

Diseño del módulo de préstamos de recursos en la plataforma COMA y desarrollo e implantación de los servicios orientados al actor propietario.

Felipe Andres Cepeda Ortiz y Jeicob Gilmar Restrepo Gómez

Trabajo de Grado para Optar al Título de Ingeniero de Sistemas

Director

Luis Ignacio González Ramírez

Magister en Informática

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2026

### **Dedicatoria**

Este título está especialmente dedicado a Dios, por darme la fuerza y la salud para continuar y culminar mi carrera de manera satisfactoria.

También se lo dedico a mi papá, Luis Felipe, que desde el cielo me ha brindado la sabiduría para lograr todo lo que me propongo; y a mi mamá, Fanny, por ser una mujer luchadora, por sacarnos siempre adelante, y por darme ese amor incondicional que me acompaña desde el día en que llegué a este mundo. Gracias, mami, por todo.

A mis hermanos Mauricio y Emelith, por su compañía constante y por ser ese apoyo tan valioso en cada momento.

Y, por último, a todos mis seres queridos, por ser esa ancla que mantiene mis pies sobre la tierra. Este título es para ellos, por ser parte fundamental de este proceso, por desearme siempre lo mejor y por ser mi todo en esta vida.

*Felipe Cepeda*

Dedico este logro a mis padres, Freddy Restrepo y Tatiana Gómez, quienes con su amor, esfuerzo y ejemplo me han guiado y acompañado en cada paso de mi vida. Gracias por enseñarme a no rendirme, por confiar en mí incluso en los momentos más difíciles y por estar siempre presentes para apoyarme.

A mis hermanos, Santiago y Angie, por su cariño, motivación y por recordarme que la familia es una de las mayores fortalezas que uno puede tener.

También a mis familiares que, en distintos momentos, me brindaron su apoyo y compañía, contribuyendo de una u otra forma a que hoy pueda alcanzar esta meta.

Este título no solo representa mi esfuerzo, sino también el de todos ustedes, que han sido parte de este camino.

*Jeicob Restrepo*

### **Agradecimientos**

A mi director, profesor Luis Ignacio, por su valiosa guía, su apoyo constante y la confianza depositada en mí durante todo este proceso. Su experiencia y liderazgo no solo han fortalecido mi formación profesional, sino que también han contribuido a mi crecimiento personal. Me siento profundamente agradecido por haber contado con su acompañamiento en esta etapa tan importante.

A mi familia, por ser mi pilar en cada momento, incluso en las circunstancias más difíciles. Su paciencia, apoyo incondicional y amor han sido la fuerza que me impulsó a seguir adelante y a alcanzar cada una de mis metas.

A mis compañeros, por caminar junto a mí en este recorrido, compartiendo esfuerzos, alegrías y desafíos, y por fortalecerme en los momentos en que más lo necesitaba.

Y, de manera muy especial, a mi compañero de grado, Jeicob, por ser parte esencial de este logro. Gracias por su apoyo, su compromiso y por enfrentar conmigo cada reto. Este triunfo también es suyo.

*Felipe Cepeda*

A mi director, profesor Luis Ignacio, por su orientación y apoyo a lo largo de este trabajo. Su compromiso, paciencia y consejos fueron fundamentales para avanzar con seguridad y aprender más allá de lo académico. Su acompañamiento ha dejado una huella valiosa en mi formación profesional y personal.

A mi familia, por ser la base que me sostiene en todo momento. Gracias por su amor, su confianza y por darme fuerzas incluso en los días más complicados. Sin ustedes, este camino habría sido mucho más difícil.

A mis compañeros, por compartir cada paso de este proceso, las jornadas de esfuerzo, las metas alcanzadas y los retos superados. Su compañía hizo más llevadero este recorrido y me motivó a seguir adelante.

Y, por supuesto, a mi compañero de grado y amigo, Felipe, por su dedicación, apoyo constante y por afrontar conmigo cada desafío. Este logro es tanto mío como suyo, y me enorgullece haberlo alcanzado juntos.

*Jeicob Restrepo*

## Tabla de Contenido

	<b>Pág.</b>
Introducción .....	18
1. Planteamiento y justificación del problema .....	19
1.1 Definición de la situación actual .....	19
1.2 Justificación .....	20
1.3 Objetivos .....	23
1.3.1 Objetivo general .....	23
1.3.2 Objetivos específicos .....	23
2. Marco de referencia .....	23
2.1 Modelado UML .....	23
2.2 Base de datos relacional .....	24
2.3 MySQL 8.3.0 .....	25
2.4 Java .....	25
2.5 JSP (JavaServer Pages) .....	26
2.6 Apache Tomcat 9.0.88 .....	27
2.7 Html .....	28
2.8 Framework de estilos .....	29
2.9 JavaScript .....	30
2.10 Git y GitHub .....	31
2.11 CSS .....	32
2.12 Servlet .....	32
3. Metodología .....	33

3.1 Enfoque general .....	34
3.2 Metodología de desarrollo: Scrum.....	35
3.2.1 Ceremonias y artefactos implementados .....	35
3.3 Herramientas de modelado .....	36
3.4 Técnicas de desarrollo.....	36
3.5 Gestión de configuración y control de versiones .....	37
3.6 Pruebas.....	37
3.7 Despliegue.....	38
3.8 Documentación .....	38
3.9 Actividades de análisis y planificación.....	38
3.10 Desarrollo de componentes utilizando metodología ágil.....	39
3.10 Prácticas implementadas.....	39
3.12 Asignación de roles en Scrum.....	41
3.13 Reflexión final sobre el uso de Scrum .....	42
3.14 Implementación de pruebas funcionales y de usabilidad.....	43
3.15 Despliegue en la base de pruebas.....	44
3.16 Optimización de la calificación y asignaciones:.....	46
3.17 Despliegue real.....	46
3.18 Objetivos del despliegue en producción .....	47
3.19 Construcción y validación de prototipos.....	49
4. Desarrollo del proyecto.....	51
4.1 Análisis y diseño detallado .....	51
4.1.1 Identificación de actores y casos de uso .....	51

4.1.1.1 Casos de uso del actor propietario. ....	51
4.1.1.2 Casos de uso del actor administrador.....	52
4.1.2 Diagramas de actividades .....	52
4.1.2.1 Proceso de préstamo de un recurso.....	53
4.1.2.2 Proceso de creación de recurso .....	53
4.1.3 Diagramas de secuencia.....	54
4.1.3.1 Diagrama de secuencia de creación de un recurso.....	54
4.1.3.2 Diagrama de secuencia de asignación por derecha.....	55
4.1.3.3 Diagrama de secuencia de creación de un almacén .....	56
4.1.3.4 Diagrama de secuencia de aceptación de solicitudes.....	56
4.1.3.5 Diagrama de secuencia de gestión de cuentas .....	57
4.1.3.6 Diagrama de secuencia de gestión de recursos administrador.....	58
4.1.4 Modelado de la base de datos .....	59
4.1.5 Maquetación.....	60
4.1.5.1 Mockup de la pestaña “Mis recursos” .....	61
4.1.5.2 Mockup de la pestaña “Configurar recursos” .....	62
4.1.5.3 Mockup de la visualización de solicitudes.....	63
4.1.5.4 Mockup de detalles de una solicitud.....	64
4.1.5.5 Mockup del historial de préstamos .....	65
4.2 Implementación de servicios del actor propietario .....	66
4.2.1 Almacenes.....	66
4.2.2 Creación de almacenes.....	67
4.2.3 Creación de recursos .....	68

4.2.4 Edición de recursos .....	69
4.2.5 Creación de un combo .....	70
4.2.6 Asignaciones por derecha de un recurso.....	71
4.2.7 Agregar recursos existentes a los almacenes .....	72
4.2.8 Solicitudes de recursos.....	73
4.2.9 Aceptación o rechazo de un recurso por parte del propietario.....	74
4.2.10 Solicitudes en curso, cuando ya fue aceptada la solicitud .....	75
4.2.11 Recursos devueltos.....	76
4.2.12 Historial de solicitudes.....	76
4.2.13 Calificación .....	77
4.3 Implementación de servicios del actor Administrador .....	78
4.3.1 Generador de PDF y CSV .....	78
4.3.2 Gestionar usuarios.....	79
4.3.3 Gestión de recursos administrador.....	80
4.4 Integración con la Plataforma COMA .....	81
4.5 Pruebas y Validación Continua.....	82
4.6 Prueba funcional: Creación de almacenes .....	83
4.7 Prueba funcional: Creación de recursos.....	84
4.8 Prueba funcional: Agregar recursos ya creados a los almacenes.....	85
4.9 Prueba funcional: Gestionar cuentas.....	86
4.10 Prueba funcional: Exportar información en PDF y CSV .....	87
4.11 Despliegue en entorno de pruebas .....	88
4.12 Principales desafíos y soluciones.....	88

4.13 Impacto y valor agregado.....	89
5. Conclusiones.....	89
5.1 Trabajo a futuro.....	90
Referencias Bibliográficas .....	93

### Lista de Figuras

Figura 1 <i>Ciclo y actores de la metodología ágil SCRUM</i> .....	42
Figura 2 <i>Modelo de casos de uso – Actor propietario</i> .....	51
Figura 3 <i>Modelo de casos de uso – Actor administrador</i> .....	52
Figura 4. <i>Diagrama de actividades – Proceso de préstamo</i> .....	53
Figura 5. <i>Diagrama de actividades – Proceso de creación de recurso</i> .....	53
Figura 6 <i>Diagrama de secuencia de creación de un recurso.</i> .....	54
Figura 7 <i>Diagrama de secuencia de asignación por derecha</i> .....	55
Figura 8 <i>Diagrama de secuencia de creación de un almacén</i> .....	56
Figura 9 <i>Diagrama de secuencia de aceptación de solicitudes</i> .....	56
Figura 10 <i>Diagrama de secuencia de gestión de cuentas</i> .....	57
Figura 11 <i>Diagrama de secuencia de gestión de recursos administrador</i> .....	58
Figura 12 <i>Modelo entidad relación del módulo</i> .....	60
Figura 13 <i>Mockup de la pantalla principal de gestión de recursos</i> .....	62
Figura 14 <i>Mockup del formulario de edición y configuración de recursos</i> .....	63
Figura 15 <i>Mockup de la pantalla de listado de solicitudes recibidas</i> .....	64
Figura 16 <i>Mockup de la pantalla de detalles de una solicitud específica</i> .....	65
Figura 17 <i>Mockup de la pantalla de historial de préstamos</i> .....	66
Figura 18 <i>Vista principal del actor propietario</i> .....	67
Figura 19 <i>Vista sobre la creación de un almacén</i> .....	68
Figura 20 <i>Vista sobre la creación de un recurso</i> .....	69
Figura 21 <i>Vista sobre la edición de un recurso</i> .....	70

Figura 22 Vista sobre la creación de un combo .....	71
Figura 23 Vista sobre una asignación por derecha .....	72
Figura 24 Vista sobre la agregación de un recurso a varios almacenes .....	73
Figura 25 Vista sobre las solicitudes que le hacen al propietario.....	74
Figura 26 Vista sobre la aceptación o rechazo de las solicitudes .....	75
Figura 27 Vista de las solicitudes que están en curso .....	76
Figura 28 Vista sobre los recursos que se devolvieron al propietario .....	76
Figura 29 Vista sobre el historial de solicitudes.....	77
Figura 30 Vista sobre la calificación del actor solicitante .....	77
Figura 31 Vista sobre la descarga en PDF .....	78
Figura 32 Vista sobre la descarga en CSV .....	78
Figura 33 Vista sobre la deshabilitación de un usuario .....	79
Figura 34 Vista sobre usuario deshabilitado .....	80
Figura 35 Vista sobre la gestión de recursos.....	81

## Lista de Apéndices

Apéndice A – Video demostrativo: Aceptación de solicitudes .....
Apéndice B – Video demostrativo: Agregar recursos a almacén .....
Apéndice C – Video demostrativo: Asignación de recursos .....
Apéndice D – Video demostrativo: Calificación de usuario .....
Apéndice E – Video demostrativo: Creación de almacén .....
Apéndice F – Video demostrativo: Creación de recurso .....
Apéndice G – Video demostrativo: Descarga de informes.....
Apéndice H – Video demostrativo: Edición de recursos .....
Apéndice I – Video demostrativo: Gestión de cuenta .....
Apéndice J – Video demostrativo: Gestión de recursos .....
Apéndice K – Video demostrativo: Vista de usuario inhabilitado .....
Apéndice L – Plan de pruebas del módulo de recursos .....

Los apéndices están adjuntos y puede visualizarlos en la base de datos de la biblioteca UIS.

## Glosario

**CALUMET:** grupo de innovación y desarrollo de la Universidad Industrial de Santander, dedicado a crear soluciones tecnológicas y proyectos de investigación. Su enfoque está en el desarrollo de software, innovación en plataformas web y aplicaciones educativas, contribuyendo al fortalecimiento de la comunidad académica.

**COMA:** acrónimo de la plataforma web Comunidad Académica, es el proyecto de portal web de las escuelas y facultades de la Universidad Industrial de Santander.

**Elise:** Framework de estilos creado por el grupo CALUMET que proporciona lineamientos y componentes visuales para aplicaciones web. Su objetivo es estandarizar el diseño gráfico y la experiencia de usuario en proyectos institucionales, asegurando coherencia estética y usabilidad.

**Git:** sistema de control de versiones distribuido ampliamente usado para el seguimiento de cambios en archivos y la colaboración en proyectos de desarrollo de software. Permite trabajar en equipo de forma paralela, guardar históricos de modificaciones, revertir errores y gestionar ramas de desarrollo.

**GitHub:** plataforma de desarrollo colaborativo en la nube que utiliza Git como sistema de control de versiones. Ofrece funcionalidades adicionales como *pull requests*, revisión de código, integración continua y gestión de proyectos, lo que lo convierte en un espacio esencial para la programación colaborativa a nivel mundial.

**Repositorio:** espacio de almacenamiento donde se guarda el código fuente de un proyecto junto con su historial de cambios, versiones y documentación. Puede estar alojado en la nube (por ejemplo, en GitHub o GitLab) o en servidores locales, permitiendo la colaboración ordenada entre varios desarrolladores.

**Servidor web:** es un ordenador o software que almacena, procesa y entrega archivos de sitios web a los usuarios a través de un navegador, utilizando el protocolo HTTP o HTTPS. Los servidores web permiten acceder a páginas, aplicaciones y recursos en línea, siendo la base de la experiencia en Internet.

**Servlet:** componente de Java que se ejecuta en un servidor de aplicaciones. Su función principal es recibir solicitudes (requests) desde el cliente, procesarlas en el servidor y generar respuestas dinámicas, generalmente en aplicaciones web. Los servlets permiten integrar lógica de negocio con interfaces web, formando parte clave del modelo Java EE.

**MySQL:** sistema de gestión de bases de datos relacional (RDBMS) de código abierto. Es ampliamente utilizado en aplicaciones web por su velocidad, fiabilidad y facilidad de integración con lenguajes como PHP, Java y Python. Permite almacenar, organizar y consultar información de manera estructurada usando el lenguaje SQL.

**JavaScript (JS):** lenguaje de programación interpretado, principalmente usado en el desarrollo web para añadir interactividad, dinamismo y lógica en el lado del cliente (navegador).

**Java:** lenguaje de programación orientado a objetos, multiplataforma y de propósito general. Es ampliamente utilizado en el desarrollo de aplicaciones empresariales, móviles y web.

**Jsp:** tecnología de Java que permite crear páginas web dinámicas mezclando código Java con HTML. Se ejecuta en el servidor y genera contenido que se envía al navegador del usuario.

## Resumen

**Título:** Diseño del módulo de préstamos de recursos en la plataforma COMA y desarrollo e implantación de los servicios orientados al actor propietario\*

**Autor:** Felipe Andres Cepeda Ortiz y Jeicob Gilmar Restrepo Gomez\*\*

**Palabras Clave:** Préstamos, Recursos, COMA, Eficiencia, Módulo

**Descripción:** La plataforma COMA (Comunidad Académica) ha sido una herramienta esencial para la Universidad Industrial de Santander (UIS), integrando a estudiantes, profesores, egresados y personal administrativo en un único ecosistema digital. Sin embargo, se ha identificado que la gestión de recursos físicos como herramientas de laboratorio, equipos tecnológicos, instrumentos musicales y otros elementos académicos y extracurriculares carece de un sistema eficiente que permita su administración adecuada. Esta carencia genera múltiples inconvenientes, como la pérdida de tiempo en procesos manuales, baja disponibilidad o subutilización de los recursos, y dificultades para el seguimiento y control de los préstamos realizados. Esto afecta directamente la calidad de las actividades académicas y administrativas, así como la experiencia de los usuarios al interactuar con estos recursos. En respuesta a esta problemática, surge la necesidad de diseñar e implementar un módulo especializado en la gestión de préstamos dentro de la plataforma COMA. Este módulo se enfocará en optimizar los procesos asociados a la gestión de recursos mediante el desarrollo de servicios orientados principalmente al actor propietario, responsable de registrar, gestionar y hacer seguimiento a los recursos, así como aprobar o rechazar solicitudes de préstamo. Además, para garantizar el correcto funcionamiento del sistema, se desarrollarán algunos servicios clave orientados al administrador, como la gestión de usuarios, la edición o eliminación de recursos, y la generación de reportes. De esta manera, se busca establecer un sistema integral que permita el uso eficiente de los recursos y facilite su acceso a la comunidad académica. Al ser implementado en una plataforma familiar para los usuarios, este módulo asegura una rápida adopción, optimizando los procesos académicos y administrativos y fomentando un uso más efectivo de los recursos disponibles.

---

\* Trabajo de Grado

\*\* Facultad de Fisicomecánicas. Escuela de Ingeniería de Sistemas. Ingeniería de Sistemas.  
Director: Luis Ignacio González Ramírez. Magíster en Tecnología.

### Abstract

**Title:** Design of the Resource Loan Module in the COMA Platform and Development and Implementation of Services Oriented to the Owner Actor\*

**Author(s):** Felipe Andrés Cepeda Ortiz and Jeicob Gilmar Restrepo Gómez\*\*

**Key Words:** Loans, Resources, COMA, Efficiency, Module

**Description:** The COMA (Academic Community) platform has been an essential tool for the Industrial University of Santander (UIS), integrating students, professors, alumni, and administrative staff into a single digital ecosystem. However, it has been identified that the management of physical resources—such as laboratory tools, technological equipment, musical instruments, and other academic or extracurricular items—lacks an efficient system for proper administration. This absence results in multiple issues, such as wasted time due to manual processes, low availability or underutilization of resources, and difficulties in tracking and controlling issued loans. These limitations directly affect the quality of academic and administrative activities, as well as the overall user experience when interacting with these resources. In response to this problem, the need arises to design and implement a specialized module for loan management within the COMA platform. This module will focus on optimizing processes related to resource management through the development of services primarily oriented toward the “owner” actor, responsible for registering, managing, and monitoring resources, as well as approving or rejecting loan requests. Additionally, to ensure the proper functioning of the system, certain key services aimed at the administrator will be developed, such as user management, resource editing or deletion, and report generation. In this way, the goal is to establish a comprehensive system that enables efficient use of resources and facilitates their access to the academic community. By being implemented on a platform already familiar to users, this module ensures quick adoption, optimizes academic and administrative processes, and promotes more effective use of available resources.

---

\* Degree Work

\*\* Faculty Of Physical-Mechanical Engineering. Director Luis Ignacio Gonzáles Ramírez, MS. in Computer Science

## Introducción

En la actualidad, las plataformas digitales juegan un papel fundamental en la gestión y optimización de los recursos en las instituciones académicas. La Universidad Industrial de Santander (UIS) cuenta con la plataforma COMA, que integra a los diversos actores de la comunidad académica —estudiantes, profesores, egresados y personal administrativo— en un ecosistema digital unificado. Sin embargo, uno de los retos que persiste es la gestión eficiente de los recursos físicos, tales como herramientas de laboratorio, equipos tecnológicos e instrumentos musicales, entre otros, los cuales son esenciales para el desarrollo de actividades académicas y extracurriculares.

Actualmente, la administración de estos recursos se realiza en muchos casos mediante procesos manuales que generan demoras, subutilización, y dificultades en el control y seguimiento de los préstamos. Esto impacta negativamente en la experiencia de los usuarios y en la calidad de las actividades desarrolladas. Por esta razón, se hace imprescindible la creación de un módulo especializado dentro de la plataforma COMA que permita automatizar y optimizar la gestión de préstamos de recursos.

Este proyecto tiene como objetivo diseñar, desarrollar e implementar un módulo de préstamos orientado principalmente al actor propietario, responsable de administrar y dar seguimiento a los recursos, así como aprobar o rechazar las solicitudes de préstamo. Además, se contemplan servicios para el administrador del sistema, que permitan gestionar usuarios, editar recursos y generar reportes, garantizando así el correcto funcionamiento y la eficiencia del sistema en su conjunto.

