Deposit: €12
Borrow Time: 2024-06-13 10:25:02
Limit Time: 2024-08-15 23:59:59

Figura 74. Pantalla de información de préstamo de libro

LIBRARY Home Orders Records Jana

Book Borrow Order List **Book Return Order List**

Returned list Refresh

Search

Reference	Book	ReturnStatus	Status of return	User	View
xK3zbGZ4e0rg	Don Quijote de la Mancha [ID:WeDqA0qxIKXx]	IS_RETURN	Normal	+34654321000	
JwDVYQRQWQVZ	The Great Gatsby [ID:L0osyn5iMN5o]	WAITING_FOR_VERIFICATION	Normal	+34654321000	

Items per page: 5 1 - 2 of 2

Figura 75. Pantalla de lista de devolución de libro

LIBRARY Home Orders Records Jana

Book Borrow Order List **Book Return Order List**

Returned list Refresh

Return Information

Reference: xK3zbGZ4e0rg
User name: Jana
Telephone: +34654321000
Book ID: WeDqA0qxIKXx
Book Name: Don Quijote de la Mancha
Book Deposit: €10
Borrow Time: 2024-06-13 00:05:19
Limit Time: 2024-06-26 23:59:59
Return Time: 2024-06-13 00:05:43
Return Status: IS_RETURN

Degree of damage to books returned

Book Degree: PERFECT
Return Addendum: The books are very well protected
Return Deposit: €10
Details Deductions: Extended by 0 day, deduction of 0. Degree of book integrity [PERFECT], deduction of 0, total deduction of 0. Hence the return of the deposit 10.00€
Addendum Time: 2024-06-13 00:14:30

Phone: 666
 Email: bibli
 Address: X
 Postal Code: 28001
 Business Hours: Monday to Friday 10:00 to 22:00 Saturday and Sunday 10:00 to 16:00

Contact Us

Figura 76. Pantalla de información de devolución de libro

La opción de 'Orders records' permite al cliente registrado visualizar los pedidos que han hecho por la biblioteca, donde se clasifica en lista de préstamo de libros y lista de devolución de libros con sus determinadas funcionalidades como devolver el libro, visualizar la información de la devolución del libro, visualizar el estado de la devolución



del libro y el comentario proporcionado por el administrador para evaluar el grado de daño al libro devuelto y depósito correspondiente.

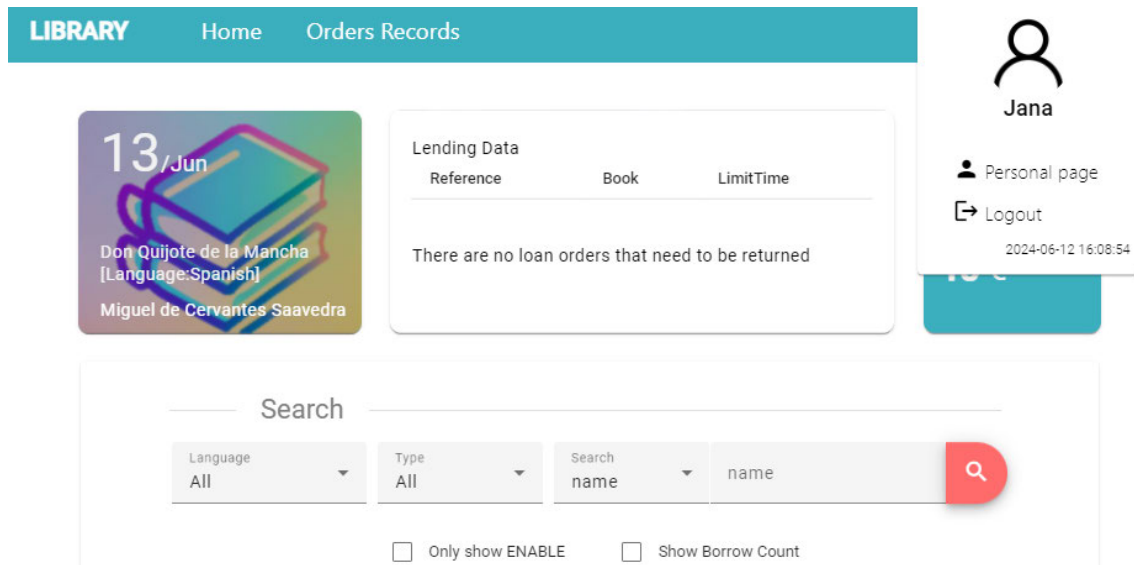


Figura 77. Pantalla de perfil de usuario

Si el cliente registrado clicla encima de su nombre, se le salen las opciones de su página personal y logout, así como su fecha que se dio de alta en el sistema.