La implementación de este módulo busca mejorar la disponibilidad y el uso eficiente de los recursos académicos, facilitando su acceso y contribuyendo a un mejor desempeño

institucional. Al integrarse en una plataforma ya conocida por la comunidad, se espera una rápida adopción y un impacto positivo tanto en los procesos académicos como administrativos.

## 1. Planteamiento y justificación del problema

### 1.1 Definición de la situación actual

La Universidad Industrial de Santander (UIS) cuenta con la plataforma COMA (Comunidad Académica), un ecosistema digital unificado que integra y facilita la interacción entre estudiantes, profesores, egresados y personal administrativo. Esta plataforma se ha consolidado como una herramienta fundamental para la gestión académica y administrativa de la institución.

Sin embargo, a pesar de su amplio alcance y capacidades, se ha identificado una brecha crítica en su funcionalidad: la gestión de préstamos de recursos físicos. Actualmente, procesos esenciales para la operación académica, como el préstamo de herramientas de laboratorio, equipos tecnológicos, instrumentos musicales y otros materiales, se manejan mediante métodos tradicionales y desconectados de la plataforma.

La situación actual se caracteriza por lo siguiente:

- **Procesos Manuales y Descentralizados:** La gestión de préstamos se realiza a través de diversos medios como planillas físicas, hojas de cálculo electrónicas (por ejemplo, en Excel), y comunicación directa. Esta descentralización genera un mayor esfuerzo, falta de estandarización y una alta probabilidad de error humano.
- **Falta de Visibilidad y Control:** No existe un sistema que ofrezca una visión en tiempo real de la disponibilidad, ubicación o estado de los recursos. Esto provoca conflictos por el uso simultáneo, subutilización de activos valiosos o, por el contrario, su saturación en periodos críticos.

- **Dificultades en el Seguimiento y la Trazabilidad:** El historial de préstamos, las fechas de devolución, y el estado de las solicitudes (pendientes, aprobadas, rechazadas). Esto complica enormemente las tareas de auditoría, mantenimiento y rendición de cuentas.
- **Ineficiencia en la Experiencia del Usuario:** Los solicitantes y los propietarios o administradores de los recursos deben invertir un tiempo considerable en coordinar manualmente cada préstamo, desde la solicitud hasta la devolución, sin contar con un canal formal, ágil y transparente para hacerlo.

Esta realidad operativa conlleva una serie de consecuencias negativas que impactan directamente la eficiencia institucional y la experiencia de la comunidad universitaria: pérdida de tiempo valioso en trámites burocráticos, uso ineficiente de los recursos institucionales, frustración entre los usuarios y un mayor riesgo de pérdida o mal uso de los materiales.

Por tanto, se evidencia la necesidad imperante de modernizar y digitalizar estos procesos mediante el diseño e implementación de un módulo de préstamos de recursos integrado a la plataforma COMA, que sirva como una solución única, centralizada y eficiente para toda la comunidad UIS.

## **1.2 Justificación**

La gestión de recursos físicos dentro de instituciones académicas como la Universidad Industrial de Santander (UIS) representa un proceso fundamental que incide directamente en la calidad de las actividades formativas e investigativas. Sin embargo, a pesar de su relevancia, en la actualidad este proceso se desarrolla de manera mayoritariamente manual, con prácticas descentralizadas y sin el respaldo de un sistema unificado que facilite un control eficiente y confiable.

La plataforma COMA ha logrado consolidarse como un espacio digital que integra a la comunidad universitaria en diferentes ámbitos, pero hasta ahora carece de un módulo especializado para la administración de préstamos de recursos físicos. Esta ausencia genera una serie de problemáticas visibles en el día a día:

- **Pérdida de tiempo:** los registros, seguimientos y devoluciones se realizan de forma manual, consumiendo tiempo valioso tanto de docentes y administrativos como de los mismos estudiantes.
- **Falta de visibilidad y control:** no se cuenta con una herramienta centralizada que muestre en tiempo real la disponibilidad, el estado ni el historial de uso de los recursos, lo que conlleva a subutilización o incluso a conflictos por solicitudes simultáneas.
- **Dificultad en la trazabilidad:** el seguimiento de préstamos activos, fechas de devolución y procesos de aprobación se apoya en métodos como hojas de cálculo, entre otros, lo cual aumenta el riesgo de pérdida, desorganización o incorrecto uso de los bienes.
- **Experiencia de usuario deficiente:** Al no existir un canal formalizado y automatizado para la gestión de recursos, los usuarios desconocen la disponibilidad real de los elementos hasta realizar consultas presenciales o directas. Esta falta de visibilidad y claridad en el proceso genera demoras y reduce la disposición de las personas para solicitar recursos.

Frente a esta realidad, se hace evidente la necesidad de diseñar e implementar un módulo de préstamos de recursos dentro de COMA, que permita automatizar los flujos de trabajo, centralizar la información y brindar transparencia, accesibilidad y confianza a toda la comunidad universitaria.

Con este nuevo módulo, la universidad podrá:

- Optimizar el uso de recursos, evitando tiempos muertos y mejorando la disponibilidad mediante un sistema de reservas ágil y con aprobaciones claras.
- Mejorar la experiencia de usuario, gracias a una interfaz sencilla, intuitiva y coherente con la plataforma COMA, lo que facilitará su rápida adopción.
- Fortalecer la gestión institucional, ya que se contará con datos centralizados, estadísticas confiables y reportes automáticos que servirán de apoyo para la planificación y la toma de decisiones.
- Garantizar seguridad y confiabilidad, con validaciones de integridad, controles de acceso según el rol y un seguimiento detallado de cada transacción realizada.

En definitiva, este proyecto no solo busca resolver una necesidad operativa inmediata, sino que también se proyecta como una oportunidad para seguir consolidando a COMA como un ecosistema digital integral, moderno y ajustado a las verdaderas dinámicas de la comunidad académica de la UIS. Con este módulo, la universidad no solo ahorrará tiempo y recursos, sino que también dará un paso firme hacia la transformación digital de sus procesos internos, fortaleciendo la confianza de sus estudiantes, docentes y administrativos en la institución.

## **1.3 Objetivos**

### ***1.3.1 Objetivo general***

Desarrollar e implementar los servicios funcionales del usuario propietario en el módulo de préstamos de la plataforma COMA, incluyendo el registro, edición, gestión y seguimiento de recursos, así como la administración de préstamos. Además, desarrollar e implementar algunos casos del actor administrador, necesarios para garantizar el correcto funcionamiento del sistema.

### ***1.3.2 Objetivos específicos***

- ✓ Diseñar y desarrollar las funcionalidades del actor propietario, incluyendo:
- ✓ Registro, edición, deshabilitación y administración de recursos.
- ✓ Aprobación o rechazo de solicitudes de préstamo.
- ✓ Seguimiento y control de los recursos en préstamo.
- ✓ Implementar el módulo en la plataforma COMA, asegurando su compatibilidad con los componentes del sistema y la usabilidad adecuada para la comunidad universitaria.

## **2. Marco de referencia**

En el desarrollo del módulo de gestión de préstamos para la plataforma COMA, se han identificado diversas herramientas y tecnologías que, por sus características y capacidades, resultan fundamentales para garantizar una implementación eficiente. A continuación, se describe cada una de ellas, destacando su funcionalidad y aplicabilidad dentro del proyecto:

### **2.1 Modelado UML**

Las metodologías basadas en diagramas UML (Unified Modeling Language) son especialmente útiles en la etapa de análisis y diseño del software, ya que proporcionan una visión detallada de los casos de uso, flujos de información y relaciones entre los actores y los sistemas involucrados.

## 2.2 Base de datos relacional

La estructura de datos en el proyecto COMA se organiza bajo un modelo relacional, que emplea tablas para estructurar la información. Las relaciones entre estas tablas, definidas a través de claves primarias y foráneas, permiten establecer vínculos lógicos entre diferentes conjuntos de datos, posibilitando consultas complejas y estructuradas.

Esta aproximación es idónea para escenarios donde se requiere integridad, consistencia y escalabilidad, ya que garantiza que los datos se mantengan organizados y confiables incluso a medida que el sistema crece.

Entre las ventajas del uso de bases de datos relacionales se incluyen:

- **Integridad referencial:** Garantiza la consistencia de los datos al evitar registros huérfanos o duplicados.
- **Escalabilidad:** Permite el crecimiento de la base de datos sin afectar de forma significativa el rendimiento del sistema.
- **Flexibilidad en las consultas:** Facilita el acceso a la información mediante el lenguaje SQL, permitiendo desde búsquedas simples hasta análisis avanzados.
- **Normalización de datos:** Reduce la redundancia y optimiza el almacenamiento al organizar la información en estructuras más eficientes.
- **Seguridad y control de acceso:** Ofrece mecanismos para gestionar permisos y roles de usuarios, asegurando la protección de la información sensible.

### 2.3 MySQL 8.3.0

El sistema de gestión de bases de datos seleccionado para este proyecto es MySQL en su versión 8.3.0. Este motor ofrece herramientas avanzadas para la administración de datos, asegurando consultas rápidas, seguras y escalables.

MySQL permite gestionar transacciones, índices, vistas y procedimientos almacenados, lo que facilita la optimización del rendimiento en un entorno con múltiples usuarios y datos dinámicos. Además, proporciona mecanismos de replicación y alta disponibilidad, esenciales para garantizar la continuidad del servicio en escenarios críticos.

Las bases de datos principales en el sistema son:

- **“Poseidón”**: Base de datos compartida entre las distintas escuelas de la comunidad COMA, utilizada para centralizar información común y mantener coherencia en los procesos globales.
- **“Diamante”**: Base de datos exclusiva para cada escuela, lo que permite una gestión personalizada de sus recursos, garantizando independencia y flexibilidad en la administración de datos particulares.

Adicionalmente, MySQL soporta el manejo eficiente de datos relacionales mediante operaciones CRUD (Create, Read, Update, Delete), lo que facilita la interacción con la aplicación. A esto se suma su compatibilidad con múltiples plataformas y lenguajes de programación, lo que lo convierte en una herramienta versátil y ampliamente utilizada en proyectos de gran escala.

### 2.4 Java

El lenguaje de programación Java se emplea para implementar la lógica del módulo. Este lenguaje, ampliamente utilizado en aplicaciones empresariales, es reconocido por su portabilidad, orientación a objetos y su capacidad de integrarse en entornos distribuidos.

En este proyecto, Java gestiona operaciones clave como la conexión a la base de datos, la ejecución de consultas CRUD (*Create, Read, Update, Delete*), el manejo de transacciones y la manipulación de datos que posteriormente se presentan en la interfaz web. Además, se apoya en servlets y JSP para la comunicación entre el servidor y el cliente, asegurando un flujo dinámico y seguro de la información.

- **Portabilidad:** Capacidad de ejecutarse en diferentes plataformas sin modificaciones, gracias a la Java Virtual Machine (JVM).
- **Robustez:** Mecanismos avanzados de gestión de errores y excepciones que aumentan la confiabilidad del sistema.
- **Comunidad activa:** Amplia cantidad de bibliotecas, frameworks y recursos de soporte que facilitan el desarrollo y mantenimiento.
- **Seguridad:** Proporciona herramientas y buenas prácticas para la protección de datos y control de accesos.
- **Escalabilidad:** Su arquitectura modular permite desarrollar sistemas que pueden crecer en funcionalidades y usuarios sin comprometer el rendimiento.

## 2.5 JSP (JavaServer Pages)

Se utiliza para la creación de las vistas dinámicas del sistema. Este enfoque permite separar el contenido HTML de la lógica del servidor, haciendo más sencillo el desarrollo, mantenimiento y escalabilidad de la aplicación.

En este proyecto, JSP interactúa directamente con el backend en Java, generando respuestas dinámicas según las solicitudes enviadas por los usuarios a través del navegador. De

esta forma, se asegura que la información mostrada en pantalla esté siempre actualizada y sincronizada con la base de datos.

Las ventajas de utilizar JSP incluyen:

- **Separación de preocupaciones:** Facilita el mantenimiento del código al distinguir entre la lógica del negocio y la presentación.
- **Integración con Java EE:** Permite aprovechar las capacidades del entorno empresarial de Java, incluyendo servlets, EJB y controladores de seguridad.
- **Soporte para expresiones y etiquetas personalizadas:** Simplifica el desarrollo de páginas web interactivas y modulares, reduciendo la cantidad de código repetido.
- **Compatibilidad con tecnologías web:** Se integra de forma sencilla con HTML, CSS, JavaScript y frameworks frontend, lo que permite construir interfaces más atractivas y funcionales.
- **Escalabilidad y modularidad:** Facilita la creación de componentes reutilizables que pueden integrarse en diferentes partes del sistema.

## 2.6 Apache Tomcat 9.0.88

El servidor de aplicaciones Apache Tomcat se utiliza como contenedor web para ejecutar la aplicación desarrollada. Este servidor es responsable de gestionar las solicitudes de los clientes, procesarlas y dirigir las a los componentes correspondientes del sistema, asegurando una comunicación eficiente entre la capa de presentación (JSP), la lógica del negocio (Java) y la base de datos (MySQL).

La versión 9.0.88 garantiza compatibilidad con las tecnologías empleadas en el proyecto, así como alta estabilidad y un rendimiento confiable en entornos de producción.

Las principales características de Apache Tomcat incluyen:

- **Compatibilidad con Java Servlet y JSP:** Facilita el despliegue de aplicaciones basadas en Java al proporcionar un contenedor confiable para la ejecución de estos componentes.
- **Eficiencia en el manejo de conexiones simultáneas:** Permite soportar entornos con alto tráfico de usuarios sin comprometer la estabilidad del sistema.
- **Herramientas de administración integradas:** Ofrece un panel de gestión para el monitoreo, configuración y ajuste del rendimiento en tiempo real.
- **Ligereza y simplicidad:** A diferencia de otros servidores de aplicaciones más complejos, Tomcat es ligero y fácil de configurar, lo que reduce los tiempos de despliegue.
- **Escalabilidad y extensibilidad:** Se adapta a sistemas en crecimiento y permite la integración con otras tecnologías del ecosistema Java EE o frameworks externos.
- **Comunidad activa y soporte continuo:** Al ser un proyecto de la Apache Software Foundation, cuenta con actualizaciones frecuentes y documentación extensa.

## 2.7 Html

HTML (HyperText Markup Language) define la estructura y el contenido de las páginas web del módulo. Es la base sobre la que se construye la interfaz, ya que organiza los elementos visuales y garantiza una navegación clara, coherente y accesible para los usuarios.

En este proyecto, HTML se combina con CSS, JavaScript, JSP y Bootstrap, lo que permite enriquecer la experiencia del usuario con diseños responsivos, interactividad y contenido dinámico generado desde el servidor.

Los elementos más relevantes incluyen:

- **Etiquetas semánticas:** Mejoran la accesibilidad y la comprensión del contenido tanto para los usuarios como para motores de búsqueda y lectores de pantalla.
- **Formularios interactivos:** Facilitan la entrada y validación de datos por parte de los usuarios, permitiendo el envío de información hacia el backend.
- **Estructuras jerárquicas:** Organizan el contenido en secciones claras, favoreciendo la navegación intuitiva.
- **Compatibilidad con estándares web:** Asegura que las páginas puedan visualizarse correctamente en distintos navegadores y dispositivos.
- **Integración con hojas de estilo y scripts:** Proporciona flexibilidad para separar la presentación (CSS) y el comportamiento (JavaScript) del contenido estructural.

## 2.8 Framework de estilos

El equipo de desarrollo emplea un framework de estilos propio del grupo Calumet, diseñado específicamente para mantener la coherencia visual en toda la plataforma COMA. Este conjunto de plantillas, guías y componentes estándar facilita la integración entre diferentes módulos y asegura una experiencia de usuario uniforme y profesional, independientemente del número de desarrolladores involucrados en el proyecto.

Las características clave del framework incluyen:

- **Consistencia visual:** Define lineamientos comunes para paletas de colores, tipografías, iconografía y espaciados, garantizando una identidad gráfica unificada.
- **Componentes reutilizables:** Incluye botones, formularios, tarjetas y menús estandarizados que simplifican el desarrollo y reducen la redundancia de código.

- **Escalabilidad:** Facilita la incorporación de nuevos módulos y funcionalidades sin comprometer la armonía visual del sistema.
- **Mantenibilidad:** Centraliza las reglas de estilo, lo que agiliza la actualización de la interfaz y evita inconsistencias entre secciones de la plataforma.

## 2.9 JavaScript

Es utilizado para enriquecer la interacción del usuario con el sistema. Este lenguaje de programación del lado del cliente permite manipular los elementos de la página en tiempo real, validar formularios, gestionar eventos y realizar solicitudes asíncronas al servidor, mejorando la experiencia de usuario al reducir tiempos de espera y evitar recargas completas de la página.

En este proyecto se emplea *vanilla JavaScript*, lo que significa que se trabaja directamente con las APIs nativas del navegador (sin librerías externas). Esto permite mantener el sistema más ligero, eficiente y fácil de mantener, aprovechando al máximo las capacidades estándar del lenguaje. Entre sus funcionalidades más importantes se incluyen:

- **Interactividad dinámica:** Responde de manera inmediata a las acciones del usuario sin necesidad de recargar la página.
- **Manipulación del DOM:** Permite modificar la estructura, estilo y contenido de la página de forma programática para reflejar cambios en tiempo real.
- **Gestión de eventos:** Controla interacciones como clics, envíos de formularios o cambios en campos de entrada.
- **Soporte para programación asíncrona:** A través de *fetch*, *promesas* y *async/await*, se realizan solicitudes al servidor sin interrumpir la experiencia del usuario.

- **Integración con APIs:** Facilita la comunicación con servicios externos y el intercambio de datos en tiempo real.

## 2.10 Git y GitHub

Git es el sistema de control de versiones empleado para gestionar los cambios en el código fuente del proyecto. Su uso asegura la integridad del código, permite llevar un historial detallado de modificaciones y facilita la colaboración entre desarrolladores, incluso cuando trabajan en paralelo en distintas funcionalidades.

GitHub se utiliza como repositorio remoto y entorno centralizado de colaboración. Además de almacenar el código, proporciona herramientas para la gestión de ramas, la revisión de cambios y la integración continua, lo que refuerza la calidad y estabilidad del software en cada entrega. Las buenas prácticas adoptadas incluyen:

- **Ramas de desarrollo independientes:** Facilitan el trabajo paralelo en nuevas funcionalidades o correcciones, evitando conflictos en la rama principal.
- **Revisiones de código (Pull Requests):** Mejoran la calidad del software mediante la detección temprana de errores y el fomento de buenas prácticas de programación.
- **Integración continua (CI):** Automatiza pruebas y despliegues, garantizando que las nuevas funcionalidades se incorporen sin afectar la estabilidad del sistema.
- **Documentación y versionado semántico:** Asegura claridad en los cambios realizados y facilita el seguimiento de versiones estables.
- **Gestión de incidencias y proyectos:** Uso de *Issues* y *Projects* en GitHub para organizar tareas, asignar responsables y dar seguimiento al avance del desarrollo.

## 2.11 CSS

(Cascading Style Sheets) es el lenguaje utilizado para definir la presentación visual de las páginas web del sistema. Permite separar el contenido (HTML) de su diseño, garantizando interfaces atractivas, consistentes y responsivas para los usuarios.

En este proyecto, CSS se emplea para personalizar la apariencia del módulo y se integra con el framework de estilos propio de Calumet, asegurando uniformidad en toda la plataforma COMA. Las características más relevantes incluyen:

- **Separación de contenido y diseño:** Mantiene el HTML enfocado en la estructura y el CSS en la apariencia.
- **Diseño responsivo:** Uso de reglas y unidades flexibles que permiten adaptar la interfaz a diferentes dispositivos y tamaños de pantalla.
- **Selectores y pseudoclasas:** Facilitan la aplicación de estilos específicos y comportamientos dinámicos según la interacción del usuario.
- **Animaciones y transiciones:** Mejoran la experiencia visual con efectos fluidos y atractivos.
- **Reutilización de estilos:** Permite definir clases comunes que se aplican en múltiples elementos, reduciendo la redundancia y mejorando la mantenibilidad.
- **Compatibilidad con estándares web:** Asegura una visualización coherente en los principales navegadores.

## 2.12 Servlet

Los Servlets son componentes del lado del servidor desarrollados en Java que permiten gestionar las solicitudes y respuestas dentro de una aplicación web. Su principal función es actuar

como intermediarios entre el cliente (generalmente un navegador) y la lógica de negocio del sistema, procesando peticiones HTTP y generando respuestas dinámicas.

En este proyecto, los Servlets se utilizan para manejar la comunicación entre el cliente y el servidor, garantizando la correcta interacción con la base de datos y los recursos del sistema. Gracias a ellos, las operaciones como registrar, actualizar, consultar o eliminar información se realizan de manera estructurada y segura.

Entre sus características y ventajas más relevantes destacan:

- **Procesamiento de peticiones HTTP:** Manejan métodos como GET y POST para recibir datos desde formularios, botones o llamadas AJAX.
- **Generación de contenido dinámico:** Crean respuestas personalizadas (HTML, JSON, XML, etc.) según las necesidades del usuario.
- **Integración con JSP y JavaBeans:** Trabajan en conjunto con otras tecnologías Java para separar la lógica de negocio de la presentación.
- **Escalabilidad y eficiencia:** Los Servlets se cargan una sola vez en memoria, lo que mejora el rendimiento frente a tecnologías más antiguas como CGI.
- **Seguridad:** Ofrecen soporte para autenticación, control de sesiones y validación de datos, protegiendo la aplicación frente a accesos indebidos.

En este proyecto, los Servlets desempeñan un papel central, ya que son los encargados de recibir las solicitudes desde la interfaz web (JavaScript/CSS/HTML), ejecutar la lógica de negocio y devolver una respuesta adecuada, asegurando una experiencia fluida para el usuario final.

### 3. Metodología

Para garantizar el desarrollo exitoso del módulo de préstamos de recursos en la plataforma COMA, se adoptó una metodología híbrida, la cual combina elementos de enfoques ágiles, particularmente Scrum, con prácticas más tradicionales de ingeniería de software. Esta mezcla permitió mantener un equilibrio adecuado entre la flexibilidad y la rapidez de respuesta ante cambios en los requisitos, junto con el rigor y la trazabilidad necesarios para asegurar la calidad en el diseño, la validación y la documentación del sistema.

La elección de este enfoque metodológico respondió a la naturaleza del proyecto: un desarrollo académico con requerimientos cambiantes y la necesidad de entregar resultados funcionales en plazos cortos, sin sacrificar la organización ni la claridad del proceso.

### 3.1 Enfoque general

Se utilizó un ciclo de vida iterativo e incremental, lo que significó que el módulo se fue construyendo en pequeñas entregas funcionales y cada una de ellas fue revisada y mejorada antes de continuar con la siguiente. Esto permitió:

- Reducir riesgos, ya que los errores podían detectarse de manera temprana.
- Validar continuamente los avances con el director del proyecto.
- Ajustar el diseño según las observaciones de los usuarios de prueba. Las fases principales fueron

**1. Análisis y planificación:** identificación de requerimientos funcionales y no funcionales, definición de casos de uso y estimación de esfuerzo.

**2. Diseño y modelado:** elaboración de diagramas UML que representaran la lógica del sistema, la interacción entre actores y la estructura de la base de datos.

**3. Desarrollo iterativo:** construcción progresiva de funcionalidades en pequeños incrementos.

4. **Pruebas y validación:** aplicación de pruebas funcionales, de integración y de usabilidad en cada iteración.
5. **Despliegue y monitoreo:** liberación en el entorno de producción y seguimiento de su comportamiento real.

### 3.2 Metodología de desarrollo: Scrum

Se eligió Scrum como marco de trabajo ágil porque se ajusta muy bien a proyectos de software que requieren retroalimentación constante y capacidad de adaptación. Las razones principales fueron:

- **Entregas incrementales y funcionales:** cada sprint generaba una parte del sistema completamente operativa.
- **Adaptación a cambios:** los requisitos del módulo podían ajustarse en cada iteración según observaciones o pruebas.
- **Colaboración continua:** se fomentó la comunicación entre los integrantes del equipo y el director del proyecto.

#### 3.2.1 Ceremonias y artefactos implementados

- **Sprints:** iteraciones de 2 a 3 semanas de duración.
- **Product Backlog:** lista priorizada de funcionalidades, mantenida y actualizada según la evolución del proyecto.
- **Sprint Planning:** reuniones de planificación al inicio de cada sprint para asignar tareas y definir objetivos.
- **Daily Stand-ups:** encuentros breves para reportar avances, dificultades y coordinar al equipo.

- **Sprint Review y Retrospective:** espacios de revisión para mostrar resultados, recoger retroalimentación y proponer mejoras en la dinámica de trabajo.

### 3.3 Herramientas de modelado

Para el diseño y documentación del sistema se utilizó UML (Lenguaje Unificado de Modelado), estándar ampliamente aceptado en ingeniería de software.

Se generaron los siguientes diagramas:

- **Casos de uso:** capturaron los requisitos funcionales desde la perspectiva de los actores (propietario, administrador y solicitante).
- **Diagramas de clases:** definieron la estructura de la base de datos, atributos y relaciones entre entidades.
- **Diagramas de actividades:** detallaron flujos de procesos clave como el préstamo y creación de los recursos

Esto permitió tener una visión clara y compartida de la arquitectura del módulo antes de su construcción.

### 3.4 Técnicas de desarrollo

Durante la implementación se adoptaron las siguientes prácticas:

- **Desarrollo por componentes:** el sistema fue dividido en módulos independientes que se integraron de manera progresiva. Esto facilitó su mantenimiento y escalabilidad.
- **Prototipado evolutivo:** se construyeron prototipos visuales y funcionales de las interfaces, los cuales fueron validados con usuarios clave antes de iniciar el desarrollo completo.

### 3.5 Gestión de configuración y control de versiones

Se utilizó Git como sistema de control de versiones y GitHub como plataforma de gestión del repositorio. El flujo de trabajo fue el siguiente:

1. Cada desarrollador trabajaba sobre una rama local derivada de module/prestamo-de-recurso.
2. Los cambios se subían a la rama remota correspondiente.
3. Una vez validados, se realizaba un merge a la rama de pruebas.
4. Solo tras superar las pruebas se integraban los cambios a la rama de producción.

Esto aseguró un desarrollo ordenado, redujo los riesgos de errores en producción y permitió mantener un historial detallado de los cambios.

### 3.6 Pruebas

La estrategia de pruebas fue iterativa e incremental. Cada nueva funcionalidad desarrollada fue validada de inmediato para detectar errores rápidamente. Tipos de pruebas realizadas:

- **Pruebas funcionales:** verificaron que cada módulo cumpliera con los requisitos especificados.
- **Pruebas de integración:** confirmaron la correcta interacción entre los nuevos componentes y el resto del sistema.
- **Pruebas de validación:** se realizaron sobre la rama de pruebas, simulando el flujo de préstamo de recursos desde la perspectiva de diferentes actores.
- **Pruebas de usabilidad interna:** el propio equipo adoptó los roles de propietario y solicitante para evaluar la facilidad de uso y la coherencia de la interfaz.

Gracias a esta estrategia, se garantizó que los errores fueran detectados y corregidos oportunamente antes de llegar a producción.

### 3.7 Despliegue

El despliegue siguió el flujo de ramas establecido en el repositorio:

- El desarrollo de nuevas funcionalidades se hacía en ramas locales.
- Luego se integraban en la rama de pruebas para validar en un entorno controlado.
- Finalmente, tras superar validaciones, se trasladaban a la rama de producción.

Este proceso permitió un control estricto de versiones, minimizó riesgos en la liberación y aseguró que solo funcionalidades estables llegarán a los usuarios finales.

### 3.8 Documentación

La documentación se gestionó en dos niveles:

- 1. Documentación técnica:** se incluyó directamente en el código fuente mediante comentarios y descripciones, lo que facilita el mantenimiento por parte de futuros desarrolladores.
- 2. Manual de usuario:** dirigido a propietarios y administradores, elaborado en un lenguaje práctico y accesible para explicar las principales funcionalidades del módulo de préstamos.

La documentación se construyó en paralelo al desarrollo, garantizando su actualización constante.

### 3.9 Actividades de análisis y planificación

Antes de iniciar el desarrollo, se realizó un análisis exhaustivo de los requerimientos funcionales y no funcionales del módulo. Este proceso incluyó:

- Reuniones con los actores principales (propietarios, administradores y solicitantes) para recopilar necesidades específicas y validar los objetivos del sistema.

- Elaboración de diagramas de procesos, casos de uso y flujos de datos, con el fin de estructurar la lógica del sistema y definir sus interacciones clave.
- Priorización de funcionalidades de acuerdo con el impacto, la criticidad y el valor agregado para los usuarios, estableciendo un orden de desarrollo progresivo.
- Identificación de restricciones y riesgos, considerando aspectos técnicos, de seguridad y escalabilidad, lo cual permitió plantear estrategias de mitigación tempranas.

Este análisis permitió definir una línea base del proyecto y sirvió como fundamento para el plan de desarrollo, asegurando que cada avance respondiera a las necesidades reales de los usuarios y a los objetivos del proyecto de grado.

### **3.10 Desarrollo de componentes utilizando metodología ágil**

Para la construcción del módulo de préstamos de recursos en la plataforma COMA, se adoptó la metodología ágil Scrum, un marco de trabajo ampliamente utilizado en proyectos de desarrollo de software que requieren flexibilidad y retroalimentación constante. Esta elección metodológica permitió organizar el trabajo en ciclos cortos e iterativos, entregando incrementos funcionales al final de cada fase y asegurando que los avances se ajustarán a las expectativas de los actores involucrados.

La aplicación de Scrum resultó especialmente valiosa en este contexto, ya que el proyecto requería no solo cumplir con requisitos técnicos, sino también garantizar que las funcionalidades respondieran de manera efectiva a las dinámicas académicas de la comunidad.

#### **3.10 Prácticas implementadas**

- **Sprints:** Cada sprint tuvo un objetivo claro y alcanzable, lo que permitió organizar el trabajo en entregables manejables. Por ejemplo, en un sprint se abordó el desarrollo del registro y

edición de recursos, en otro la implementación del chat de solicitudes y en un tercero la generación de reportes para los administradores. Al concluir cada iteración, se obtenía una versión parcial pero funcional del sistema, lista para ser validada y retroalimentada por los usuarios clave.

- **Revisiones constantes internas:** Dentro de cada sprint, el equipo realizó sesiones de validación técnica y funcional. Estas revisiones permitieron comprobar que los componentes recién implementados cumplían con los requisitos establecidos, identificar ajustes necesarios y proponer mejoras antes de avanzar al siguiente ciclo. Este proceso evitó la acumulación de errores y permitió mantener un producto en constante evolución y mejora.
- **Gestión de cambios:** Una de las fortalezas de Scrum es su capacidad para adaptarse a cambios. Durante el desarrollo del módulo, se recibieron observaciones y sugerencias tanto del director como de los mismos desarrolladores al usar el sistema en pruebas internas. Estos cambios fueron documentados, priorizados y atendidos en las iteraciones siguientes, asegurando que el sistema evolucionara alineado con las necesidades reales de la comunidad académica COMA.
- **Ceremonias ágiles:** Se llevaron a cabo las principales ceremonias definidas en Scrum:
  - **Planificación de sprint:** donde se definieron objetivos concretos y tareas específicas.
  - **Daily meetings:** reuniones breves para compartir avances, dificultades y coordinar al equipo (cuando fue posible realizarse de manera continua).
  - **Retrospectivas:** al finalizar los sprints, se reflexionó sobre lo que funcionó bien y lo que debía mejorar, fortaleciendo así la dinámica de trabajo.

### 3.12 Asignación de roles en Scrum

En el marco de Scrum, se asignaron roles específicos que facilitaron la organización del equipo y la correcta aplicación de la metodología. Aunque el grupo de trabajo fue reducido, cada integrante asumió responsabilidades claras y, al mismo tiempo, colaboró en las diferentes áreas del desarrollo.

- Felipe Andrés Cepeda Ortiz – Scrum Master y miembro del Development Team: Felipe asumió el rol de Scrum Master, encargado de facilitar las ceremonias ágiles, supervisar que se respetaran los principios de Scrum y, sobre todo, eliminar los posibles impedimentos que surgieran en el desarrollo. Este rol fue clave para mantener el orden metodológico y garantizar que el equipo trabajara bajo un enfoque ágil y disciplinado. Además de sus funciones como Scrum Master, Felipe participó activamente en el Development Team, contribuyendo en el diseño de la arquitectura del módulo, el desarrollo de las funcionalidades principales y la integración del código con la plataforma COMA. Su doble papel permitió un equilibrio entre la gestión del proceso ágil y la implementación técnica del proyecto.
- Jeicob Gilmar Restrepo Gómez – Product Owner y miembro del Development Team: Jeicob desempeñó el rol de Product Owner, siendo el responsable de mantener la visión del producto, priorizar las funcionalidades dentro del Product Backlog y asegurarse de que los entregables respondieran a las necesidades de los usuarios finales. Su participación fue fundamental para garantizar la coherencia entre los objetivos planteados y las funcionalidades desarrolladas, manteniendo siempre la perspectiva de valor hacia los actores del sistema. De manera complementaria, Jeicob también formó parte del Development Team, colaborando

en el análisis de requerimientos, el diseño de modelos UML y el desarrollo de componentes clave como la gestión de solicitudes y el chat de comunicación. Su rol dual fortaleció la conexión entre la visión estratégica y la implementación técnica del módulo.

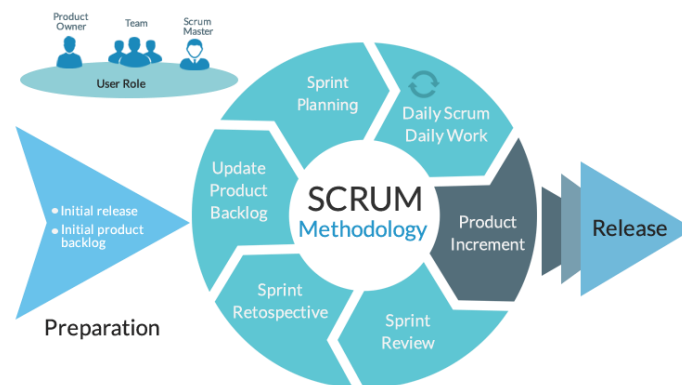
### 3.13 Reflexión final sobre el uso de Scrum

La aplicación de Scrum permitió que el desarrollo del módulo fuera un proceso ágil, adaptable y centrado en el usuario. Gracias a la organización en sprints, las constantes revisiones internas y la clara asignación de roles, el equipo logró avanzar con orden y eficiencia, entregando un producto funcional y ajustado a las necesidades de la comunidad COMA.

En conclusión, la metodología ágil no solo facilitó el trabajo colaborativo, sino que también aportó valor al producto final, al permitir una construcción iterativa, incremental y con capacidad de responder a cambios sin comprometer la calidad del sistema

#### Figura 1

*Ciclo y actores de la metodología ágil SCRUM*



Nota. Imagen tomada de Drew. (2019). SCRUM Methodology.

<https://blog.wearedrew.co/productividad/-ventajas-y-desventajas-de-la-metodologia-scrum>

### 3.14 Implementación de pruebas funcionales y de usabilidad

El proceso de pruebas será esencial para garantizar que el módulo funcione correctamente, cumpla con los requerimientos y ofrezca una experiencia satisfactoria a los usuarios. Las actividades de pruebas incluyeron:

- **Pruebas de integración:** Validación de la interacción entre las funcionalidades del módulo y la plataforma COMA, comprobando la correcta comunicación entre los distintos servicios y la base de datos.
- **Pruebas de usabilidad internas:** Estas pruebas se realizaron de manera interna por el equipo de desarrollo, ya que la participación de usuarios finales únicamente era posible durante el despliegue real del módulo. En ellas, los integrantes del equipo simulaban distintos escenarios de uso para evaluar la facilidad de navegación, la coherencia de la interfaz y la experiencia general de interacción con el sistema.
- **Pruebas de rendimiento y tiempos de respuesta:** Aunque no se realizaron pruebas de estrés a gran escala, el equipo llevó a cabo evaluaciones orientadas a medir los tiempos de carga y la eficiencia en la ejecución de procesos clave. Se verificó el desempeño en escenarios como la apertura de páginas, la creación y edición de recursos, el trámite completo de solicitudes (registro, edición, aceptación y finalización), así como la consulta de historiales. Estas pruebas permitieron identificar posibles cuellos de botella y garantizar una experiencia fluida para los usuarios durante las operaciones más frecuentes.
- **Pruebas de seguridad básicas:** Revisión de aspectos como el control de sesiones, validación de entradas para evitar inyecciones y protección de datos sensibles.

- **Pruebas de aceptación:** Comparación de los resultados obtenidos con los requerimientos funcionales previamente definidos, asegurando la trazabilidad entre lo solicitado y lo entregado.

### 3.15 Despliegue en la base de pruebas

El despliegue en la base de pruebas se realizó en un entorno aislado que replicaba fielmente el entorno real de la plataforma COMA. Esta fase resultó fundamental para verificar la estabilidad del módulo, corregir errores y realizar ajustes de usabilidad antes de dar paso al despliegue en producción.

Los principales objetivos de este despliegue fueron:

- **Instalación en entorno controlado:** Implementación del módulo en un servidor de pruebas que permitiera validar su correcto funcionamiento sin comprometer el entorno productivo.
- **Simulación de escenarios reales:** Aunque los usuarios finales aún no tenían acceso directo, el equipo desarrollador asumió el rol de usuario para ejecutar tareas cotidianas como crear almacenes, registrar recursos, editar información, gestionar solicitudes y consultar historiales.
- **Corrección de errores:** Se atendieron los problemas detectados en el proceso, ajustando la lógica de negocio, la organización de rutas y el flujo general del sistema.

Durante esta etapa, se presentaron múltiples hallazgos y mejoras. Inicialmente, surgieron inconvenientes relacionados con el cambio de rutas de PascalCase a kebab-case, lo que ocasiona que algunas funciones no fueran reconocidas correctamente. Estos problemas se solucionaron mediante una reorganización sistemática de las rutas y una mejor documentación del código, garantizando coherencia en toda la estructura del proyecto.

Posteriormente, las mejoras se orientaron hacia la experiencia de uso y la comprensión del sistema. Para ello, el equipo elaboró manuales de funciones y guías de uso, que facilitaron la futura

incorporación de usuarios reales, permitiendo que ellos mismos pudieran registrar recursos, gestionarlos y realizar solicitudes a otros miembros de la comunidad. Estos documentos fueron clave para asegurar que, una vez en producción, los usuarios contarán con un apoyo claro y práctico para aprovechar todas las funcionalidades del módulo.

En cuanto a funcionalidades específicas, se introdujeron cambios significativos:

- **Gestión de recursos con identificadores únicos:** Se reemplazaron los ID genéricos por códigos individuales que permitieran distinguir con claridad cada objeto (por ejemplo, 10 protoboards con numeración específica), garantizando trazabilidad y un control más detallado.
- **Campo JSON para listados:** Se creó un mecanismo flexible para almacenar información estructurada de los recursos, facilitando la asociación de atributos particulares y mejorando la organización de datos.
- **Calendario para propietarios:** Se añadió una vista que permitía a los propietarios conocer la disponibilidad de sus recursos, los períodos de préstamo y el orden en que debían atenderse las solicitudes.
- **Recursos sin bodega:** Para evitar que los recursos quedaran en estado intermedio, se implementó una pestaña que mostraba aquellos que aún no estaban asignados a un almacén, brindando una opción para administrarlos y organizarlos posteriormente.
- **Normalización de caracteres:** Se adoptó la codificación UTF-8 para garantizar el correcto manejo de tildes y caracteres especiales, evitando inconsistencias en la visualización de información.

En fases más avanzadas, se incorporaron dos mejoras clave que respondieron directamente a necesidades funcionales:

- **Inclusión de imágenes en los recursos:** Se habilitó la posibilidad de que los propietarios subieran fotografías de cada recurso, mostrando su estado físico y características. Esto no solo aportó transparencia a la comunidad, sino que también generó confianza al momento de solicitar préstamos, al permitir visualizar previamente el objeto.

### **3.16 Optimización de la calificación y asignaciones:**

Inicialmente, el sistema permitía calificar y recalificar a un solicitante indefinidamente, lo que distorsionaba su puntaje acumulado. Se ajustó esta lógica para que la calificación quedara ligada a una solicitud específica, registrándose en el historial y sumando de manera justa al promedio del usuario evaluado. Además, se mejoró el proceso de asignación de recursos, garantizando que el prestatario pudiera marcar el recurso como devuelto únicamente cuando realmente lo recibiera de vuelta, asegurando así mayor control sobre el ciclo de préstamo.

En conclusión, la fase de despliegue en la base de pruebas fue mucho más que una validación técnica: representó un espacio para pulir la experiencia de usuario, reforzar la lógica del sistema y preparar tanto la plataforma como a los futuros usuarios. Gracias a estas simulaciones internas y a las mejoras implementadas, el módulo alcanzó un nivel de madurez suficiente para pasar a producción, reduciendo riesgos y consolidando un flujo más estable, intuitivo y transparente.

De esta manera, el sistema quedó listo para su despliegue en producción.

### **3.17 Despliegue real**

El despliegue en producción se llevó a cabo una vez culminada satisfactoriamente la fase de pruebas, garantizando que todas las funcionalidades del módulo hubieran sido validadas, optimizadas y alineadas con los requerimientos de la plataforma COMA. Este despliegue marcó

la transición definitiva del módulo hacia un entorno real, habilitando su uso por parte de los usuarios finales de manera controlada y segura.

Gracias al proceso previo de pruebas, el entorno productivo se encontró libre de errores críticos, con una arquitectura estable y una experiencia de usuario claramente definida. El sistema fue implementado en la infraestructura oficial de la plataforma, asegurando continuidad operativa y compatibilidad con los demás módulos existentes.