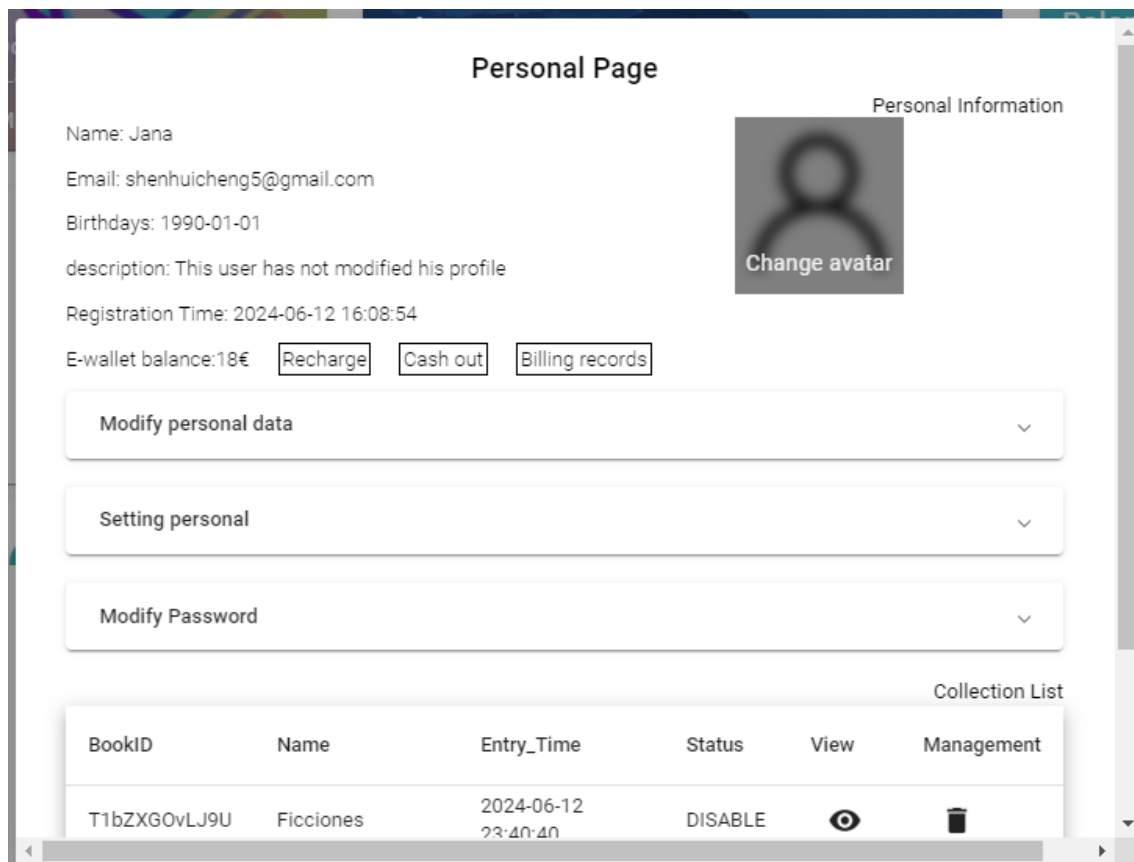


Figura 78. Pantalla de página personal de usuario

3 / Jun

Landing Data

Recharge

Recharge Account [TELEPHONE: +34654321000]

Balance: 18 €

Name: Jan
Email: she
Birthdays:
description:
Registratio
E-wallet ba

Sum of money
0

Please select a payment method

Card number

Expiration Date CVV

First Name Last Name

City Postal Code

Billing Address

Recharge Close

Modification

Information

Hel
Janal
Bala

Gabriel Garc

el de Cervante
adra

Figura 79. Pantalla de recarga de saldo



Cash out

Account [TELEPHONE: +34654321000]