### 3.18 Objetivos del despliegue en producción

- **Implementación en entorno real:** El módulo fue instalado en el servidor productivo de la plataforma COMA, integrándose correctamente con la base de datos principal y los servicios existentes, sin afectar la operación de los sistemas ya en funcionamiento.
- **Habilitación para usuarios finales:** A diferencia del entorno de pruebas, en esta fase el sistema quedó disponible para toda la comunidad de usuarios, permitiendo que propietarios y solicitantes interactuarán directamente con el módulo para registrar recursos, gestionar solicitudes, realizar préstamos y consultar historiales.
- **Validación operativa continua:** Aunque el sistema ya había sido probado internamente, durante el despliegue en producción se realizó un monitoreo activo del comportamiento del módulo, asegurando que las operaciones reales se ejecutarán de forma fluida y sin inconsistencias.
- **Estabilidad y mejoras consolidadas:** Durante esta fase no se presentaron inconvenientes relacionados con la estructura del sistema, ya que los problemas detectados en el entorno de pruebas —como la normalización de rutas, la codificación de caracteres y la coherencia en la navegación— fueron completamente resueltos antes del despliegue definitivo. Esto permitió

que todas las funcionalidades estuvieran correctamente reconocidas y accesibles desde el primer momento. Se elaboraron manuales de uso con el fin de que los usuarios puedan adaptarse de manera rápida al módulo, facilitando así su correcta utilización y mejorando su usabilidad.

### **Funcionalidades implementadas en producción**

El módulo fue desplegado con todas las mejoras y ajustes previamente validados, entre las que se destacan:

- **Gestión avanzada de recursos:** Cada recurso cuenta con identificadores únicos, lo que permite una trazabilidad precisa, un control detallado del inventario y una gestión clara de préstamos y devoluciones.
- **Estructuración flexible de datos:** Se mantuvo el uso de campos JSON para almacenar información específica de los recursos, permitiendo una administración más ordenada y adaptable a diferentes tipos de objetos.
- **Control de disponibilidad mediante calendario:** Los propietarios pueden visualizar la disponibilidad de sus recursos, los periodos de préstamo activos y el orden de atención de las solicitudes, facilitando una gestión más eficiente.
- **Administración de recursos no asignados:** El sistema incluye una vista dedicada para los recursos que aún no pertenecen a un almacén, evitando estados inconsistentes y permitiendo su organización posterior.
- **Soporte completo para caracteres especiales:** La implementación de codificación UTF-8 aseguró una visualización correcta de la información, evitando errores en tildes, nombres y descripciones.

- **Incorporación de imágenes:** Los propietarios pueden adjuntar fotografías a cada recurso, permitiendo a los solicitantes conocer su estado físico antes de realizar una solicitud, fortaleciendo la transparencia y la confianza dentro de la comunidad.
- **Calificaciones y asignaciones optimizadas:** El sistema de calificación quedó ligado a solicitudes específicas, evitando manipulaciones del puntaje y garantizando un promedio justo y representativo. Asimismo, el flujo de asignación y devolución de recursos se encuentra correctamente controlado, asegurando que cada préstamo cumpla su ciclo completo.

### 3.19 Construcción y validación de prototipos

El despliegue real del módulo no se limita únicamente a ponerlo en funcionamiento dentro de la plataforma COMA, sino que representa un proceso estratégico que busca garantizar su correcto desempeño, estabilidad y aceptación por parte de la comunidad universitaria. Para ello, se plantea una ruta organizada de implementación que permita minimizar riesgos, asegurar la calidad del software y, al mismo tiempo, garantizar una transición fluida desde la fase de pruebas hasta el uso cotidiano en el entorno académico.

Los pasos contemplados son los siguientes:

- **Preparación del entorno de producción:** antes de llevar el módulo a la plataforma principal, es indispensable realizar la configuración y optimización del servidor de producción. Esto incluye ajustes en la infraestructura tecnológica, instalación de dependencias necesarias, validación de la seguridad del entorno y pruebas de carga iniciales que permitan asegurar que el sistema pueda soportar el tráfico y el volumen de solicitudes reales sin comprometer el rendimiento.

- **Migración de datos:** en caso de que durante las fases de prueba se haya generado información relevante (como registros de recursos, solicitudes de prueba o configuraciones iniciales), se evaluará la pertinencia de migrar dichos datos hacia la base de datos de producción. Este proceso debe realizarse con cuidado para evitar duplicidades o inconsistencias, garantizando siempre la integridad de la información.
- **Despliegue del módulo en producción:** una vez asegurada la preparación del entorno y la migración de datos, se procederá con la implementación de la versión final del módulo en la plataforma COMA. Esta versión será accesible para todos los usuarios de la comunidad universitaria, por lo que debe incluir las funcionalidades validadas en los prototipos, las mejoras derivadas de las pruebas piloto y las configuraciones finales que garanticen una experiencia de usuario estable y confiable.
- **Monitoreo post-despliegue:** el despliegue no culmina con la puesta en marcha del sistema; por el contrario, se inicia una fase crítica de supervisión en tiempo real. Durante este periodo, se monitorea de manera continua el comportamiento del módulo, verificando la correcta ejecución de procesos como la solicitud de recursos, aprobación, devolución y generación de reportes. Este monitoreo permitirá identificar rápidamente errores o situaciones no previstas, de manera que puedan corregirse de forma oportuna sin afectar significativamente a los usuarios.

En conjunto, estos pasos aseguran que el nuevo módulo de préstamos de recursos no solo sea funcional en un entorno controlado de pruebas, sino que también pueda integrarse de manera efectiva y segura a la plataforma COMA en su uso cotidiano. Además, este enfoque sistemático permitirá ganar la confianza de la comunidad académica, al demostrar que el proceso de

construcción, validación y despliegue del sistema ha sido riguroso, transparente y orientado a la mejora de la experiencia universitaria.

#### **4. Desarrollo del proyecto**

El desarrollo del módulo de préstamos de recursos para la plataforma COMA se llevó a cabo mediante un proceso estructurado e iterativo, alineado con la metodología Scrum y los objetivos planteados por el grupo CALUMET. Este trabajo no solo abarcó el diseño técnico y la implementación de funcionalidades, sino también la validación constante con escenarios reales de uso y la preparación de manuales que garantizaran la apropiación de la herramienta por parte de los usuarios finales. A continuación, se detallan las actividades realizadas en cada fase:

##### **4.1 Análisis y diseño detallado**

Antes de iniciar la codificación, se realizó un análisis exhaustivo de los requerimientos funcionales y no funcionales del sistema. Este proceso incluyó:

###### ***4.1.1 Identificación de actores y casos de uso***

Se definieron los roles principales (Propietario, Solicitante, Administrador) y sus interacciones con el sistema mediante diagramas de casos de uso UML, lo que permitió detectar con claridad los escenarios de préstamo, devolución, aprobación y rechazo.

###### **4.1.1.1 Casos de uso del actor propietario.**

#### **Figura 2**

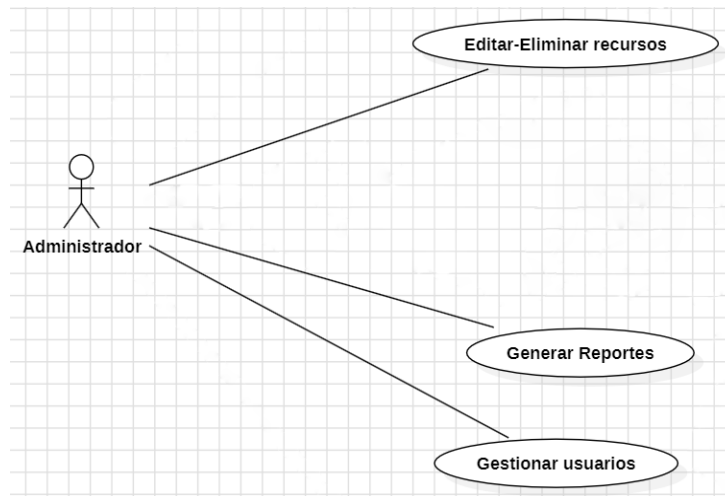
*Modelo de casos de uso – Actor propietario*



**4.1.1.2 Casos de uso del actor administrador**

**Figura 3**

*Modelo de casos de uso – Actor administrador*



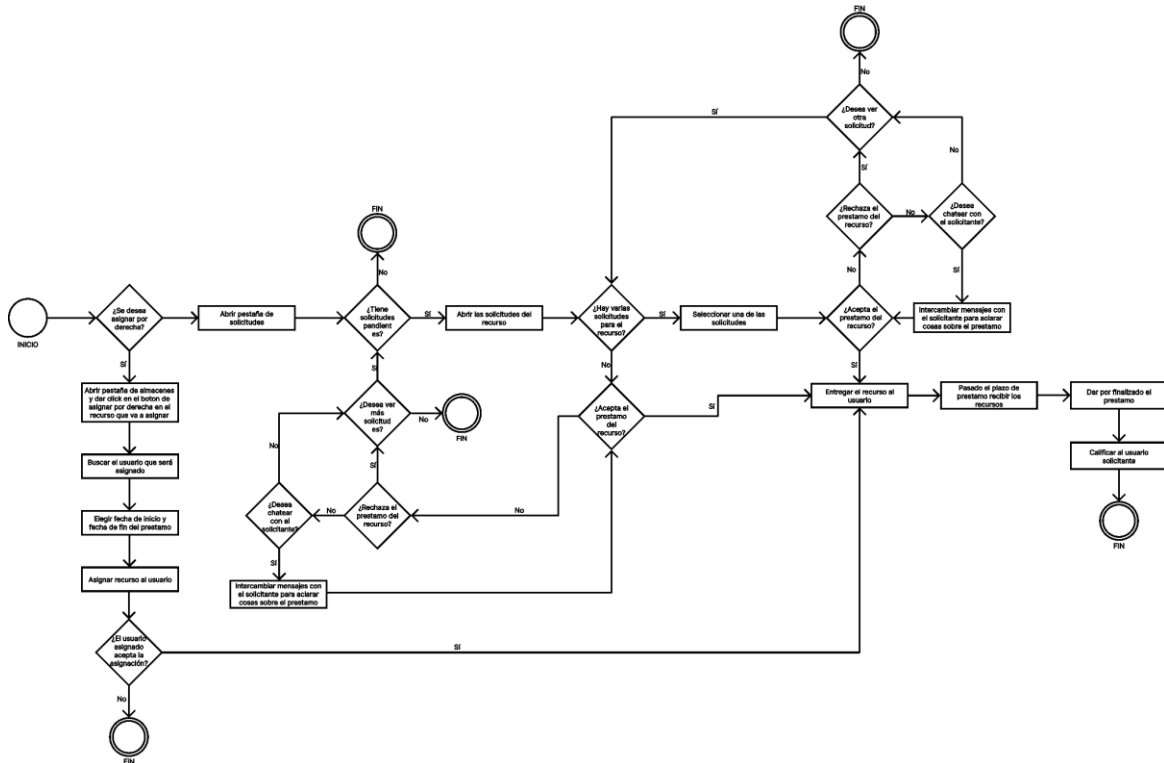
**4.1.2 Diagramas de actividades**

Modelan los flujos de trabajo y procesos del módulo.

4.1.2.1 Proceso de préstamo de un recurso

Figura 4.

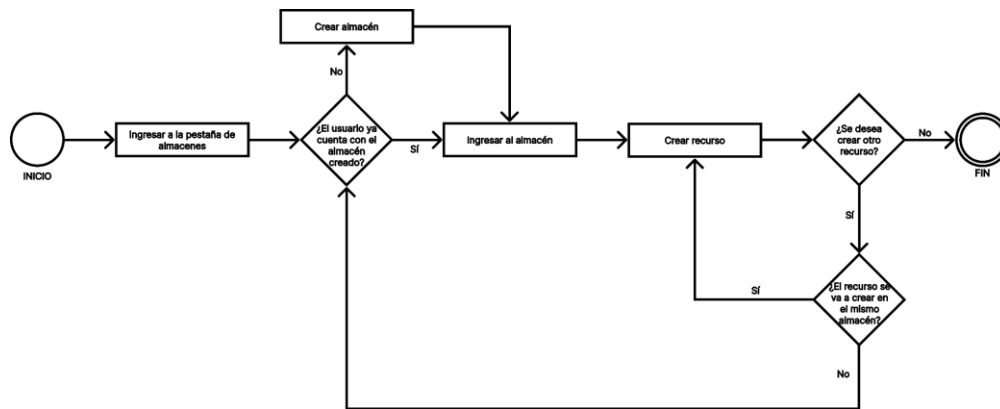
Diagrama de actividades – Proceso de préstamo



4.1.2.2 Proceso de creación de recurso

Figura 5.

Diagrama de actividades – Proceso de creación de recurso



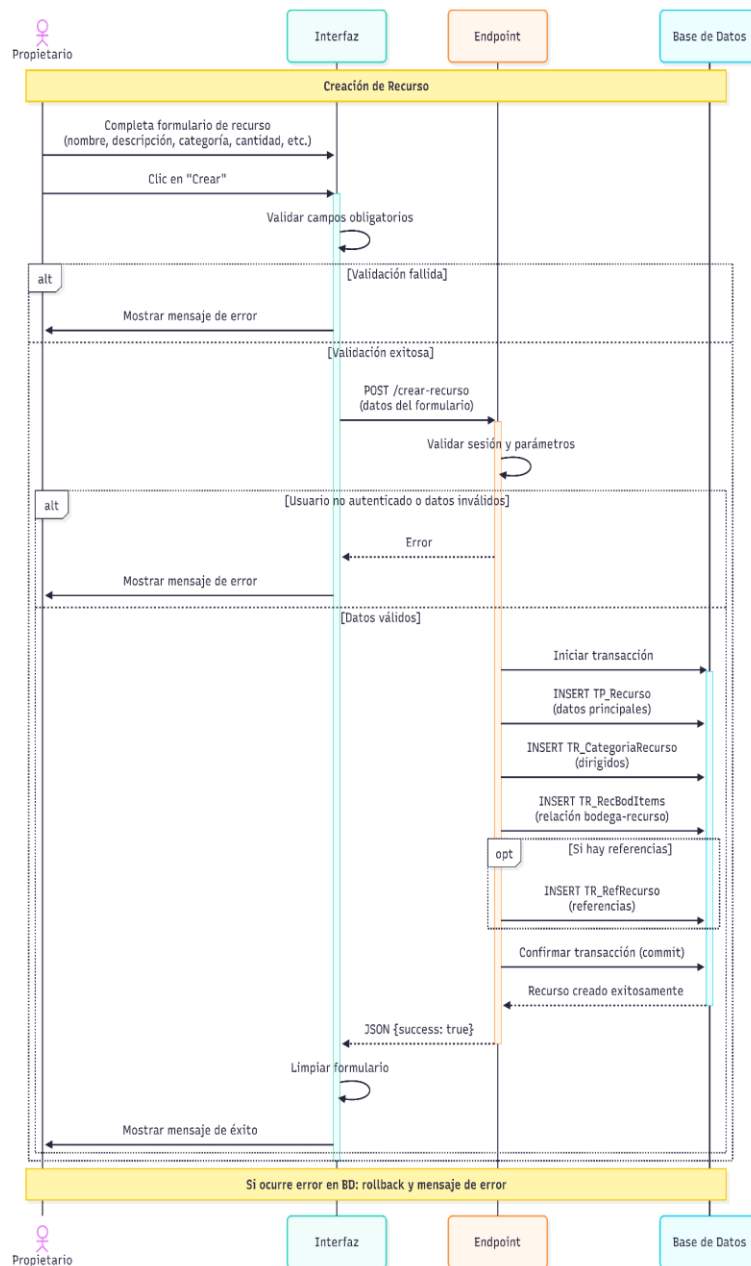
4.1.3 Diagramas de secuencia

El Modelado de Flujos de Trabajo y Procesos del Módulo.

4.1.3.1 Diagrama de secuencia de creación de un recurso

Figura 6

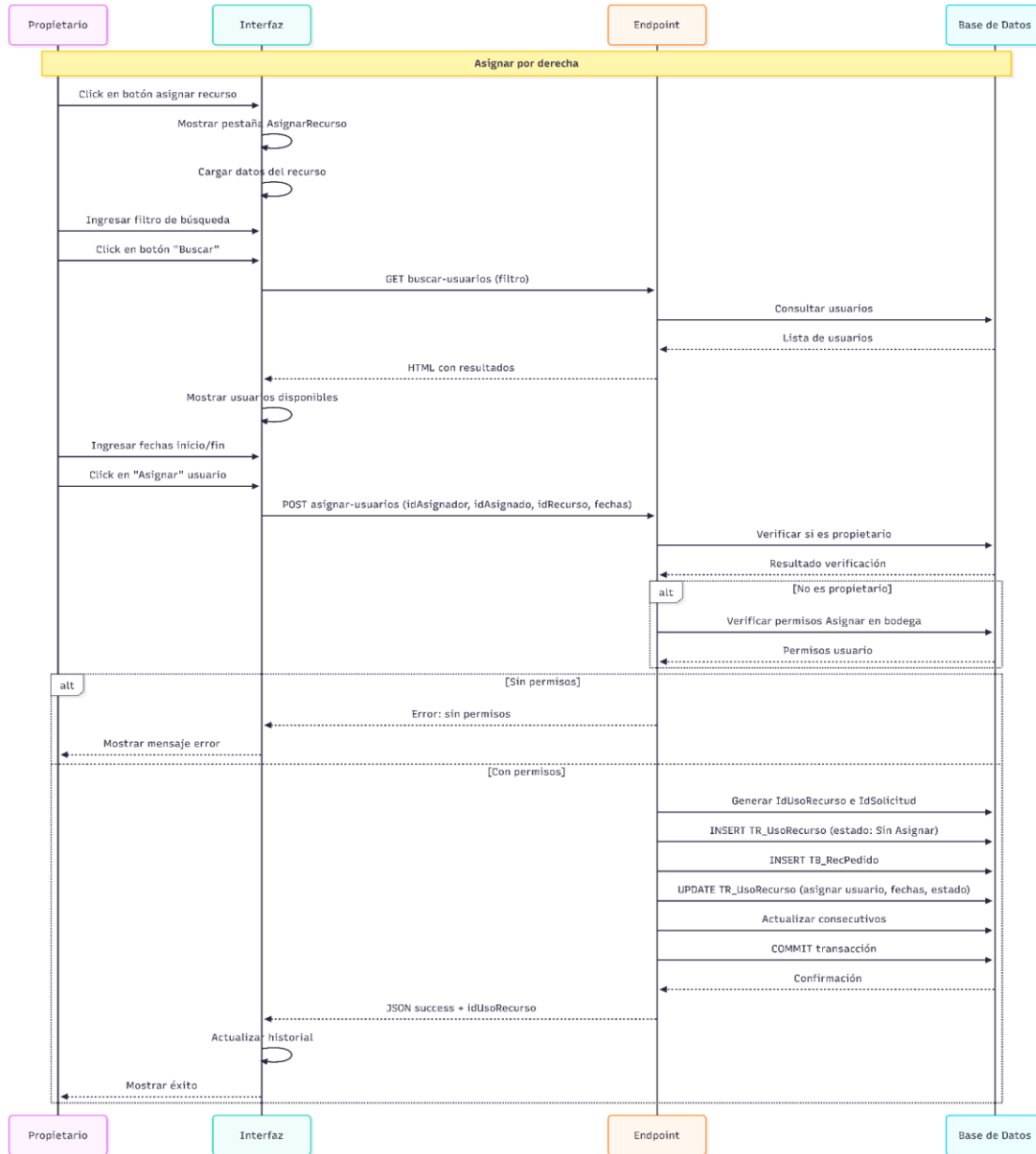
Diagrama de secuencia de creación de un recurso