Balance: 18 €

Sum of money
0

Please select card type

Card number

First Name Last Name

City Postal Code

Billing Address

Confirm Close

Figura 80. Pantalla de retiro de saldo

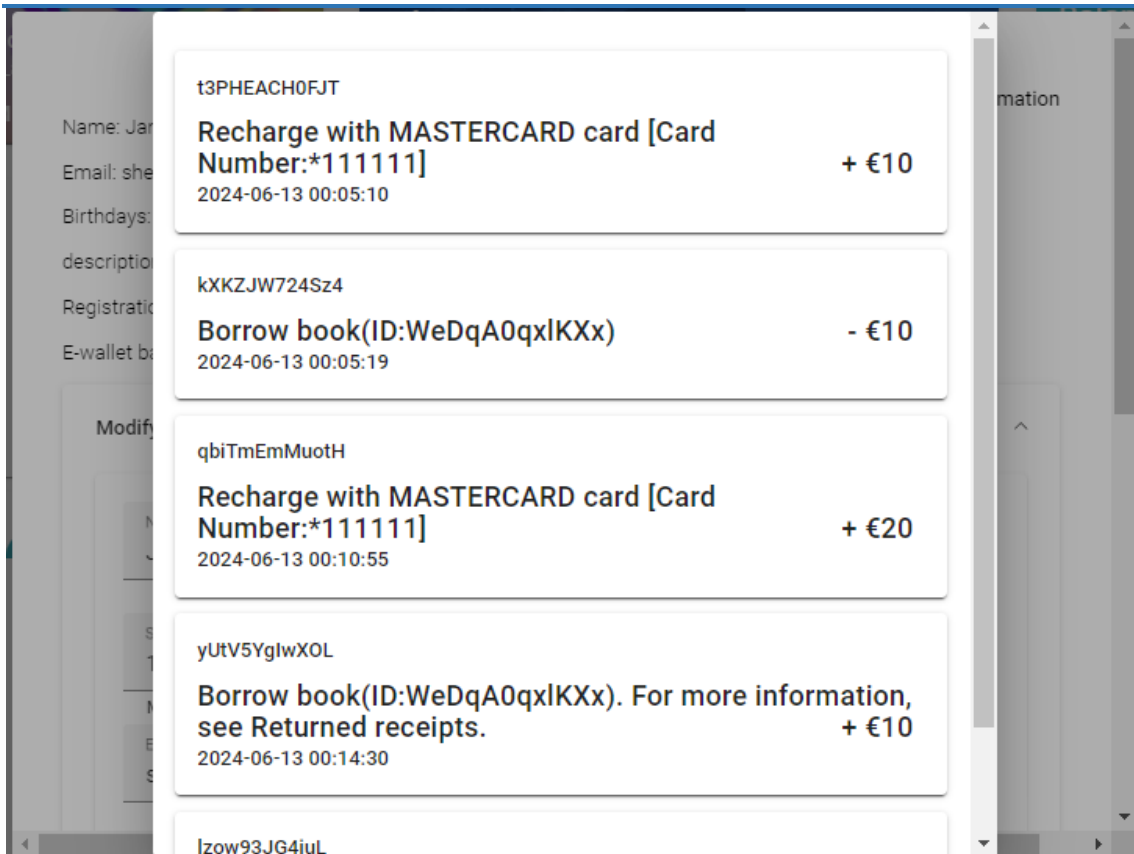


Figura 81. Pantalla de registro de transacciones

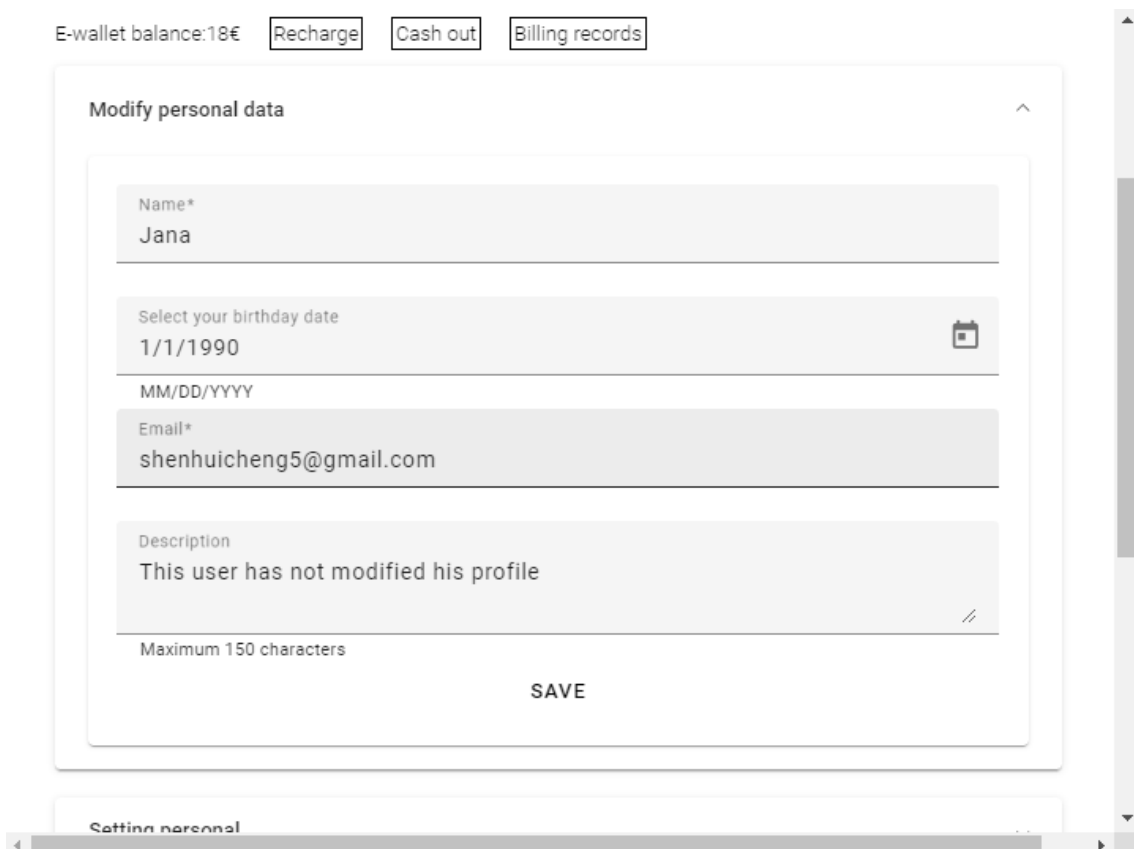


Figura 82. Pantalla de perfil de usuario (1)



Maximum 150 characters

SAVE

Setting personal

- I would like to receive email alerts of successful transactions
- I would like to receive email alerts for order generation
- I would like to receive email alerts that my order is approaching its deadline
- I would like to receive email alerts with more detailed information on book return orders

OK

Modify Password

New Password

.....

Figura 83. Pantalla de perfil de usuario (2)

Modify Password

New Password

.....

Confirm Password

Collection List

BookID	Name	Entry_Time	Status	View	Management
T1bZXGOvLJ9U	Ficciones	2024-06-12 23:40:40	DISABLE		
xcdi6dNnwzVm	1984	2024-06-12 22:06:07	DISABLE		
vF46or9VbWgt	La sombra del viento	2024-06-12 21:55:58	DISABLE		

Figura 84. Pantalla de perfil de usuario (3)

En su página personal, el cliente registrado le permite hacer operaciones como cambiar avatar del perfil, hacer operaciones correspondientes al monedero electrónico (hacer recargas de saldo, retiros de saldo y visualizar las transacciones del monedero), cambiar datos personales, cambiar las configuraciones personales, modificar la contraseña y visualizar la lista de libros favoritos, así como quitar el libro favorito.

3. VISTA DE ADMINISTRADOR

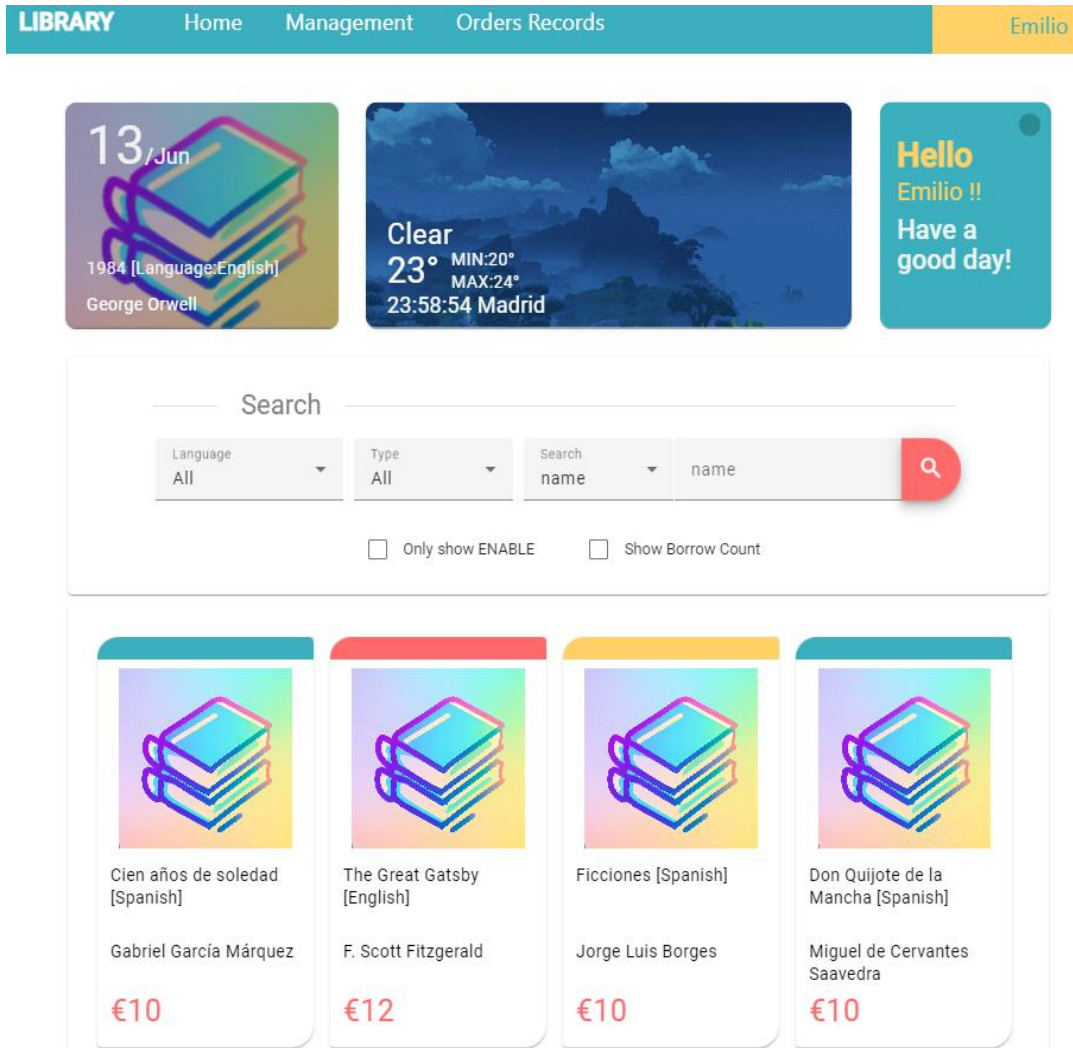


Figura 85. Pantalla de página de inicio (1) de administrador

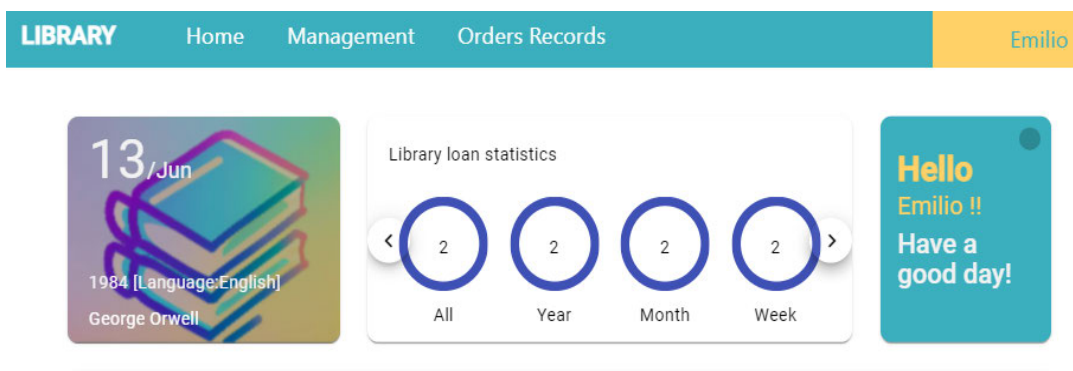


Figura 86. Pantalla de página de inicio (2) de administrador



Si el administrador inicia su cuenta en el sistema web de biblioteca, se le muestra la página inicial similar que un cliente registrado, pero ahora en la página inicial en vez de mostrar los datos de libros de préstamo, se le muestra al administrador los datos totales de libros prestados al año, mes y semana.

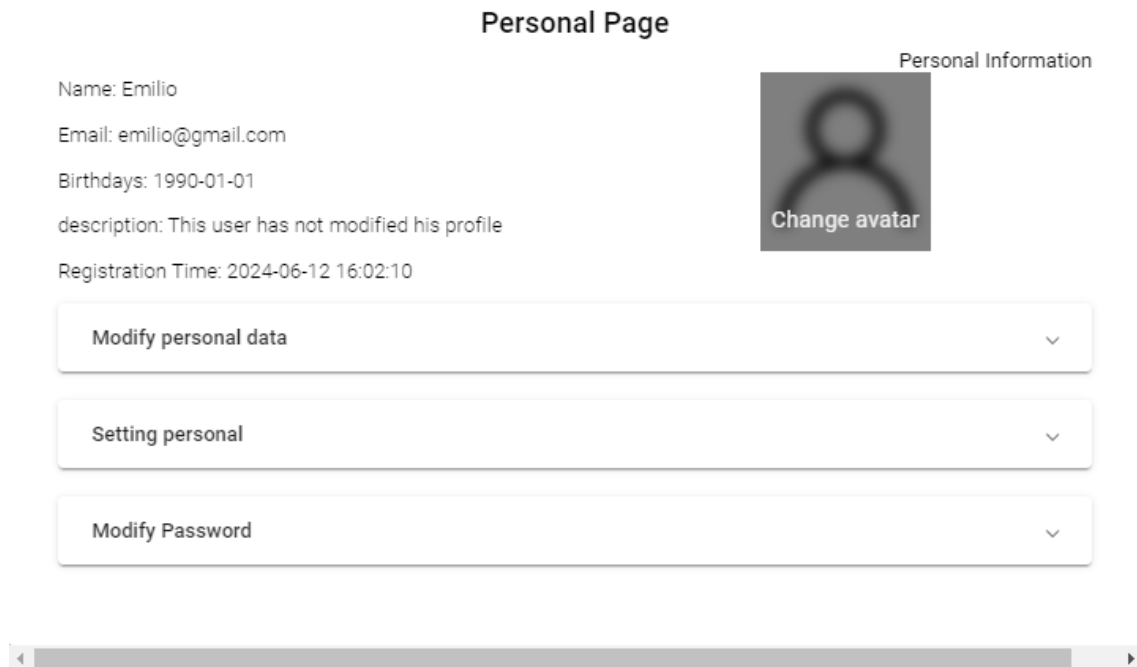


Figura 87. Pantalla de página personal de administrador

Igual que un cliente registrado, le permite hacer operaciones como cambiar el avatar del perfil, cambiar datos personales, cambiar las configuraciones personales y modificar la contraseña.