### 4.1.3.2 Diagrama de secuencia de asignación por derecha

Figura 7

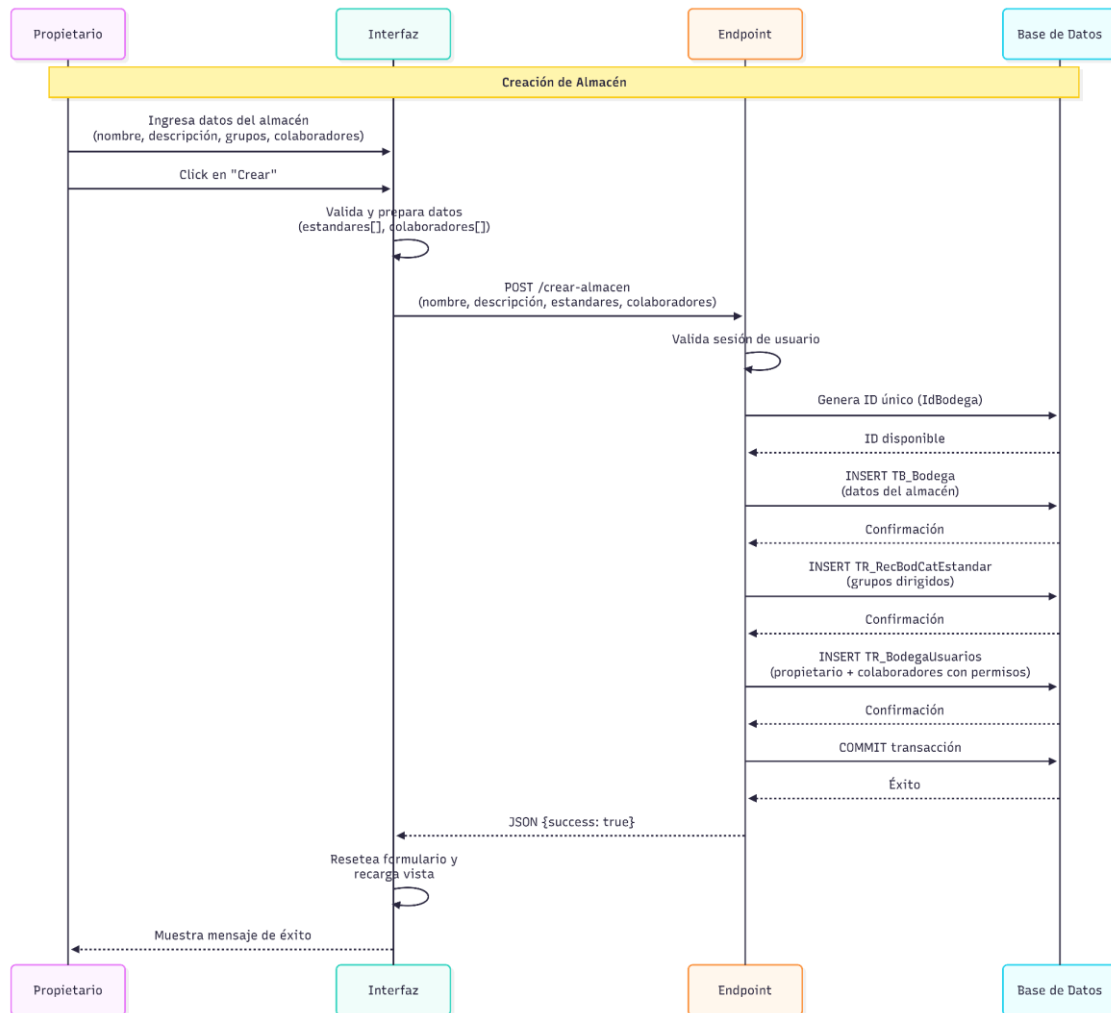
Diagrama de secuencia de asignación por derecha



4.1.3.3 Diagrama de secuencia de creación de un almacén

Figura 8

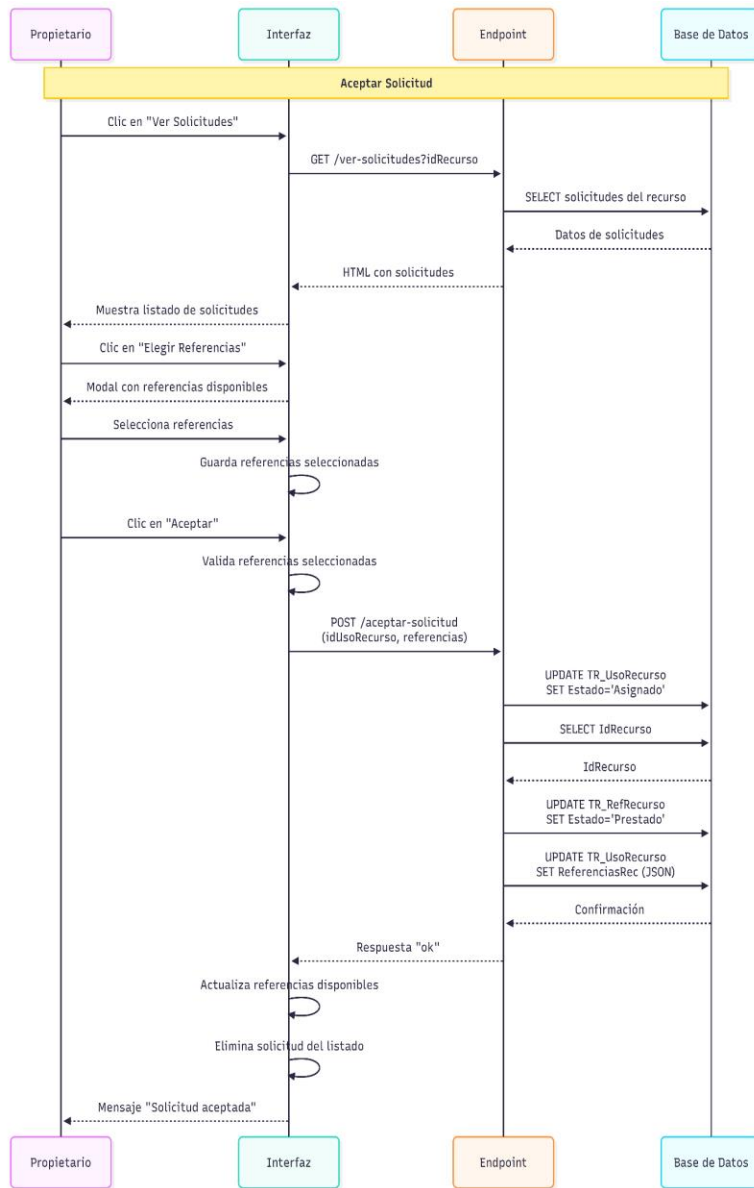
Diagrama de secuencia de creación de un almacén



4.1.3.4 Diagrama de secuencia de aceptación de solicitudes

Figura 9

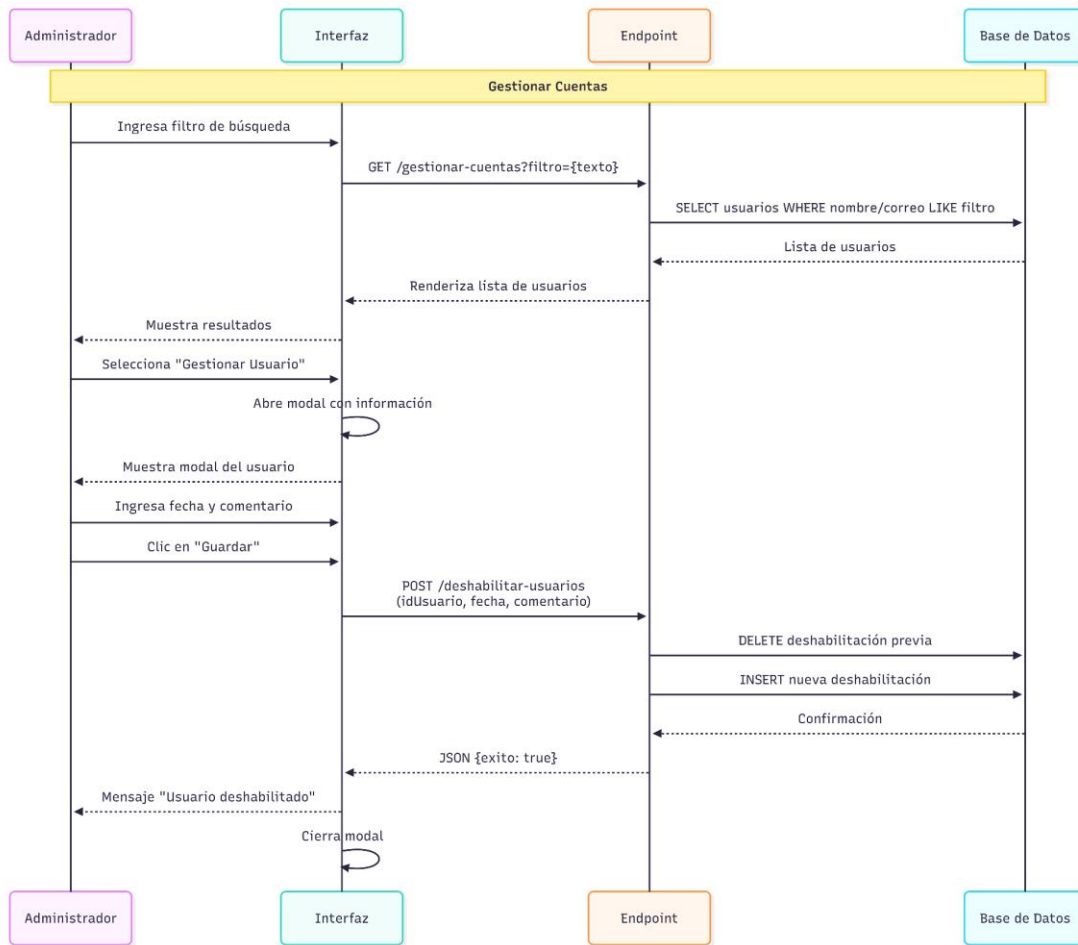
Diagrama de secuencia de aceptación de solicitudes



4.1.3.5 Diagrama de secuencia de gestión de cuentas

Figura 10

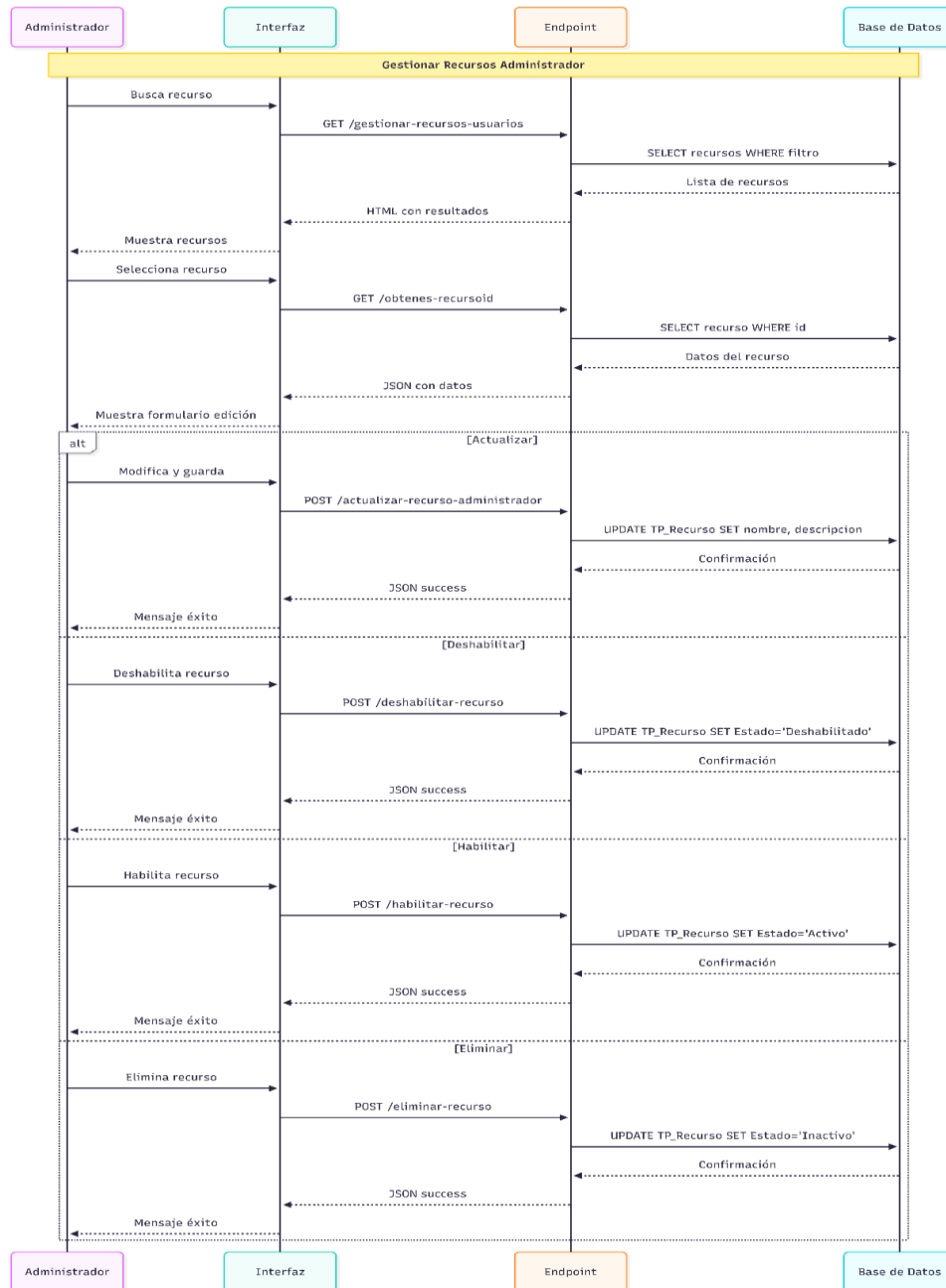
Diagrama de secuencia de gestión de cuentas



4.1.3.6 Diagrama de secuencia de gestión de recursos administrador

Figura 11

Diagrama de secuencia de gestión de recursos administrador



#### 4.1.4 Modelado de la base de datos

Se modeló un esquema relacional robusto que incluye las entidades necesarias para la gestión del sistema, estableciendo relaciones y restricciones de integridad que garantizan la consistencia y confiabilidad de la información.

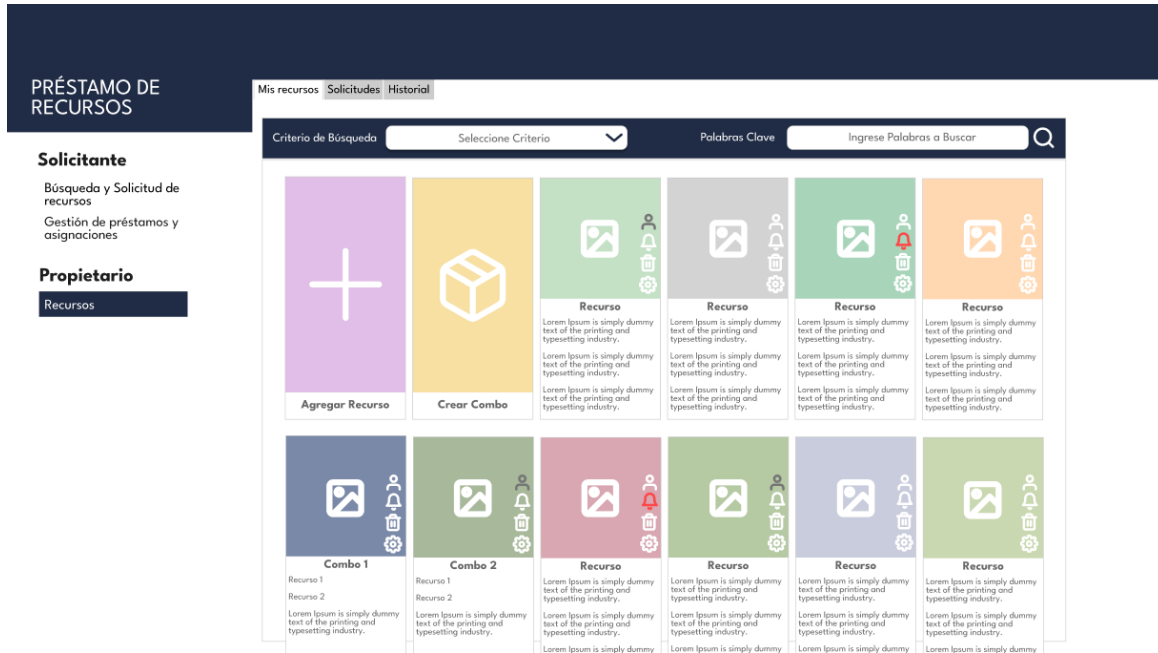


del módulo (gestión de recursos, solicitudes, historial, calendario) para asegurar coherencia visual y una experiencia de usuario intuitiva. Figma fue fundamental para crear prototipos interactivos, colaborar en tiempo real y validar el diseño antes de codificar, garantizando que todos los miembros del equipo compartieran la misma visión del proyecto.

#### **4.1.5.1 Mockup de la pestaña “Mis recursos”**

Esta interfaz fue concebida como el punto central para que los propietarios administren de manera eficiente su inventario. Se encuentra organizada en dos secciones claramente diferenciadas: un panel superior de pestañas que concentra las opciones principales del sistema (“Mis recursos”, “Solicitudes” e “Historial”), y un área de contenido principal destinada a la gestión de los recursos.

Desde esta pantalla, el propietario puede agregar nuevos recursos mediante el botón correspondiente o crear combos a través de la opción de creación de combos. Los recursos se presentan en forma de tarjetas, las cuales muestran su información básica y disponen de diversos botones que permiten ejecutar distintas funcionalidades asociadas a cada recurso.

**Figura 13***Mockup de la pantalla principal de gestión de recursos*

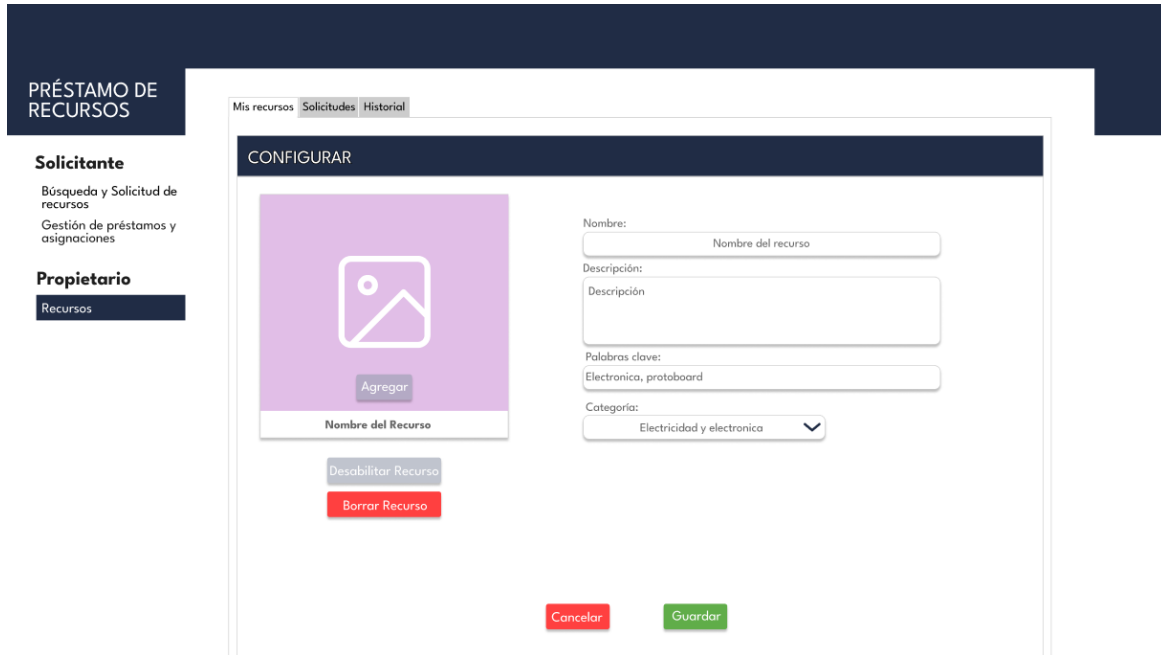
#### 4.1.5.2 Mockup de la pestaña “Configurar recursos”

El diseño propuso un formulario estructurado con campos para "Nombre del recurso", "Descripción", "Palabras clave" y un selector para "Categoría".

Se incluyeron controles de estado ("Deshabilitar Recurso") y la opción de borrar el recurso, junto con botones para "Cancelar" o "Guardar" los cambios. Esta disposición tiene como objetivo simplificar el proceso de catalogación, garantizando que la información necesaria para futuras búsquedas y filtros sea registrada de manera clara, completa y consistente.

**Figura 14**

*Mockup del formulario de edición y configuración de recursos*



PRÉSTAMO DE RECURSOS

Mis recursos Solicitudes Historial

**Solicitante**  
Búsqueda y Solicitud de recursos  
Gestión de préstamos y asignaciones

**Propietario**  
Recursos

**CONFIGURAR**

Nombre del Recurso

Nombre:

Descripción:

Palabras clave:

Categoría:

Agregar

Desabilitar Recurso

Borrar Recurso

Cancelar Guardar

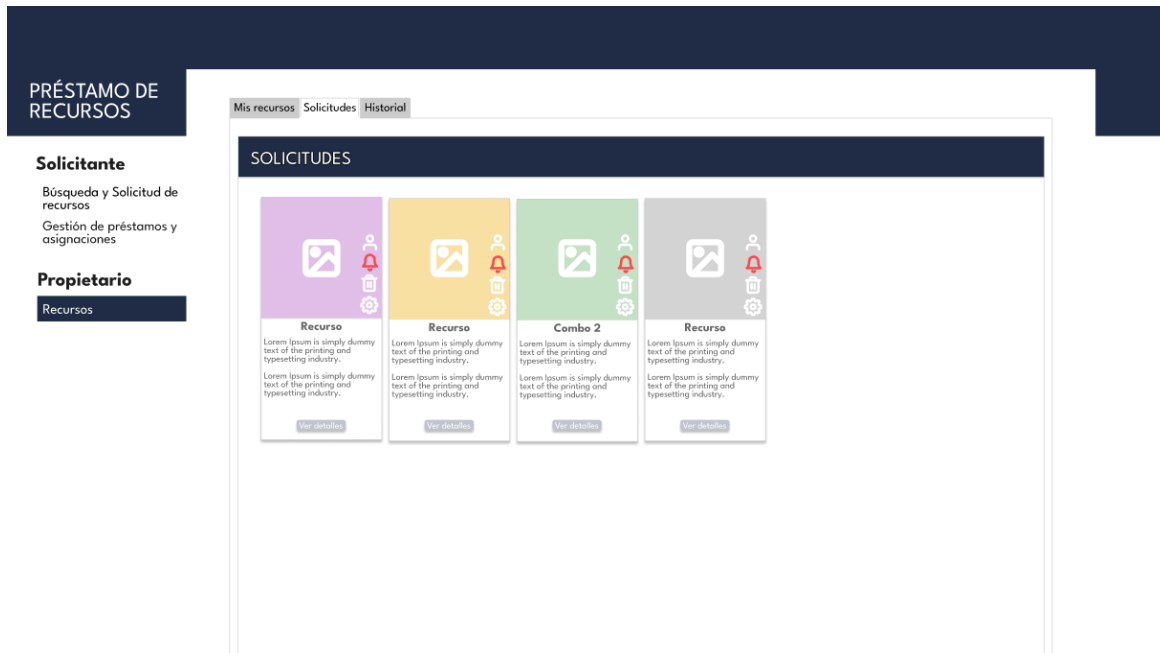
#### 4.1.5.3 Mockup de la visualización de solicitudes

La pestaña de "Solicitudes" presenta a los propietarios una lista clara de las peticiones de préstamo pendientes. Cada solicitud se representa como un elemento de lista independiente, acompañado de la información del recurso y de la solicitud y un botón de ver detalles.

El diseño priorizó la legibilidad y la acción rápida, permitiendo al propietario identificar de un vistazo las solicitudes recibidas y acceder a más información para su gestión. Esta pantalla actúa como el punto de entrada para la revisión y el proceso de aprobación o rechazo de préstamos.

**Figura 15**

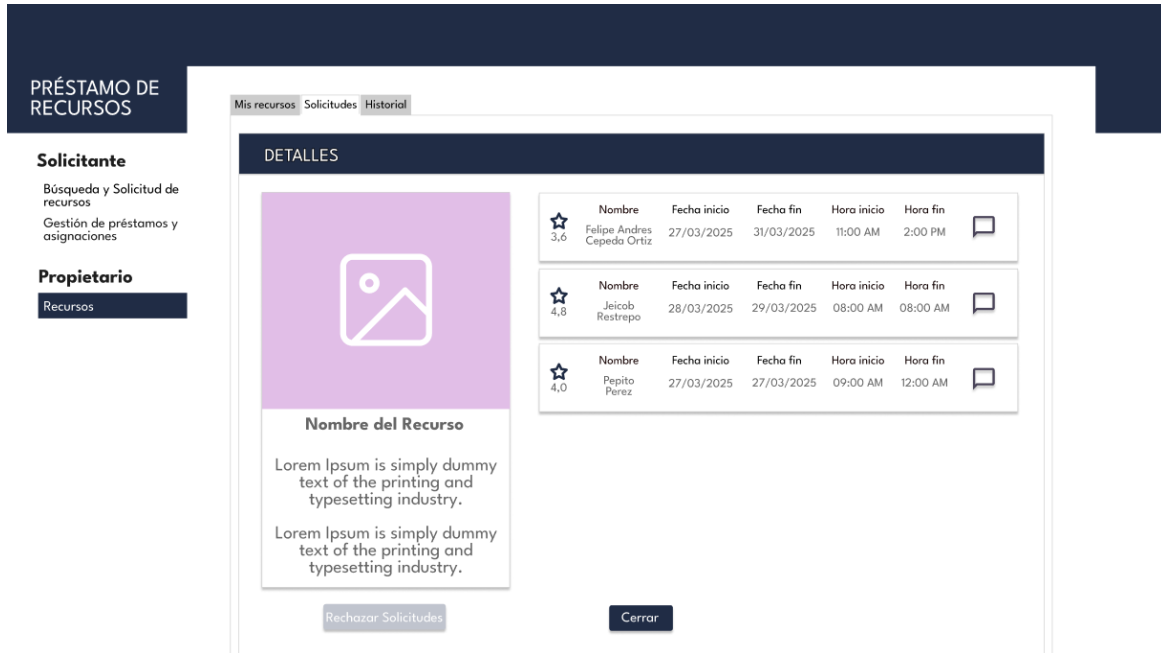
*Mockup de la pantalla de listado de solicitudes recibidas*



#### 4.1.5.4 Mockup de detalles de una solicitud

La ventana de los detalles se conceptualizó para ofrecer al propietario una vista ampliada de una solicitud específica. Aquí se muestra la información básica del recurso.

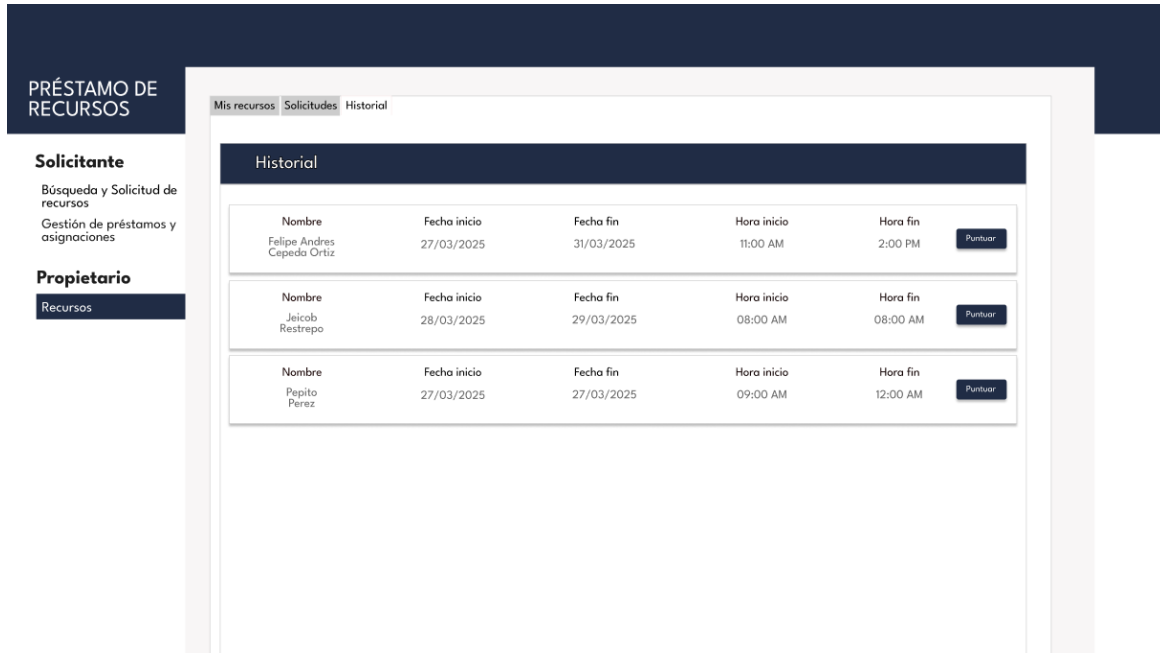
La interfaz incluye acciones decisivas para el propietario: puede ya sea rechazar las solicitudes o entrar a ver el detalle de cada una de las solicitudes, estas solicitudes muestran información de quien está pidiendo el préstamo. Este diseño busca facilitar la toma de decisiones con la información necesaria concentrada en un solo lugar.

**Figura 16***Mockup de la pantalla de detalles de una solicitud específica*

#### 4.1.5.5 Mockup del historial de préstamos

Esta interfaz fue diseñada para proporcionar una trazabilidad completa de los préstamos gestionados dentro del sistema. La información se presenta en una tabla que registra, para cada transacción, datos como el nombre del solicitante, la fecha de inicio y de finalización, así como la hora de inicio y la hora de fin.

Cada fila de la tabla representa un préstamo histórico y, al final de cada registro, se incorpora un botón de puntuación que permite realizar la fase de retroalimentación una vez finalizado el préstamo.

**Figura 17***Mockup de la pantalla de historial de préstamos*

## 4.2 Implementación de servicios del actor propietario

### 4.2.1 Almacenes

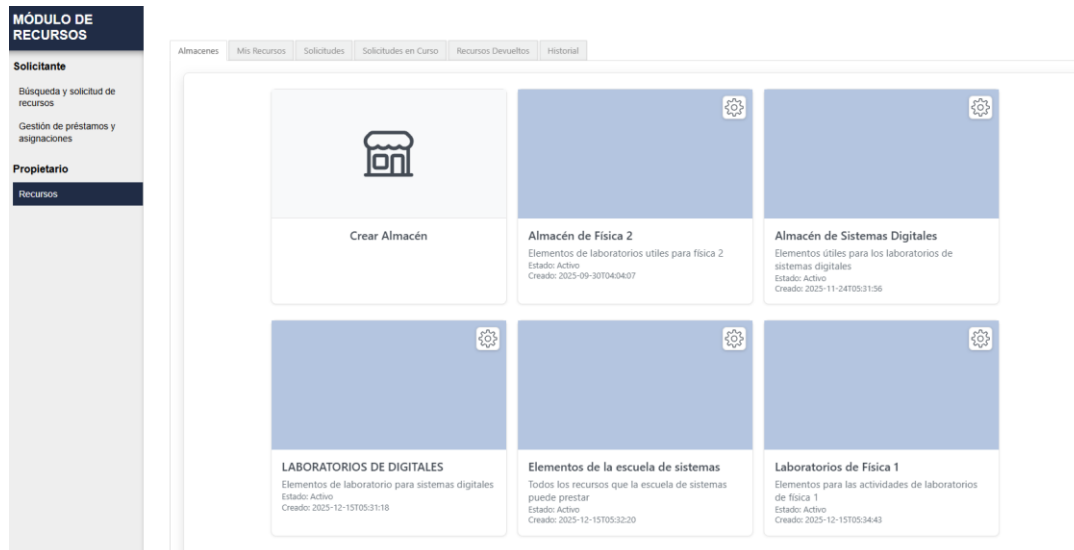
Esta pestaña permite al propietario visualizar y administrar de forma centralizada todos los almacenes que ha creado dentro del sistema. Al acceder a esta sección, el usuario podrá ver el listado completo de sus almacenes, junto con su información básica, facilitando una gestión clara y organizada.

Desde esta interfaz, el propietario tiene la posibilidad de editar los almacenes existentes, pudiendo modificar aspectos como el nombre, la descripción, los usuarios a los que va dirigido y los colaboradores asignados, así como los roles asociados a cada uno. Esto garantiza que la estructura del almacén pueda ajustarse en cualquier momento según las necesidades operativas.

Adicionalmente, la pestaña ofrece la opción de crear nuevos almacenes, permitiendo al propietario ampliar y organizar los recursos del sistema de manera flexible.

## Figura 18

### *Vista principal del actor propietario*



### 4.2.2 Creación de almacenes

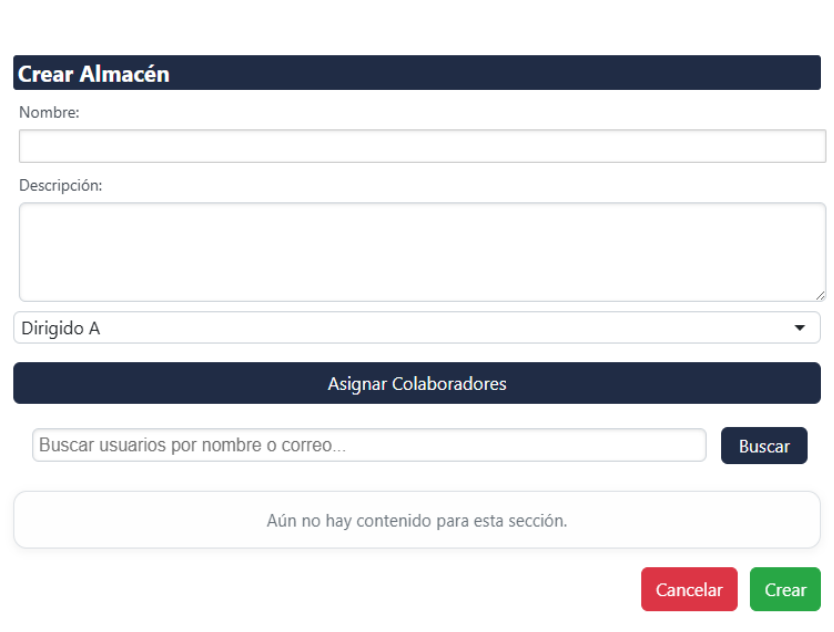
Permite al propietario organizar y administrar de manera estructurada los recursos disponibles dentro del sistema. A través de esta opción, el usuario puede crear uno o varios almacenes, definiendo un nombre representativo y una descripción que facilite su identificación y propósito.

Adicionalmente, el propietario puede establecer a quién va dirigido cada almacén, lo que permite segmentar y controlar el acceso según el tipo de usuario o grupo definido. Como parte de la gestión colaborativa, el sistema ofrece la posibilidad de asignar colaboradores a cada almacén, otorgándoles roles específicos que delimitan sus responsabilidades.

Los roles disponibles incluyen la gestión de recursos, la gestión de préstamos y la asignación por derecha, garantizando así una distribución clara de funciones y un control adecuado sobre las operaciones del almacén.

### Figura 19

*Vista sobre la creación de un almacén*



**Crear Almacén**

Nombre:

Descripción:

Dirigido A

**Asignar Colaboradores**

Buscar usuarios por nombre o correo...

Aún no hay contenido para esta sección.

#### 4.2.3 Creación de recursos

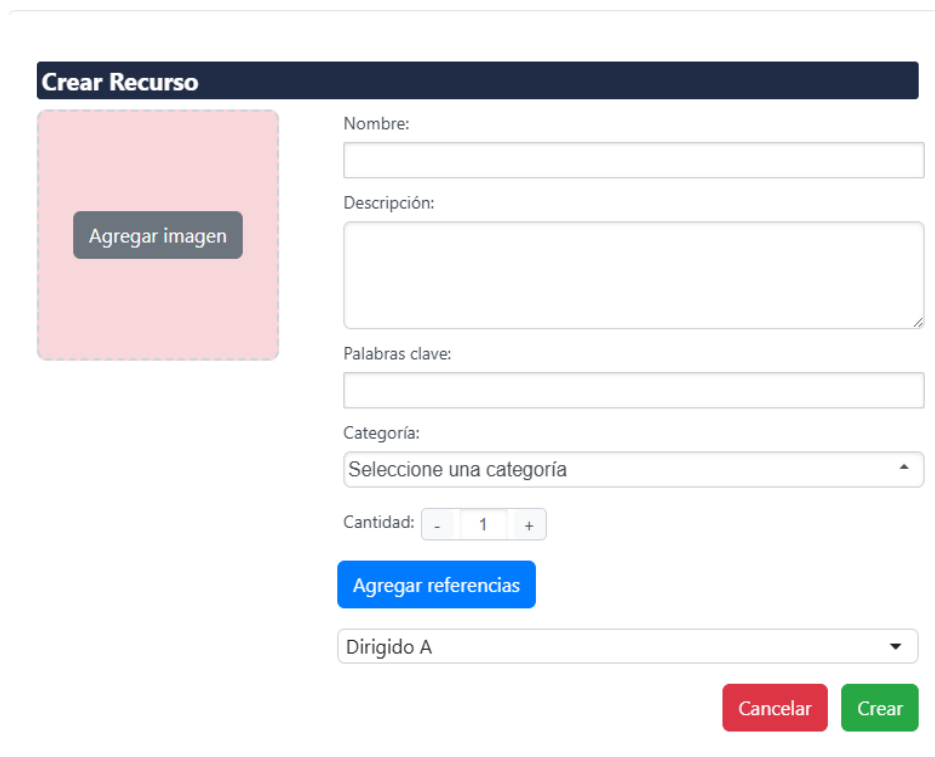
Permite al propietario registrar y organizar de forma detallada cada recurso dentro del sistema. Durante este proceso, el usuario puede asignar un nombre y una descripción que identifique claramente el recurso, así como definir palabras clave y una categoría que faciliten su búsqueda y clasificación.

Asimismo, es posible especificar la cantidad disponible del recurso y registrar referencias asociadas, lo que permite un control preciso del inventario. El sistema también permite indicar a quién va dirigido el recurso, asegurando que su uso esté alineado con el público objetivo definido.

Finalmente, el propietario puede agregar una imagen representativa del recurso, mejorando su identificación visual dentro del sistema y facilitando su reconocimiento por parte de los usuarios.

## Figura 20

*Vista sobre la creación de un recurso*



The screenshot shows a web form titled "Crear Recurso". On the left, there is a red dashed box containing a button labeled "Agregar imagen". To the right, the form contains the following fields and controls:

- Nombre:** A text input field.
- Descripción:** A larger text area with a small icon in the bottom right corner.
- Palabras clave:** A text input field.
- Categoría:** A dropdown menu with the text "Seleccione una categoría" and a small upward arrow.
- Cantidad:** A numeric input field with a value of "1" and minus/plus buttons on either side.
- Agregar referencias:** A blue button.
- Dirigido A:** A dropdown menu with the text "Dirigido A" and a downward arrow.
- Cancelar:** A red button.
- Crear:** A green button.

### 4.2.4 Edición de recursos

La funcionalidad de edición de recursos permite al usuario modificar y administrar la información de los recursos previamente registrados en el sistema. A través de esta opción, es posible actualizar el nombre, la descripción, las palabras clave, la categoría y la cantidad disponible del recurso, garantizando que la información se mantenga correcta y actualizada.

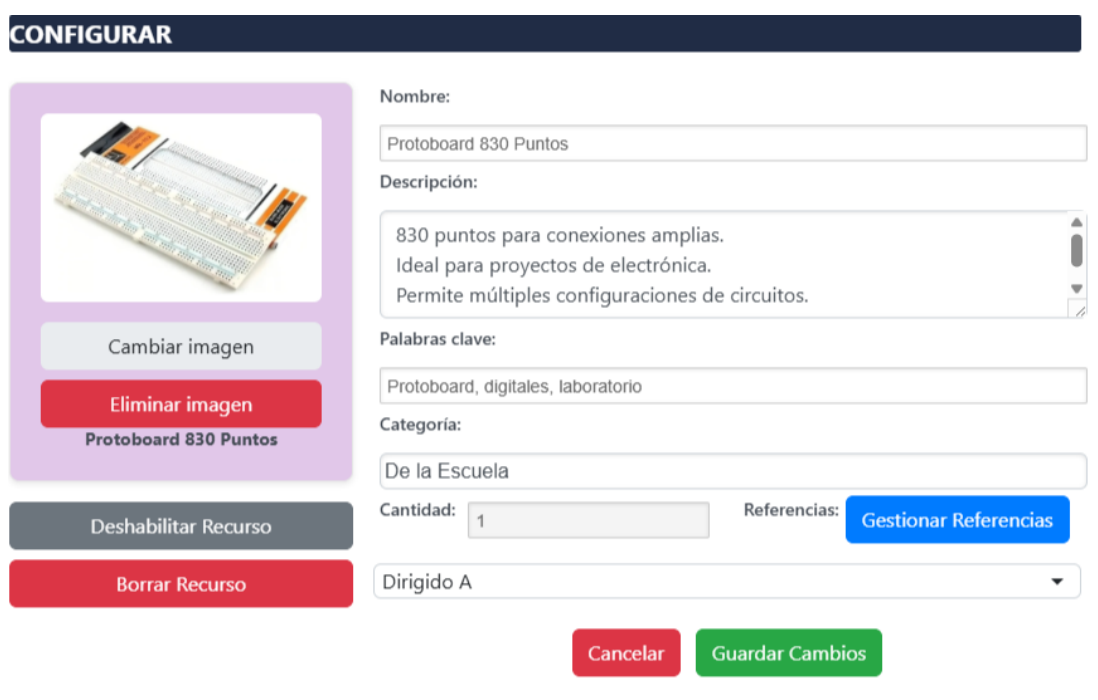
Asimismo, el usuario puede gestionar las referencias asociadas al recurso, definir o modificar a quién va dirigido y administrar su imagen, permitiendo cambiarla, agregar una nueva

o eliminarla según sea necesario. El sistema también ofrece la posibilidad de habilitar o deshabilitar un recurso, controlando su disponibilidad sin necesidad de eliminarlo permanentemente.

Finalmente, el usuario puede eliminar el recurso de manera definitiva cuando lo requiera.

## Figura 21

*Vista sobre la edición de un recurso*



**CONFIGURAR**

**Nombre:**  
Protoboard 830 Puntos

**Descripción:**  
830 puntos para conexiones amplias.  
Ideal para proyectos de electrónica.  
Permite múltiples configuraciones de circuitos.