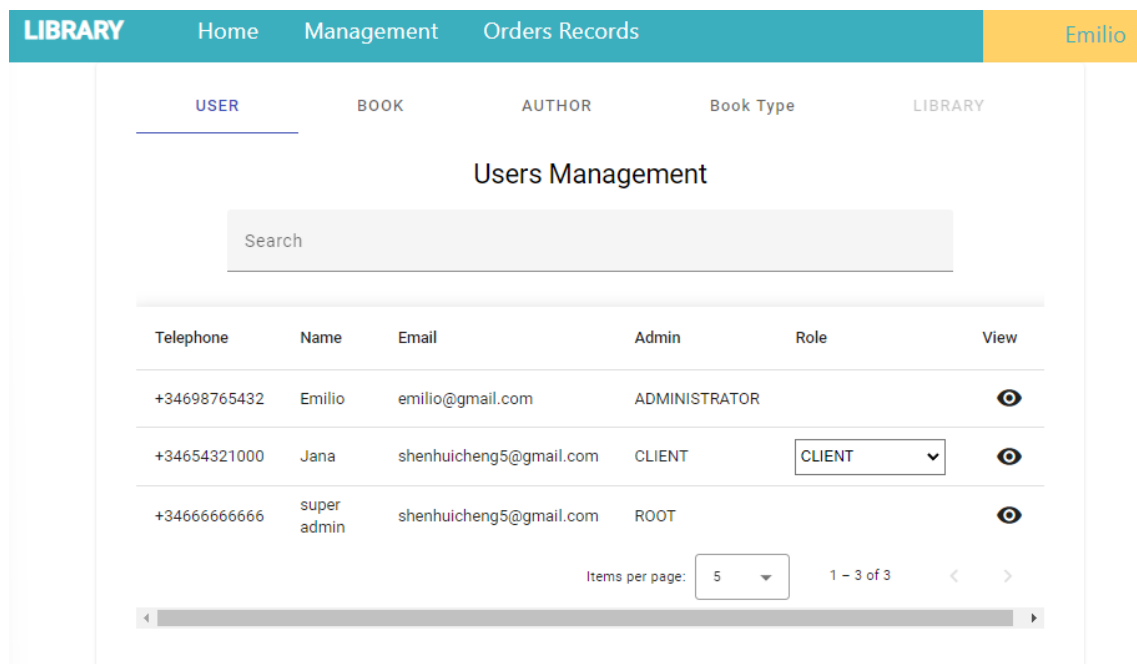


Figura 88. Pantalla de gestión de usuarios

Personal Page

Name: Jana

Email: shenhuicheng5@gmail.com

Birthdays: 1990-01-01

description: This user has not modified his profile

Registration Time: 2024-06-12 16:08:54

Personal Information

[Collection List](#)

BookID	Name	Entry_Time	Status	View	Management
T1bZXGOvLJ9U	Ficciones	2024-06-12 23:40:40	ENABLE	👁	
xcdi6dNnwzVm	1984	2024-06-12 22:06:07	DISABLE	👁	
vF46or9VbWgt	La sombra del viento	2024-06-12 21:55:58	DISABLE	👁	

Figura 89. Pantalla de página personal de otros usuarios

Siendo una cuenta de administrador, le permite hacer gestión de usuario, es decir, se puede cambiar el rol de un cliente registrado, dándole un rol de administrador. Y le permite visualizar la página personal de otros usuarios.

LIBRARY
Home Management Orders Records
Emilio

USER BOOK AUTHOR Book Type LIBRARY

Books Management

Add Book


BookID	Name	Status	View	Management
WeDqA0qxIKXx	Don Quijote de la Mancha	ENABLE	👁	🚫 🗑
FKfD5WNf5afv	Cien años de soledad	ENABLE	👁	🚫 🗑
vF46or9VbWgt	La sombra del viento	DISABLE	👁	✅ 🗑
hvyrlq9IMHGE	Pride and Prejudice	DISABLE	👁	✅ 🗑
L0osyn5iMN5o	The Great Gatsby	OCCUPIED	👁	🗑

Items per page:
1 - 5 of 7
< >

Figura 90. Pantalla de gestión de libros



Bookid: WeDqA0qxlKXx
Barcode: 11111111
Book Name: Don Quijote de la Mancha
ISBN: 978-84-206-8184-5
Language: Spanish
Publisher: Editorial Planeta
Deposit: €10
Author: Miguel de Cervantes Saavedra[spain]
Book Type:



Modify book data

BookID WeDqA0qxlKXx	Book name Don Quijote de la Mancl
Publisher Editorial Planeta	Deposit 10
Language Spanish	

Description
Don Quijote de la Mancha es una novela escrita por el autor español Miguel de Cervantes Saavedra. Publicada su primera parte con el título de 'El ingenioso hidalgo don Quijote de la Mancha' a comienzos de 1605, es la obra más destacada de la literatura española y una de las principales de la literatura universal. En ella, un hidalgo envejecido, de nombre Alonso Quijano, pierde su cordura y decide convertirse en caballero andante, autodenominándose Don

Maximum 5000 characters

Book Author

Figura 91. Pantalla de modificar libros

A un administrador le permite también hacer gestión de libros, como visualizar la información del libro, su estado y poner en disponible el libro para que los clientes puedan prestar, así como eliminar un libro y modificar la información del libro.

USER BOOK AUTHOR Book Type LIBRARY

AUTHOR Management

Search

AuthorID	Name	Nationality	Management
D8UrpfrbWJuB	Miguel de Cervantes Saavedra	spain	
fyWjLBzUv060	Gabriel García Márquez	Colombia	
OualSwLXOCzq	Carlos Ruiz Zafón	spain	
3ByASD91HE49	Jane Austen	United Kingdom	
BL2VRBJhB0UX	F. Scott Fitzgerald	United States	

Items per page: 5 1 - 5 of 7 < >

Add Author ▼

Figura 92. Pantalla de gestión de autores

Miguel de Cervantes Saavedra ✎

Nationality: spain

Description: Miguel de Cervantes Saavedra (1547-1616) fue un novelista, poeta y dramaturgo español considerado la máxima figura de la literatura española. Es universalmente conocido por haber escrito 'Don Quijote de la Mancha', que muchos críticos han descrito como la primera novela moderna y una de las mejores obras de la literatura mundial. La vida de Cervantes, marcada por las adversidades, incluyó periodos como soldado, durante los cuales fue herido en la batalla de Lepanto, y como

Figura 93. Pantalla de perfil de autor

El administrador también le permite hacer gestión de autores, visualizar la información del autor, añadir autor, así como modificar el perfil del autor y eliminar el autor.



LIBRARY Home Management Orders Records Emilio

USER BOOK AUTHOR **Book Type** LIBRARY

All Type

Novela ✕ Novela de realismo mágico ✕ Novela de misterio y literatura gótica ✕
Novela de ficción clásica ✕ Novela de ficción modernista ✕ Novela distópica ✕
Colección de cuentos ✕

Add Book Type

Type Name

Description

Maximum 5000 characters

Modify Book Type

Type Name

Description

Maximum 5000 characters

Figura 94. Pantalla de tipo de libros

Por último, el administrador también puede hacer gestión de tipo de libro, como añadir el tipo del libro, modificar el tipo del libro.

LIBRARY Home Management Orders Records Emilio

Return box Book Borrow Order List Book Return Order List

Return box Refresh

Search

Reference	Book	Status	User	Status of return	View
JwDVYQRQWQVZ	The Great Gatsby [ID:L0osyn5IMN5o]	WAITING_FOR_VERIFICATION	+34654321000	✓ ✕	👁

Items per page: 5 1 - 1 of 1

Figura 95. Pantalla de caja de devolución de libros

feedback page

Degree of book damage

PERFECT
SLIGHTLY_DAMAGED
MISSING_PAGES
BADLY_DAMAGED
IRREPARABLE

Addendum

The books are very well protected

Maximum 400 characters

submit

Figura 96. Pantalla de comentarios de libros

Con la opción de 'Orders Records', le permite al administrador gestionar los pedidos de libros de la biblioteca. En 'return box', le permite aprobar o rechazar el pedido de devolución de libro por parte de cliente, el administrador puede verificar si el libro ha sido realmente devuelto. En caso afirmativo, el sistema pide al administrador llenar un formulario sobre el estado de daño del libro y con esto se hace referencia de devolución de depósito al cliente.

LIBRARY
Home Management Orders Records
Emilio

Return box
Book Borrow Order List
Book Return Order List

Lending list

Refresh

Only Show no Return
Show Beyond 30 Days no Return
Send reminder email

Send order overdue email
Ban users with orders that have not been returned for more than 30 days

Reference	Book	BorrowTime	LimitTime	User	Return
xK3zbGZ4e0rg	Don Quijote de la Mancha[ID:WeDqA0qxIKXx]	2024-06-13 00:05:19	2024-06-26 23:59:59	+34654321000	
JwDVYQRQWQVZ	The Great Gatsby[ID:L0osyn5iMN5o]	2024-06-13 10:25:02	2024-08-15 23:59:59	+34654321000	
gnw7Os6Y0imC	Ficciones[ID:T1bZXGOvLJ9U]	2024-06-14 00:48:42	2024-08-09 23:59:59	+34654321000	Return
KIW6tNISFXqq	Cien años de soledad[ID:FKfD5WNf5afv]	2023-05-09 09:48:37	2024-06-15 23:59:59	+34654321123	Return
R86NToJvPXtx	Don Quijote de la Mancha[ID:WeDqA0qxIKXx]	2023-06-14 16:39:01	2023-06-15 23:59:59	+34654321123	Return

Items per page:

5

1 - 5 of 5

Figura 97. Pantalla de lista de préstamo de libros de administrador (1)



Return box **Book Borrow Order List** Book Return Order List

Lending list Refresh

Search

Only Show no Return Show Beyond 30 Days no Return Send reminder email

Order overdue email Ban users with orders that have not been returned for more than 30 days

	Book	BorrowTime	LimitTime	User	Return	View
Org	Don Quijote de la Mancha[ID:WeDqA0qxIKXx]	2024-06-13 00:05:19	2024-06-26 23:59:59	+34654321000		
WQVZ	The Great Gatsby[ID:L0osyn5iMN5o]	2024-06-13 10:25:02	2024-08-15 23:59:59	+34654321000		
JimC	Ficciones[ID:T1bZXGOvLJ9U]	2024-06-14 00:48:42	2024-08-09 23:59:59	+34654321000	Return	
qq	Cien años de soledad[ID:Fkfd5WNf5afv]	2023-05-09 09:48:37	2024-06-15 23:59:59	+34654321123	Return	
Xtx	Don Quijote de la Mancha[ID:WeDqA0qxIKXx]	2023-06-14 16:39:01	2023-06-15 23:59:59	+34654321123	Return	

Items per page: 5 1 - 5 of 5 < >

Figura 98. Pantalla de lista de préstamo de libros de administrador (2)

En 'Book Borrow Order List', le permite al administrador hacer varias operaciones relacionadas con los libros que han sido prestados. Le permite visualizar la información de los libros. Si el libro todavía no está en estado 'devuelto' y está cerca a la fecha límite de devolución, le aparece en color amarillo. Y si el libro ha llegado a la fecha límite y todavía no ha sido devuelto, le aparece en color rojo. Con los colores le permite al administrador realizar diferentes recordatorios por email. Además, en la columna de 'Return' permite al administrador ayudar a un cliente a devolver libro, en caso de que el cliente no conozca muy bien el proceso de devolución por el sistema web.

LIBRARY Home Management Orders Records Emilio

Return box Book Borrow Order List **Book Return Order List**

Returned list Refresh

Search

Reference	Book	ReturnStatus	Status of return	User	View
xK3zbGZ4e0rg	Don Quijote de la Mancha [ID:WeDqA0qxIKXx]	IS_RETURN	Normal	+34654321000	
JwDVYQRQWQVZ	The Great Gatsby [ID:L0osyn5iMN5o]	WAITING_FOR_VERIFICATION	Normal	+34654321000	
KIW6tNISFXqq	Cien años de soledad [ID:Fkfd5WNf5afv]	NO_RETURN	Normal	+34654321123	

Items per page: 5 1 - 3 of 3 < >

Figura 99. Pantalla de lista de devolución de libros de administrador

Con la opción de 'Book return Order List', le permite al administrador visualizar la información de los libros sobre su estado de devolución y el grado de daño del libro.

4. VISTA DE ADMINISTRADOR SUPERIOR

LIBRARY Home Management Orders Records super admin

13/Jun

Don Quijote de la Mancha [Language:Spanish]
Miguel de Cervantes Saavedra

Clear

23° MIN:20° MAX:24°
00:38:38 Madrid

Hello super admin!!

Have a good day!

Search

Language: All Type: All Search name: name

Only show ENABLE Show Borrow Count

Cien años de soledad [Spanish]
Gabriel García Márquez
€10

The Great Gatsby [English]
F. Scott Fitzgerald
€12

Ficciones [Spanish]
Jorge Luis Borges
€10

Don Quijote de la Mancha [Spanish]
Miguel de Cervantes Saavedra
€10

Figura 100. Página de inicio de administrador superior (1)

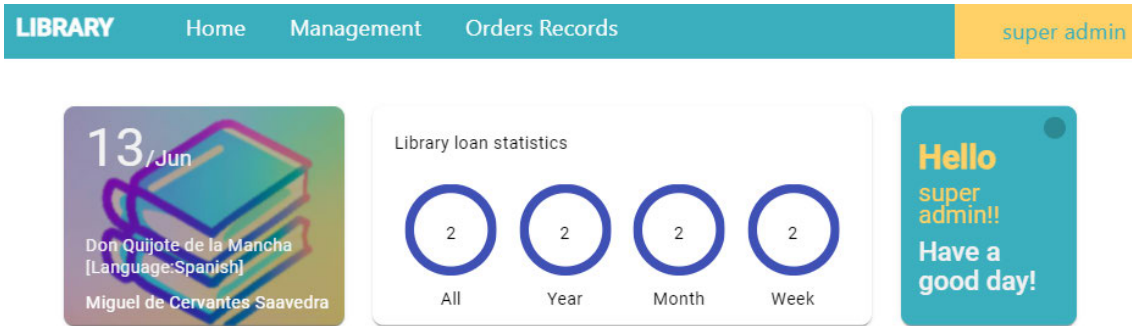


Figura 101. Página de inicio de administrador superior (2)

Al administrador superior se le muestra una página de inicio igual que un administrador normal una vez iniciada su cuenta en el sistema de biblioteca.