**Palabras clave:**  
Protoboard, digitales, laboratorio

**Categoría:**  
De la Escuela

**Cantidad:** 1 **Referencias:** [Gestionar Referencias](#)

**Dirigido A:** Dirigido A

**Acciones:**  
Cambiar imagen  
Eliminar imagen  
Protoboard 830 Puntos  
Deshabilitar Recurso  
Borrar Recurso  
Cancelar  
Guardar Cambios

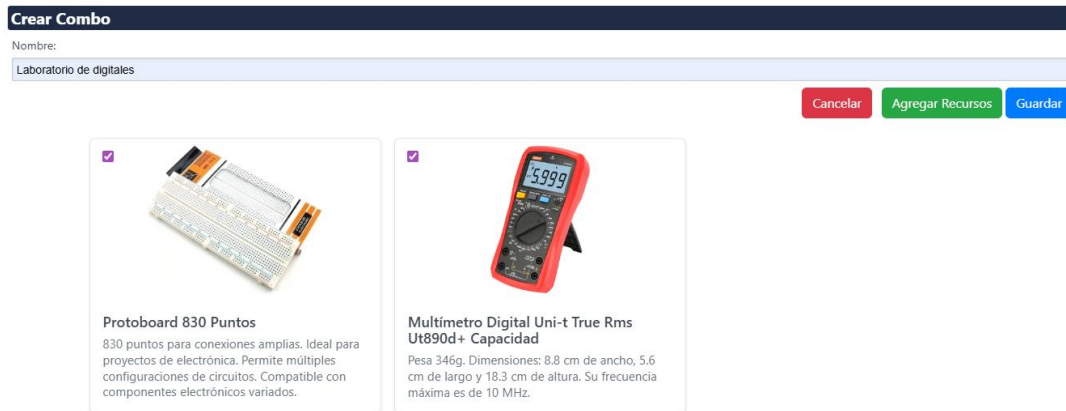
### 4.2.5 Creación de un combo

Permite agrupar varios recursos pertenecientes a un mismo almacén en una sola unidad lógica, facilitando su gestión y solicitud. A través de esta opción, el propietario puede seleccionar múltiples recursos disponibles y asociarlos a un combo, definiendo un nombre representativo que identifique claramente el conjunto.

Los combos pueden ser solicitados por los usuarios, permitiendo que todos los recursos que los componen se gestionen y se soliciten de manera conjunta en una sola operación.

**Figura 22**

*Vista sobre la creación de un combo*

**4.2.6 Asignaciones por derecha de un recurso**

Permite al propietario asignar directamente un préstamo a un usuario específico, sin necesidad de que este realice la solicitud inicial. Mediante esta opción, el propietario puede buscar y seleccionar al usuario a través de un buscador integrado, definir las fechas correspondientes del préstamo y realizar la asignación.

Una vez asignado, el usuario debe aceptar el préstamo para que este continúe su flujo normal, funcionando exactamente igual que un préstamo solicitado de manera tradicional.

**Figura 23**

*Vista sobre una asignación por derecha*

The screenshot shows a web interface titled "Asignar Usuarios". On the left, there is a large empty box labeled "Imagen recurso" with a blue "Computador" button at the bottom. On the right, there is a search section "Buscar usuario para asignar:" with an input field containing "abadmin" and a blue "Buscar" button. Below this is a table with two columns: "Usuario" and "Fecha/Hora Inicio". The table contains one row with the following data:

Usuario	Fecha/Hora Inicio
Abadmin Tibursio EISIWeb Tocancipa (caluwebapp@gmail.com)	dd/mm/aaaa --:--:--

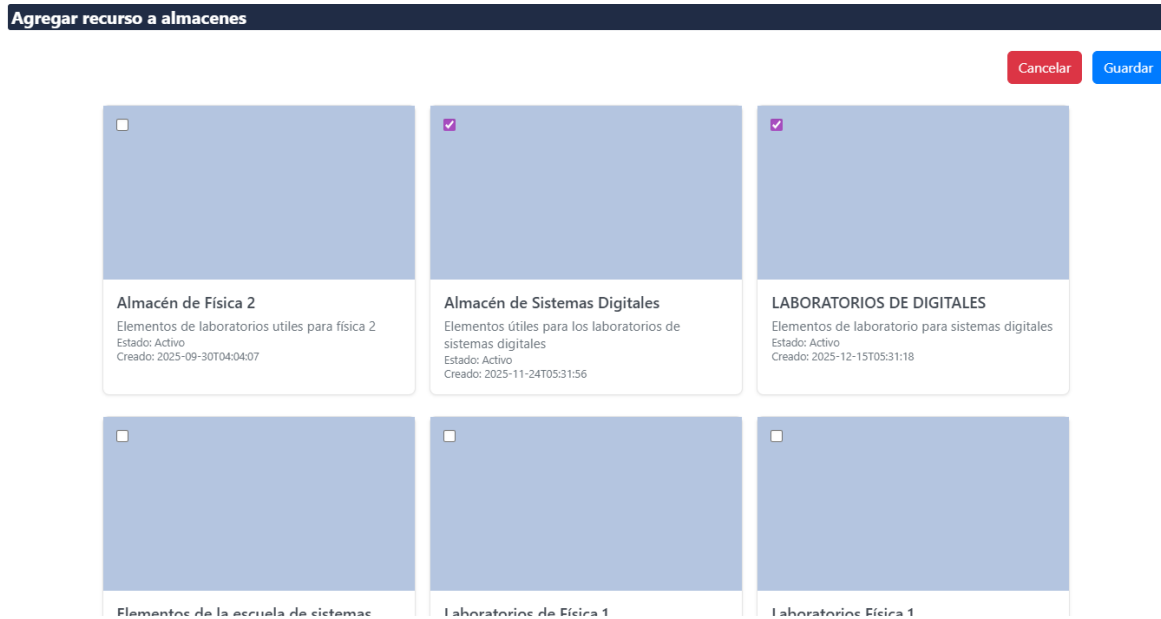
Below the table is a "Historial de Asignaciones" section with an empty input field.

#### ***4.2.7 Agregar recursos existentes a los almacenes***

Permite al propietario asociar un recurso previamente registrado a uno o varios almacenes del sistema. Mediante esta opción, el usuario puede seleccionar de forma explícita en qué almacenes desea que el recurso esté disponible, sin necesidad de duplicar su registro.

**Figura 24**

*Vista sobre la agregación de un recurso a varios almacenes*

**4.2.8 Solicitudes de recursos**

Permite al usuario visualizar y gestionar las solicitudes realizadas sobre los recursos de su propiedad. Estas solicitudes se presentan agrupadas por recurso, facilitando su identificación y organización.

El sistema ofrece una vista de calendario que permite consultar las fechas en las que cada recurso ya se encuentra prestado, brindando una visión clara de su disponibilidad. Adicionalmente, el usuario puede expandir cada solicitud para acceder a información más detallada.

**Figura 25**

*Vista sobre las solicitudes que le hacen al propietario*

The screenshot displays a user interface for managing resource requests. At the top, there is a navigation bar with tabs: 'Almacenes', 'Mis Recursos', 'Solicitudes', 'Solicitudes en Curso', 'Recursos Devueltos', and 'Historial'. The 'Solicitudes' tab is active, showing a list of requests under the heading 'Solicitudes'. Two request cards are visible:

- Protoboard 830 Puntos:** Requested by Abadmin Tibursio EIPWeb on 2026-01-05 09:41:23. Usage dates are from 2026-01-06 to 2026-01-21.
- Uni-t Multímetro Digital True Rms Ut890d+ Capacidad:** Requested by Abadmin Tibursio EIPWeb on 2026-01-05 09:42:24. Usage dates are from 2026-01-07 to 2026-01-19.

#### **4.2.9 Aceptación o rechazo de un recurso por parte del propietario**

Permite al propietario evaluar y gestionar las solicitudes realizadas sobre sus recursos. Al acceder al detalle de cada solicitud, el sistema presenta información relevante que facilita la toma de decisiones, incluyendo la calificación ponderada del solicitante, el evento o propósito para el cual se requiere el recurso, la cantidad solicitada y las fechas y horas de inicio y finalización del préstamo.

Adicionalmente, el propietario puede seleccionar de manera específica las referencias que serán prestadas. A partir de esta información, el usuario puede aceptar o rechazar la solicitud según su criterio.

Como apoyo al proceso de decisión, el sistema también permite iniciar un chat con el solicitante, facilitando la comunicación directa para aclarar dudas o acordar detalles antes de definir la aceptación o el rechazo del préstamo.

## Figura 26

*Vista sobre la aceptación o rechazo de las solicitudes*

← Volver



Protoboard 830  
Puntos

★ 0,0  
**Solicitante: Cristian Andres Cepeda**  
Evento: Laboratorio digitales  
ID Uso Recurso: UR2629  
Cantidad: 1  
Fecha solicitud: 2026-01-05 09:41:23  
Fecha uso: 2026-01-06  
Hora inicio: 12:00:00 | Hora fin: 00:00:00

Elegir referencias  
No has seleccionado referencias.

✓ ✗ 💬

### 4.2.10 Solicitudes en curso, cuando ya fue aceptada la solicitud

Permite al usuario visualizar un listado de los préstamos que se encuentran actualmente activos, es decir, aquellos que ya han sido aceptados y se encuentran en proceso. Este listado presenta información clave para el seguimiento de los préstamos, incluyendo el nombre del usuario solicitante, el identificador de la solicitud, el nombre del recurso prestado y las fechas y horas de inicio y finalización del préstamo.

**Figura 27**

*Vista de las solicitudes que están en curso*

Solicitudes en Curso							
Nombre Usuario	ID	Recurso	Fecha Inicio	Hora Inicio	Fecha Fin	Hora Fin	
Leidy Andrea Carreño Galvis	UR2630	Uni-t Multímetro Digital True Rms Ut890d+ Capacidad	2026-01-07	13:00:00	2026-01-19	12:00:00	
Cristian Andres Cepeda	UR2629	Protoboard 830 Puntos	2026-01-06	12:00:00	2026-01-21	00:00:00	

**4.2.11 Recursos devueltos**

Permite al usuario visualizar los préstamos que han sido marcados como devueltos por parte del solicitante. En esta vista se presenta la misma información que en las solicitudes en curso, incluyendo el nombre del usuario, el identificador de la solicitud, el nombre del recurso y las fechas y horas de inicio y finalización del préstamo.

Adicionalmente, esta funcionalidad ofrece la opción de finalizar el préstamo, acción que debe ser realizada por el propietario una vez confirmada la recepción del recurso.

**Figura 28**

*Vista sobre los recursos que se devolvieron al propietario*

Recursos Devueltos								
Nombre Usuario	ID	Recurso	Fecha Inicio	Hora Inicio	Fecha Fin	Hora Fin	Acción	
Leidy Andrea Carreño Galvis	UR2630	Uni-t Multímetro Digital True Rms Ut890d+ Capacidad	2026-01-07	13:00:00	2026-01-19	12:00:00	Finalizar	
Cristian Andres Cepeda	UR2629	Protoboard 830 Puntos	2026-01-06	12:00:00	2026-01-21	00:00:00	Finalizar	

**4.2.12 Historial de solicitudes**

Permite al usuario consultar los préstamos que ya han finalizado dentro del sistema. Esta vista presenta la misma información que la sección de recursos devueltos, incluyendo los datos del usuario solicitante, el identificador de la solicitud, el recurso prestado y las fechas y horas correspondientes al préstamo.

Adicionalmente, el sistema habilita la opción de calificar al usuario solicitante, permitiendo asignar una nota basada en el desarrollo del préstamo. Esta calificación solo puede realizarse una única vez y se tiene en cuenta en el cálculo del ponderado de la calificación general del usuario.

## Figura 29

*Vista sobre el historial de solicitudes*



Nombre Usuario	ID	Recurso	Fecha Inicio	Hora Inicio	Fecha Fin	Hora Fin	Acción
Leidy Andrea Carreño Galvis	UR2630	Uni-t Multímetro Digital True Rms Ut890d+ Capacidad	2026-01-07	13:00:00	2026-01-19	12:00:00	Calificado: 5
Cristian Andres Cepeda	UR2629	Protoboard 830 Puntos	2026-01-06	12:00:00	2026-01-21	00:00:00	<a href="#">★ Calificar</a>

### 4.2.13 Calificación

Permite al propietario evaluar la experiencia del préstamo una vez este ha finalizado. A través de esta opción, el usuario puede asignar una puntuación en una escala del 1 al 10 y registrar un comentario descriptivo que respalde la calificación otorgada.

## Figura 30

*Vista sobre la calificación del actor solicitante*



★ Calificar a Leidy Andrea Carreño Galvis

Comentario:

Calificación (1 a 10):

5

Enviar Calificación

### 4.3 Implementación de servicios del actor Administrador

#### 4.3.1 Generador de PDF y CSV

Permite al administrador exportar la información de los préstamos registrados en el sistema, basándose en los filtros aplicados previamente en la vista de auditoría.

Los reportes pueden ser exportados tanto en formato PDF como en formato CSV e incluyen información detallada de los préstamos, como el identificador de la solicitud, el identificador del asignador, el identificador del solicitante, el identificador del usuario asignado, el identificador de uso del recurso, el identificador del recurso, el evento asociado y el estado del préstamo.

#### Figura 31

Vista sobre la descarga en PDF

#### Auditoría de Recursos

ID Solicitud	Asignador	Solicitante	Asignado	ID Uso Recurso	Recurso	Evento	Estado
RP1			U5944	UR971	RC106	Primera prueba del módulo de recursos	En Estudio
RP27			U0	UR1009	RC106		Sin Asignar
RP3			U0	UR978	RC106	Prueba1	CanceladoPorUsrSolicitante
RP9			U0	UR986	RC106	Felipe	En Estudio
			U919	UR188	RC106	Evento presentación	Asignado
			U213	UR192	RC106	situacion	Asignado

#### Figura 32

Vista sobre la descarga en CSV

A	B	C	D	E	F	G	H
Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8
ID Solicitud	Asignador	Solicitante	Asignado	ID Uso Recurso	Recurso	Evento	Estado
RP1	null	null	U5944	UR971	RC106	Primera prueba del módulo de recursos	En Estudio
RP27	null	null	U0	UR1009	RC106		Sin Asignar
RP3	null	null	U0	UR978	RC106	Prueba1	CanceladoPorUsrSolicitante
RP9	null	null	U0	UR986	RC106	Felipe	En Estudio
null	null	null	U919	UR188	RC106	Evento presentación	Asignado
null	null	null	U213	UR192	RC106	situacion	Asignado

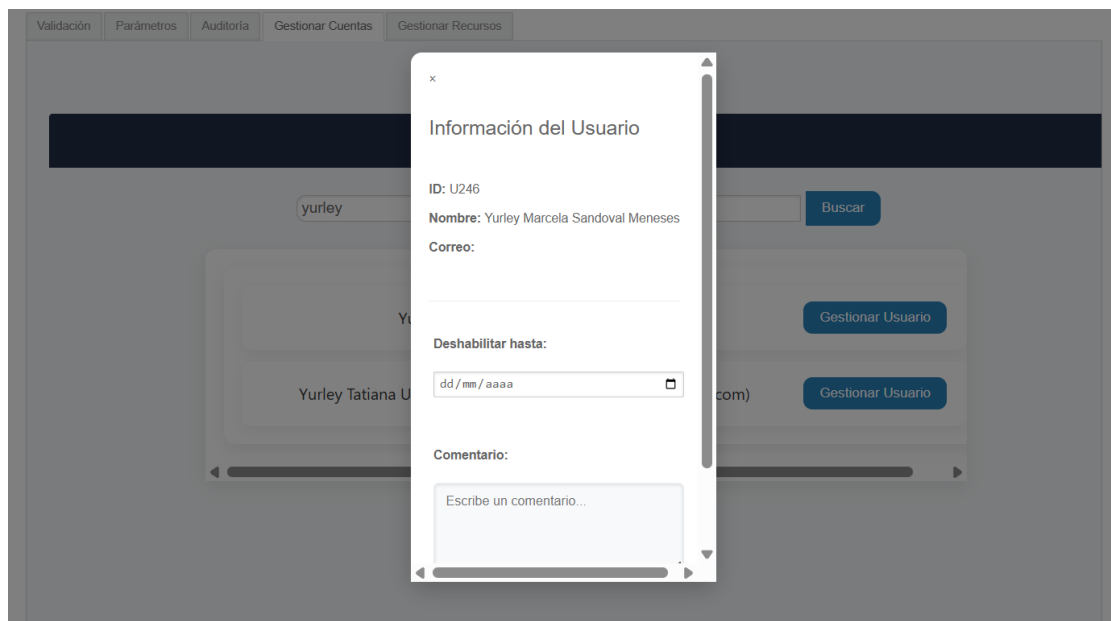
### 4.3.2 Gestionar usuarios

Permite al administrador administrar las cuentas de los usuarios registrados en el sistema. A través de esta opción, el administrador puede buscar al usuario que desea gestionar mediante su nombre o correo electrónico.

Una vez seleccionado el usuario, el administrador puede acceder a la opción de gestión, la cual despliega un modal con la información detallada del usuario. Desde esta vista, es posible definir la fecha hasta la cual la cuenta del usuario permanecerá suspendida y registrar un comentario asociado a la acción realizada.

#### Figura 33

*Vista sobre la deshabilitación de un usuario*



Cuando un usuario se encuentra suspendido, al ingresar al módulo el sistema le muestra un mensaje de cuenta inhabilitada, informando de manera clara que su acceso ha sido restringido temporalmente. En este mensaje se indica la fecha hasta la cual la cuenta permanecerá suspendida.

Adicionalmente, el sistema presenta el comentario registrado por el administrador, brindando contexto sobre el motivo de la suspensión.

### Figura 34

*Vista sobre usuario deshabilitado*



#### 4.3.3 Gestión de recursos administrador

Permite al administrador supervisar y administrar los recursos registrados por cualquier usuario dentro del sistema. A través de esta opción, el administrador puede buscar un recurso específico utilizando su nombre.

Una vez seleccionado el recurso, el administrador puede acceder a su configuración, donde es posible modificar sus datos básicos, como el nombre del recurso, la descripción y las palabras clave. Asimismo, el sistema permite habilitar o deshabilitar el recurso.

Finalmente, el administrador cuenta con la opción de eliminar el recurso de manera definitiva cuando éste infringe las normas.

**Figura 35**

*Vista sobre la gestión de recursos*



The screenshot shows a web interface titled "CONFIGURAR RECURSO" (Configure Resource). The resource name is "Salón Azul". At the top, there are two buttons: "Deshabilitar Recurso" (Disable Resource) and "Eliminar Recurso" (Delete Resource). Below these are three input fields: "Nombre del Recurso \*" (Resource Name \*) containing "Salón Azul", "Descripción \*" (Description \*) containing "Salón Azul", and "Palabras Clave" (Keywords) containing "Ej: Electrónica, protoboard, laboratorio". At the bottom, there are two buttons: "Cancelar" (Cancel) and "Guardar Cambios" (Save Changes).

Las funcionalidades descritas en esta sección, correspondientes a la implementación de los servicios del actor propietario y administrador, se encuentran respaldadas con evidencias audiovisuales. Dichas evidencias muestran el funcionamiento del sistema, incluyendo la gestión, creación y edición de almacenes y recursos, así como la interacción del usuario propietario y del administrador.

El material audiovisual puede consultarse en el **Apéndice A**, donde se dispone de enlaces a vídeos alojados en Google Drive que permiten observar el comportamiento real del sistema.

#### **4.4 Integración con la Plataforma COMA**

La integración fue completa y transparente para los usuarios finales:

- **Reutilización de la seguridad:** Se aprovechó el sistema de autenticación/autorización implementado en COMA.
- Se aseguró una apariencia uniforme en toda la interfaz aplicando la guía de estilos Elise, utilizando principalmente CSS puro para mantener la identidad gráfica institucional sin depender de frameworks externos.
- El sistema se conectó únicamente a Diamante, aprovechando su estructura específica para cada escuela y centralizando allí toda la gestión de información.

#### 4.5 Pruebas y Validación Continua

Para asegurar la calidad, cada funcionalidad pasó por distintas etapas de prueba:

- **De integración:** Comprobando el flujo entre JSP, Servlets y MySQL.
- **De usabilidad interna:** El equipo probó los escenarios reales de propietarios y solicitantes, ajustando la experiencia.
- **De rendimiento:** Midiendo tiempos de carga en listados y consultas.
- **Manuales de uso:** Gracias a estas pruebas, se iniciaron también la redacción de manuales de usuario, con guías paso a paso para que la comunidad pudiera comprender fácilmente cómo registrar recursos, realizar solicitudes, gestionar préstamos o generar reportes.

Las pruebas funcionales descritas en los numerales 5.6 y 5.7 fueron documentadas y registradas de manera sistemática por los integrantes del semillero de investigación.

La evidencia de dichas pruebas se presenta en una hoja de cálculo que contiene los escenarios evaluados, los resultados obtenidos y las observaciones correspondientes, permitiendo verificar el cumplimiento de los requisitos funcionales del sistema.

Este registro puede consultarse en el **Apéndice B**, mediante un enlace a una carpeta de Google Drive disponible para su visualización.