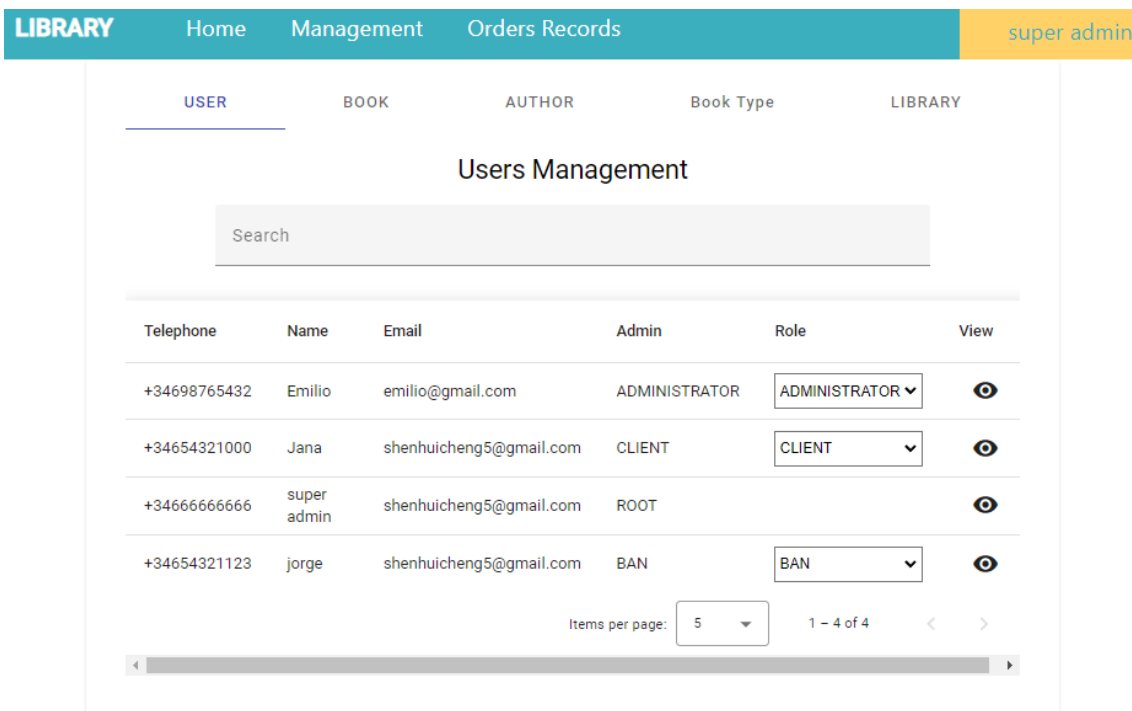


Figura 102. Pantalla de gestión de usuarios de administrador superior

LIBRARY Home Management Orders Records super admin

USER BOOK AUTHOR Book Type LIBRARY


Users Management

Search

Personal Page

Personal Information

Name: jorge
Email: shenhuicheng5@gmail.com
Birthdays: 1990-01-01
description: This user has not modified his profile
Registration Time: 2024-06-14 09:40:48



- Modify personal data
- Setting personal
- Modify Password

Figura 103. Pantalla de página personal de otros usuarios de administrador superior



LIBRARY Home Management Orders Records super admin

USER BOOK AUTHOR Book Type LIBRARY

Library Setting

Phone
6669996669

Email
bibliotecaMadrid@email.com

address
X.º XX XXXXXXXX, XX, 28001, XXXXXX, Madrid

Postal Code
28001

Business Hours
Monday to Friday 10:00 to 22:00 Saturday and Sunday 10:00 to 16:00

Introduction
this is test introduction

Maximum 150 characters

Google Mail Address
https://mail.google.com/

Instagram Address
https://www.instagram.com/

Facebook Address
https://www.facebook.com/

Discord Address
https://www.discord.com/

Twitter Address
https://www.twitter.com/

OK

Figura 104. Pantalla de gestión de información de biblioteca de administrador superior

Siendo un administrador superior, se le permite hacer todo lo que puede hacer un administrador normal, y se puede cambiar el rol de usuario a 'ban' o modificar la información personal o resetear la contraseña de otros usuarios. Además, le permite cambiar la información de la biblioteca con la opción de 'Library'.

5. VISTA DE USUARIO PROHIBIDO

LIBRARY Home Sign up Login

14/Jun
Pride and Prejudice
[Language:English]
Jane Austen

Clear
29° MIN:27°
MAX:30°
17:29:28 Madrid

Login
Sign up

Search

Language All Type All Search name name 654321123

Only show ENABLE Show Borrow Count

Cien años de soledad [Spanish]
Gabriel García Márquez
€10

The Great Gatsby [English]
F. Scott Fitzgerald
€12

Ficciones [Spanish]
Jorge Luis Borges
€10

Don Quijote de la Mancha [Spanish]
Miguel de Cervantes Saavedra
€10

Your account has been banned. Error

Figura 105. Pantalla de página de inicio de usuario prohibido

A los usuarios 'prohibidos' (ban), quienes no han devuelto los libros después de 30 días de la fecha límite, no se permite hacer login para acceder a las funcionalidades específicas con sus cuentas clientes.



CONCLUSIONES

Durante este Trabajo de Fin de Máster (TFM) hemos aplicado y expandido el conocimiento adquirido durante el período de estudios para construir una aplicación web de gestión bibliotecaria. Los clientes pueden crear cuentas para realizar operaciones de préstamo y devolución de libros, mientras que los administradores pueden gestionar los datos de la aplicación.

Adoptamos una arquitectura de separación frontend-backend para mejorar la eficiencia de la colaboración en equipo. La parte frontend ha utilizado Angular, lo que facilita el desarrollo y reduce la repetición y redundancia del código mediante la reutilización de componentes. El backend se ha construido con el marco de trabajo Spring Boot y aplicamos una arquitectura hexagonal, lo que proporciona una estructura de proyecto con bajo acoplamiento.

En cuanto a la base de datos, elegimos MongoDB. Para evitar la contaminación de datos de prueba con los datos del entorno de producción, implementamos la separación entre los entornos de producción y prueba. Durante el desarrollo del proyecto, utilizamos GitHub Actions para crear un entorno de trabajo integrado, logrando una compilación y prueba automatizadas, lo que mejoró significativamente la eficiencia y la calidad del desarrollo del proyecto.

Además, adoptamos el método de gestión de desarrollo ágil para responder de manera flexible a los cambios en los requisitos del proyecto y mejorar la eficiencia del proceso de desarrollo.

En conclusión, la implementación de este Trabajo de Fin de Máster (TFM) ha demostrado ser una excelente oportunidad para aplicar de manera práctica los conocimientos adquiridos durante nuestro curso académico. Al elegir tecnologías avanzadas y una arquitectura robusta, no solo hemos logrado crear una aplicación web de gestión bibliotecaria funcional y eficiente, sino que también hemos establecido una base sólida para futuras mejoras y escalabilidad. La integración de prácticas de desarrollo ágil y herramientas de CI/CD ha permitido un proceso de desarrollo más dinámico y ha minimizado los riesgos, asegurando un producto final de alta calidad. Este proyecto no solo cumple con los requisitos establecidos, sino que también proporciona una plataforma versátil para futuras innovaciones en el campo de la gestión bibliotecaria.

POSIBLES AMPLIACIONES

Para mejorar y expandir nuestra aplicación web de gestión bibliotecaria, se consideran varias ampliaciones futuras que enriquecerán la funcionalidad y la interacción del usuario. Estas ampliaciones incluyen:

1. **Comentarios de Usuarios sobre los Libros:** Permitirá a los usuarios dejar comentarios y valoraciones sobre los libros que han leído, fomentando una comunidad más interactiva y brindando información valiosa a otros lectores.
2. **Gestión y Visualización de la Ubicación de los Libros en las Estanterías:** Implementar un sistema para visualizar y gestionar la ubicación física de los libros dentro de la biblioteca facilita la búsqueda y el acceso a los recursos por parte de los usuarios.
3. **Módulo de Artículos Personales de los Usuarios:** Este módulo permitiría a los usuarios guardar y administrar información sobre artículos personales relacionados con la lectura, como notas de libros y listas de lectura.
4. **Módulo de Comunicación entre Usuarios:** Establecer un canal de comunicación para que los usuarios interactúen, discutan y compartan ideas sobre libros o temas relacionados, enriqueciendo la experiencia de la comunidad.
5. **Visualización de Perfiles Personales:** Los usuarios podrán crear y visualizar perfiles personales que muestren sus intereses literarios, historial de lectura y contribuciones a la comunidad.
6. **Función de Compartir Favoritos:** Facilitar a los usuarios la posibilidad de compartir sus libros favoritos con otros dentro de la plataforma, lo que puede incentivar la interacción y el descubrimiento de nuevos libros.
7. **Módulo de Venta de Libros:** Ampliar la aplicación para incluir una función de comercio electrónico que permita la compra y venta de libros entre usuarios, ofreciendo una forma adicional de acceso a recursos literarios.

Actualmente, el proyecto aún no está desplegado en Internet, pero estamos en la búsqueda de una plataforma de despliegue gratuita que permita a un mayor número de usuarios acceder y beneficiarse de estas funcionalidades. Con estas ampliaciones, nuestra aplicación no solo se convierte en una herramienta más completa y útil para la gestión bibliotecaria, sino que también fomenta una comunidad activa y comprometida en torno a la lectura y el intercambio de conocimientos.



BIBLIOGRAFÍA

- [1] Wikipedia, <<Agile_inception_deck,>> [En línea]. Available: https://www.scrummanager.com/bok/index.php/Agile_inception_deck [Último acceso: 5 de mayo de 2024]
- [2] Institutoagile, <<estimaciones-con-story-points-en-scrum,>> [En línea]. Available: <https://www.institutoagile.com/post/estimaciones-con-story-points-en-scrum> [Último acceso: 28 de mayo de 2024]
- [3] Asana, <<planning-poker,>> [En línea] Available: <https://asana.com/es/resources/planning-poker> [Último acceso: 28 de mayo de 2024]
- [4] Uax, <<cuanto-cobra-un-ingeniero-informático,>> [En línea]. Available: <https://www.uax.com/blog/ingenieria/cuanto-cobra-un-ingeniero-informatico> [Último acceso: 30 de mayo de 2024]
- [5] Cei, <<diseñador-ux-ui-que-hace-cuanto-gana,>> [En línea] Available: <https://cei.es/diseñador-ux-ui-que-hace-cuanto-gana/> [Último acceso: 30 de mayo de 2024]
- [6] Grover, <<Aplicación-de-alquiler-de-tecnología,>> [En línea] Available: <https://www.grover.com/es-es> [Último acceso: 30 de mayo de 2024]
- [7] E-dea, <<mongodb-por-que-usarlo,>> [En línea] Available: <https://www.e-dea.co/mongodb-por-que-usarlo> [Último acceso: 09 de junio de 2024]
- [8] Arteco, <<tutorial-springboot,>> [En línea] Available: <https://www.arteco-consulting.com/articulos/tutorial-springboot/> [Último acceso: 09 de junio de 2024]
- [9] Salessystems, <<arquitectura-hexagona,>> [En línea] Available: <https://salesystems.es/arquitectura-hexagonal/#:~:text=La%20arquitectura%20hexagonal%20facilita%20la,de%20que%20funcione%20sin%20problemas.> [Último acceso: 09 de junio de 2024]
- [10] Dev, <<arquitectura-hexagona,>> [En línea] Available: <https://dev.to/dennysimarquez/la-arquitectura-hexagonal-un-enfoque-para-desarrollar-aplicaciones-robustas-en-react-angular-y-vue-con-ejemplos-de-codigo-l4l> [Último acceso: 09 de junio de 2024]
- [11] Campusmvp, <<angular,>> [En línea] Available: <https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx> [Último acceso: 09 de junio de 2024]
- [12] Ciberseguridad, <<autenticacion-jwt,>> [En línea] Available: https://ciberseguridad.com/guias/prevencion-proteccion/autenticacion-jwt/#Usar_JWT_para_la_autenticacion_de_API [Último acceso: 09 de junio de 2024]

[13]Knowmadmood, <<importancia-del-testing-de-software-y-pruebas>> [En línea]
Available: <https://www.knowmadmood.com/blog/la-importancia-del-testing-de-software-y-de-la-automatizacin-de-pruebas> [Último acceso: 09 de junio de 2024]

[14]Google, <<SMTP>> [En línea] Available:
<https://support.google.com/a/answer/176600?hl=es> [Último acceso: 09 de junio de 2024]