#### **4.6 Prueba funcional: Creación de almacenes**

Esta prueba funcional corresponde a la funcionalidad de “Creación de almacenes”, asociado al actor Propietario, quien cuenta con acceso autorizado al módulo de préstamos de recursos. El objetivo de la prueba es verificar que el sistema permite la creación correcta de almacenes y gestione adecuadamente los distintos escenarios de uso.

##### **Escenarios de prueba evaluados:**

- **Escenario exitoso:** El usuario ingresa correctamente los datos del almacén y el sistema permite su creación sin inconvenientes.
- **Escenario de error:** El usuario no puede crear el almacén debido a información inválida o incompleta.
- **Escenario de visualización incorrecta:** Los datos ingresados no se muestran correctamente después de crear el almacén.

##### **Requerimientos validados:**

- Acceso del usuario propietario al módulo de gestión de recursos.
- Disponibilidad del formulario de creación de almacenes.
- Validación de campos obligatorios.
- Persistencia y visualización correcta de la información registrada.

##### **Comportamiento esperado:**

El sistema debe permitir al usuario digitar correctamente los datos del almacén, configurar su propósito, definir a quién va dirigido y asignar colaboradores en caso de ser necesario. Una vez creada la información, el almacén debe registrarse exitosamente y mostrarse de forma adecuada en la plataforma, indicando el estado “Completado” cuando el proceso se ejecute sin errores.

#### **4.7 Prueba funcional: Creación de recursos**

Esta prueba funcional corresponde al caso de uso “Creación de recursos”, asociado al actor Propietario, quien cuenta con acceso autorizado al módulo de gestión de recursos. El objetivo de la prueba es verificar que el sistema permite la creación correcta de recursos y gestione adecuadamente los distintos escenarios de uso.

##### **Escenarios de prueba evaluados:**

- **Escenario exitoso:** El usuario ingresa correctamente los datos del recurso, selecciona su imagen y define sus referencias, permitiendo al sistema crear el recurso sin inconvenientes.
- **Escenario de error:** El usuario no puede crear el recurso debido a información inválida, incompleta o a la falta de un almacén asociado.
- **Escenario de visualización incorrecta:** Los datos ingresados no se muestran correctamente después de crear el recurso.

##### **Requerimientos validados:**

- Acceso del usuario propietario al módulo de gestión de recursos.
- Existencia de al menos un almacén creado.
- Disponibilidad del formulario de creación de recursos.
- Validación de campos obligatorios.
- Persistencia y correcta visualización de la información registrada.

**Comportamiento esperado:**

El sistema debe permitir al usuario digitar correctamente los datos del recurso, configurarlo según sus necesidades, asignarlo a un grupo y seleccionar una imagen en caso de requerirse. Una vez creada la información, el recurso debe registrarse exitosamente y mostrarse de forma adecuada en la plataforma.

**4.8 Prueba funcional: Agregar recursos ya creados a los almacenes**

Esta prueba funcional corresponde al caso de uso “Agregar recursos ya creados a los almacenes”, asociado al actor Propietario, quien cuenta con acceso autorizado al módulo de gestión de recursos. El objetivo de la prueba es verificar que el sistema permite asociar correctamente recursos previamente creados a uno o varios almacenes y que dicha asociación se refleje de forma adecuada en la plataforma.

**Escenarios de prueba evaluados:**

- **Escenario exitoso:** El usuario selecciona uno o varios almacenes desde la sección “*Mis Recursos*” y agrega correctamente un recurso ya creado, visualizándose de inmediato en los almacenes seleccionados.
- **Escenario de error:** El usuario no puede agregar el recurso a ningún almacén.
- **Escenario de visualización incorrecta:** No se muestran los almacenes del usuario o el recurso no aparece en los almacenes a los que fue agregado.

**Requerimientos validados:**

- Acceso del usuario propietario al módulo de gestión de recursos.
- Existencia de al menos un almacén creado.
- Existencia de al menos un recurso previamente creado.

- Disponibilidad de la opción para asociar recursos a almacenes.
- Actualización y visualización correcta de la información.

#### **Comportamiento esperado:**

El sistema debe permitir al usuario seleccionar los almacenes a los cuales desea agregar un recurso ya creado. Una vez realizada la acción, el recurso debe aparecer inmediatamente en los almacenes seleccionados, reflejando correctamente la asociación sin errores en la visualización.

#### **4.9 Prueba funcional: Gestionar cuentas**

Esta prueba funcional corresponde al caso de uso “**Gestionar cuentas**”, asociado al actor **Administrador**, quien cuenta con los permisos necesarios para administrar las cuentas de los usuarios del sistema. El objetivo de la prueba es verificar que el administrador pueda buscar usuarios, gestionar sus cuentas y aplicar restricciones cuando sea necesario.

#### **Escenarios de prueba evaluados:**

- **Escenario de error 1:** El buscador de usuarios no funciona correctamente.
- **Escenario de error 2:** Al encontrar un usuario, el botón “**Gestionar usuario**” no responde o no permite realizar acciones.
- **Escenario de error 3:** Al abrir el modal, completar la información y presionar “**Guardar**”, no se crea la restricción para suspender la cuenta.
- **Escenario exitoso:** Cuando no se presentan errores y el proceso se completa correctamente, el usuario suspendido visualiza un mensaje indicando que su cuenta ha sido deshabilitada al intentar ingresar al módulo.

#### **Requerimientos validados:**

- Acceso del usuario administrador al módulo de gestión de cuentas.
- Permisos para buscar, gestionar y deshabilitar cuentas de otros usuarios.
- Disponibilidad del buscador de usuarios.
- Funcionamiento correcto del modal de gestión de cuentas.
- Aplicación efectiva de restricciones y notificación al usuario afectado.

### **Comportamiento esperado:**

El sistema debe permitir que el administrador busque usuarios desde la sección de Gestión de Cuentas, seleccione un usuario y aplique la deshabilitación indicando el motivo correspondiente. Una vez deshabilitada la cuenta, si el usuario intenta acceder al módulo, deberá visualizar un mensaje informando la razón de la deshabilitación y el tiempo durante el cual permanecerá bloqueado.

### **4.10 Prueba funcional: Exportar información en PDF y CSV**

Esta prueba funcional corresponde al caso de uso “**Exportar información en PDF y CSV**”, asociado al actor **Administrador**, quien cuenta con los permisos necesarios para acceder a la información de auditoría del sistema. El objetivo de la prueba es verificar que el sistema permite exportar correctamente la información de las solicitudes aplicando los filtros correspondientes.

### **Escenarios de prueba evaluados:**

- **Escenario de error:** Al seleccionar los filtros correspondientes y ejecutar la opción “**Exportar**”, no se genera ningún archivo ni en formato PDF ni en formato CSV.
- **Escenario exitoso:** El administrador aplica correctamente los filtros y el sistema genera los archivos de exportación en los formatos PDF y CSV.

**Requerimientos validados:**

- Acceso del usuario administrador al módulo de Gestión de Módulos.
- Permisos para acceder a la pestaña de Auditoría.
- Disponibilidad de filtros para la información de solicitudes.
- Funcionalidad de exportación en los formatos PDF y CSV.
- Generación correcta de los archivos con la información filtrada.

**Comportamiento esperado:**

El sistema debe permitir que el administrador acceda a la sección de Gestión de Módulos y posteriormente a la pestaña de Auditoría, donde podrá aplicar correctamente los filtros disponibles. Al ejecutar la opción de exportación, la información resultante debe generarse y descargarse correctamente en ambos formatos disponibles: **CSV y PDF**.

**4.11 Despliegue en entorno de pruebas**

- **Servidor de pruebas:** Apache Tomcat 9.0.88 con MySQL 8.3.0.
- **Datos simulados:** Se cargaron escenarios realistas con usuarios, solicitudes y recursos de ejemplo.
- **Pruebas de aceptación:** Validando el cumplimiento de todos los requerimientos definidos.

**4.12 Principales desafíos y soluciones**

- **Gestión de estados de préstamo:** Se estableció un ciclo de vida formal (Solicitado, Aprobado, Entregado, Devuelto, Rechazado).
- **Subida de imágenes:** Implementación de un servicio para manejar archivos de manera eficiente y segura.

- **Concurrencia en solicitudes:** Se aplicaron *locks* a nivel de base de datos para impedir la asignación doble de un mismo recurso.

#### **4.13 Impacto y valor agregado**

El módulo no solo cumple con los requerimientos técnicos, sino que fortalece la cultura de colaboración académica en la plataforma COMA. Los usuarios ahora cuentan con un sistema confiable y fácil de usar, acompañado de manuales que facilitan su apropiación. Además, la trazabilidad de los préstamos, la seguridad en el manejo de datos y la posibilidad de generar reportes estratégicos lo convierten en una herramienta clave para la gestión académica y administrativa de la Universidad.

### **5. Conclusiones**

A lo largo del desarrollo de este proyecto se evidenció la importancia de contar con un sistema centralizado y confiable para la gestión de préstamos de recursos físicos dentro de la Universidad Industrial de Santander, como parte de la plataforma COMA. El análisis de la situación inicial permitió identificar que los procesos manuales y descentralizados generaban pérdidas de tiempo, dificultades en la trazabilidad y una experiencia de usuario poco satisfactoria.

La construcción e integración del módulo de préstamos dentro de COMA representó una solución efectiva a estas problemáticas, logrando automatizar tareas críticas como la solicitud, aprobación, devolución y generación de reportes. Esto se traduce en una mejora significativa en la eficiencia de los procesos, en la transparencia de la información y en la reducción de errores humanos asociados al manejo manual.

De manera particular, la validación de los prototipos y el despliegue controlado permitieron garantizar que el sistema funcionara de manera estable, escalable y segura, asegurando la integridad de la información y ofreciendo a los usuarios una interfaz intuitiva que fomenta la adopción del módulo en el entorno universitario.

Asimismo, la implementación de este módulo no solo resuelve una necesidad operativa inmediata, sino que también fortalece la capacidad de la institución para gestionar sus recursos de forma estratégica. El acceso a información centralizada y en tiempo real abre la posibilidad de tomar decisiones más informadas en cuanto a la planeación y mantenimiento de los recursos físicos de la universidad.

Finalmente, este proyecto constituye un paso importante en el proceso de consolidación de COMA como un ecosistema digital integral. La experiencia adquirida y los resultados obtenidos sientan las bases para futuras mejoras e innovaciones, orientadas hacia la transformación digital de los procesos académicos y administrativos en beneficio de toda la comunidad universitaria.

### **5.1 Trabajo a futuro**

Si bien el desarrollo del módulo de préstamos de recursos constituye un avance significativo dentro de la plataforma COMA, también abre la puerta a nuevas oportunidades de mejora y expansión. Algunas líneas de trabajo a futuro incluyen:

- **Ampliación de funcionalidades y flujos internos:** se podrían diseñar procesos más detallados para la gestión de préstamos, como recordatorios automáticos de fechas de devolución, renovaciones en línea y sanciones por incumplimiento. Estas mejoras harían que el flujo del módulo sea más completo y cercano a las necesidades reales de los usuarios.

- **Mayor intuitividad en la interfaz:** aunque el módulo ya ofrece una experiencia clara y actualmente cuenta con calendarios que permiten visualizar la disponibilidad de los recursos, a futuro se podrían enriquecer estas funciones con elementos más dinámicos, como la posibilidad de filtrar recursos por categorías o periodos de tiempo, mostrar sugerencias inteligentes de horarios menos congestionados y habilitar vistas personalizadas según el rol del usuario (propietario y solicitante o administrador). Asimismo, la incorporación de notificaciones inteligentes y paneles interactivos contribuiría a mejorar la navegación y a ofrecer una experiencia aún más fluida y atractiva para los usuarios.
- **Fortalecimiento de las conexiones entre estudiantes:** actualmente el módulo ya permite calificar los recursos utilizados, lo cual aporta a la retroalimentación entre usuarios. Como trabajo a futuro, se podría ampliar esta funcionalidad con la incorporación de un espacio tipo foro o tablón colaborativo, donde los estudiantes puedan compartir experiencias de uso más detalladas, plantear recomendaciones sobre buenas prácticas, resolver dudas frecuentes e incluso coordinar el uso compartido de recursos. Este enfoque transformaría al módulo en un punto de interacción académica y social dentro de COMA, potenciando la colaboración entre estudiantes y fomentando un sentido de comunidad alrededor de los recursos de la universidad.
- **Integración con el sistema de biblioteca:** Como trabajo futuro, una de las proyecciones más relevantes consiste en integrar el módulo de préstamos con el sistema de biblioteca de la universidad. De este modo, los usuarios podrán gestionar desde una misma plataforma tanto los recursos académicos físicos (equipos, materiales y laboratorios) como los libros y documentos de la biblioteca, ofreciendo una experiencia unificada. Esta integración permitiría simplificar los procesos, evitar la duplicidad de registros y brindar a estudiantes y docentes una visión centralizada de todos los recursos disponibles en la institución.

En conclusión, el módulo de préstamos no debe entenderse como un producto terminado, sino como un punto de partida sólido sobre el cual la Universidad Industrial de Santander puede continuar innovando y construyendo nuevas capacidades digitales al servicio de su comunidad.

### Referencias Bibliográficas

- (S/f). Amazon.com. Recuperado el 15 de septiembre de 2025, de <https://aws.amazon.com/es/what-is/javascript/>
- (S/f). Arquitecturajava.com. <https://www.arquitecturajava.com/que-es-jsp-como-se-relaciona-con-los-servlets/>
- ¿Qué Es el JavaScript? Un Repaso al Lenguaje de Programación de Scripts Más Popular de la Web.* (2021, agosto 6). Kinsta®; Kinsta. <https://kinsta.com/es/base-de-conocimiento/que-es-javascript/>
- ¿Qué es JavaScript?* (s/f). MDN Web Docs. Recuperado el 15 de septiembre de 2025, de [https://developer.mozilla.org/es/docs/Learn\\_web\\_development/Core/Scripting/What\\_is\\_JavaScript](https://developer.mozilla.org/es/docs/Learn_web_development/Core/Scripting/What_is_JavaScript)
- 06/04/, A. V. (s/f). *Creando una aplicación Java Web con Servlet.* Alura. Recuperado el 15 de septiembre de 2025, de <https://www.aluracursos.com/blog/creando-una-aplicacion-java-web-con-servlet-completo>
- Apache Software Foundation. (s. f.). *Apache Tomcat® - Welcome!*. <https://tomcat.apache.org/>
- Bootstrap: qué es, para qué sirve y cómo usarlo.* (s/f). Arsys. Recuperado el 15 de septiembre de 2025, de <https://www.arsys.es/blog/guia-completa-sobre-bootstrap>
- Calumet Estándar. (s. f.). *Estandarización de los proyectos COMA y RealTime.* <http://cormoran.uis.edu.co/eisi/Calumet/Estandar/>
- Deyimar, A. (2020). *¿Qué es Bootstrap? - Una guía para principiantes.* Tutoriales Hostinger. <https://www.hostinger.com/co/tutoriales/que-es-bootstrap>

- Editorial EIDEC. (2021). *Java básico para aprendices*. <https://editorialeidec.com/wp-content/uploads/2021/06/JAVA-BASICO-PARA-APRENDICES-ISBN-978-958-53018-8-7.pdf>
- Fol, P. (s/f). *Java basics: What is Apache Tomcat?* JRebel. Recuperado el 15 de septiembre de 2025, de <https://www.jrebel.com/blog/what-is-apache-tomcat>
- Git. (s. f.). *Git*. <https://git-scm.com/>
- Gómez Carmona, C. A., & Loaiza Correa, A. E. (2023). Diseño de módulo para préstamos académicos. Institución Universitaria de Envigado. <https://bibliotecadigital.iue.edu.co/handle/20.500.12717/3472>
- GFA Consulting Group GmbH. (s. f.). *Introduction to the Unified Modeling Language (UML)*. [https://www.gfa-group.de/web-archive/inspire/www.inspiration-westernbalkans.eu/5/9/5/3/7/7/Introduction\\_to\\_the\\_Unified\\_Modeling\\_Language\\_\\_UML\\_\\_\\_.pdf](https://www.gfa-group.de/web-archive/inspire/www.inspiration-westernbalkans.eu/5/9/5/3/7/7/Introduction_to_the_Unified_Modeling_Language__UML___.pdf)
- IBM. (s. f.). *Rational Software Architect RealTime Edition 9.5.0*. <https://www.ibm.com/docs/es/dmrt/9.5?topic=files-javascript-pages-jsp-technology>
- Java Servlet*. (2021, junio 9). Great Learning Blog: Free Resources What Matters to Shape Your Career!; Great Learning. <https://www.mygreatlearning.com/blog/java-servlet/>
- Javascript - Javascript en español*. (s/f). Lenguajejs.com. Recuperado el 15 de septiembre de 2025, de [https://lenguajejs.com/javascript/?utm\\_source](https://lenguajejs.com/javascript/?utm_source)

La respuesta, 7. 1. Introducción Contenedores de Servlets 7. 2. El Api de Servlets 7. 3. Ciclo de Vida de un Servlet Peticiones y. Threads 7. 4. La Petición 7. 55. (s/f). *Tema 7. Java Servlets*. Github.io. [https://ub-gei-sd.github.io/Tema3/Servlets.pdf?utm\\_source](https://ub-gei-sd.github.io/Tema3/Servlets.pdf?utm_source)

Larsen, R. (2013). *Beginning HTML & CSS* (1.<sup>a</sup> ed.). Wiley.

*Lenguaje JS - Javascript en español*. (s/f). Lenguajejs.com. Recuperado el 15 de septiembre de 2025, de <https://lenguajejs.com>

MDN Web Docs. (2023, 24 julio). *HTML: Lenguaje de etiquetas de hipertexto*. <https://developer.mozilla.org/es/docs/Web/HTML>

Merlo-Vega, J.-A. (2015). Libros electrónicos en las bibliotecas universitarias: recursos, servicios y 20 aspectos clave para su desarrollo. *Anuario Think EPI*, 9, 087. <https://doi.org/10.3145/thinkepi.2015.16>

MySQL. (s. f.). *Developer Zone*. <https://dev.mysql.com/>

OpenJS Foundation. (s. f.). *jQuery*. <https://jquery.com/>

Oracle. (2023, 31 enero). *Java Documentation - Get started*. <https://docs.oracle.com/en/java/>

Oracle. (s. f.). *¿Qué es una base de datos relacional?*. <https://www.oracle.com/co/database/what-is-a-relational-database/>

Pavón, J. (s. f.). *JSP (JavaServer Pages)*. Universidad Complutense de Madrid. <https://www.fdi.ucm.es/profesor/jpavon/web/44-jsp.pdf>

- Persefone20. (s. f.). *La biblia de MySQL*. WordPress. <https://persefone20.wordpress.com/wp-content/uploads/2011/09/la-biblia-de-mysql.pdf>
- Qué es el lenguaje unificado de modelado (UML). (s/f). Lucidchart. Recuperado el 19 de febrero de 2025, de <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>
- Rial, S. (2019). ¿Qué es cron y para qué sirve? Ayuda | dinahosting; dinahosting. <https://dinahosting.com/ayuda/como-configurar-tareas-cron-de-forma-manual/>
- Saavedra, J. A. (2023). *¿Qué es JavaScript y cómo funciona?* Ebac. <https://ebac.mx/blog/que-es-javascript>
- Schildt, H. (2022). *Java: The Complete Reference* (12.<sup>a</sup> ed.). McGraw Hill LLC.
- Sullivan, W. (2017). *JavaScript*. PublishDrive.
- Tomcat. (2024). *Importancia y uso en Java*. GoDaddy Resources - LATAM; GoDaddy. <https://www.godaddy.com/resources/latam/stories/que-es-tomcat>
- Universidad de Alicante. (s. f.). *JSP (JavaServer Pages)*. Repositorio Institucional de la Universidad de Alicante. <https://rua.ua.es/server/api/core/bitstreams/a742b4fe-f0c3-4beb-a48d-3daa08ba8feb/contents>
- Universidad Industrial de Santander. (s. f.). *Estándar ELISE*. Escuela de Ingeniería de Sistemas e Informática. <https://ingsistemas.uis.edu.co/eisi/Calumet/Estandar/>

Universidad Técnica del Norte. (s. f.). *HTML, CSS y JavaScript*. Biblioteca Digital UTN.

<https://bibliotecadigital.utn.edu.ec/files/original/df49ad39c8a4b8ab877f3f13c7083492a1d4e8c3.pdf>

Vaidya, N. (2019). *Servlet and JSP tutorial*. Edureka. <https://www.edureka.co/blog/servlet-and-jsp-tutorial/>

Vanier, E. (2019). *Advanced MySQL 8: Discover the full potential of MySQL and ensure high performance of your database* (1.<sup>a</sup> ed.). Packt Publishing.

Web, A. W. (s/f). *Java EE – Servlets*. Ucm.es. Recuperado el 15 de septiembre de 2025, de [https://www.fdi.ucm.es/profesor/jpavon/web/43-servlets.pdf?utm\\_source](https://www.fdi.ucm.es/profesor/jpavon/web/43-servlets.pdf?utm_source)

*What is java servlet?* (s/f). Stack Overflow. <https://stackoverflow.com/questions/7213541/what-is-java-servlet>