

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

Iván Darío Cárdenas Velásquez y Angel Ronaldo Roa Prada

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

Bucaramanga

2026

Agradecimientos

A los compañeros del grupo Calumet, gracias por su colaboración, generosidad y cercanía en cada etapa del proyecto. Su disposición para compartir conocimientos, revisar avances y mantener el ánimo en los momentos más exigentes convirtió este proceso en una experiencia profundamente humana y enriquecedora.

Nuestro especial reconocimiento al MSc. Luis Ignacio González Ramírez, director de este trabajo de grado, por su guía constante, su paciencia y la confianza depositada en nosotros. Sus observaciones oportunas y su orientación metodológica fueron fundamentales para afinar los objetivos y encaminar cada entrega con claridad y rigor.

Hacemos extensivo nuestro agradecimiento a la Escuela de Ingeniería de Sistemas y a la Universidad Industrial de Santander por el espacio académico y el acompañamiento brindado.

Agradecimientos personales del autor Angel Ronaldo Roa Prada

De manera especial, agradezco a mis padres, Dayne Omaira Prada Pereira y Fabio Enrique Roa Rodríguez, por enseñarme con su ejemplo el valor del esfuerzo, el amor y la dedicación inquebrantable. A mis hermanos, quienes han estado presentes en cada paso de este camino, brindándome compañía y aliento. Y a mi compañera de vida, Juliana Marcela Suárez Casas, cuyo apoyo constante, comprensión y fe en mí me dieron la fortaleza para superar cada dificultad y alcanzar este sueño. Este logro es tanto mío como de ellos, porque nació de su confianza y respaldo en los momentos más decisivos.

Agradecimientos personales del autor Iván Darío Cárdenas Velásquez

Este trabajo lleva impreso el nombre de las dos mujeres que sostienen mi mundo: mi madre y mi nona. A ustedes, gracias por ser el amor más puro que he conocido, por acompañarme con una entrega que no conoce condiciones y por esos sacrificios silenciosos que, sin pedir reconocimiento, marcaron cada paso de mi camino. Gracias por enseñarme que con fe, disciplina y corazón, ningún sueño es demasiado grande. Verlas sonreír sigue siendo, y siempre será, mi mayor logro.

Extiendo también mi gratitud a mi padre y a mi hermano. Su apoyo constante me dio la tranquilidad para avanzar con confianza; saber que cuento con ellos ha sido un impulso invaluable en este camino.

Gracias a mi novia por traer calma al presente y luz al futuro. Tu capacidad para creer en mí, aun cuando yo me cuestiono, es el motor que vuelve posible lo imposible. Contigo, cada paso cobra un sentido más profundo y bello.

Finalmente, a mis perritos, compañeros de amor incondicional. Su sola presencia me abraza sin necesidad de palabras. En cada regreso a casa encuentro en ustedes la prueba de que la felicidad está hecha de lealtad pura, miradas transparentes y ese cariño sencillo que sirve de refugio al alma.

Tabla de Contenido

Introducción	15
1. Planteamiento y Justificación del Problema	16
2. Objetivos	18
2.1 Objetivo General	18
2.2 Objetivos Específicos	18
3. Marco de Referencia	19
3.1 Evolución tecnológica y de desarrollo	19
3.1.1 Capa de presentación	20
3.1.2 Capa de servidor	20
3.1.3 Herramientas y prácticas de desarrollo.....	21
3.1.4 Contexto histórico, evolución y deuda técnica.....	21
3.1.5 Lecciones aprendidas y buenas prácticas adoptadas	22
3.1.6 Legado, rol del módulo de los autores y perspectivas futuras.....	22
3.1.7 Seguridad, accesibilidad y calidad.....	23
3.2 Fundamentos teóricos.....	23
3.2.1 Gestión estratégica de recursos.....	23
3.2.2 Diseño centrado en el usuario.....	24
3.2.3 Mejora y automatización de procesos	24
3.2.4 Usabilidad y experiencia del usuario.....	24
3.2.5 Metodologías ágiles.....	24
3.2.6 Adopción tecnológica y mejora continua	25
4. Metodología	25
4.1 Exploración y contextualización	25
4.2 Definición de requerimientos	25
4.3 Diseño progresivo	26
4.4 Diseño y arquitectura	26
4.5 Construcción iterativa	26
4.6 Evaluación y ajustes	26
4.7 Implantación y seguimiento	26
5. Desarrollo.....	26
5.1 Fases del desarrollo	27
5.1.1 Exploración y contextualización	27

5.1.2 Definición de requerimientos	32
5.1.2.1 HU-01 Marcar recursos como favoritos.	33
5.1.2.2 HU-02 Agregar recursos al carrito.....	35
5.1.2.3 HU-03 Registro de solicitud de préstamo de recurso	37
5.1.2.4 HU-04 Consulta del estado de una solicitud de préstamo	39
5.1.2.5 HU-05 Cancelación de una solicitud antes de aprobación.....	40
5.1.2.6 HU-06 Filtrado de recursos disponibles	42
5.1.2.7 HU-07 Visualización de detalles de un recurso	45
5.1.2.8 HU-08 Seguimiento de disponibilidad de un recurso	47
5.1.2.9 HU-09 Notificaciones por acciones del solicitante.....	49
5.1.2.10 HU-10 Evaluación de un recurso después del uso.....	51
5.1.2.11 HU-11 Evaluación del propietario del recurso	53
5.1.2.12 HU-12 Historial personal de solicitudes y asignaciones.....	56
5.1.2.13 HU-13 Validación de recursos propuestos por propietarios	58
5.1.2.14 HU-14 Gestión de categorías de recursos	60
5.1.2.15 HU-15 Reportes de auditoría del módulo	63
5.1.3 Diseño progresivo.....	65
5.1.3.1 Arquitectura	66
5.1.3.1.1 Modelo	67
5.1.3.1.2 Vista	68
5.1.3.1.3 Controlador.....	68
5.1.3.2 Modelo de Datos	69
5.1.3.3 Diagramas de interacción.....	76
5.1.3.3.1 Solicitud	76
5.1.3.3.2 Gestión de solicitudes.....	79
5.1.3.3.3 Asignaciones	80
5.1.3.4 Interfaces.....	82
5.1.3.4.1 Mockup del módulo de préstamo de recursos.....	82
5.1.3.4.2 Mockup de búsqueda y exploración de recursos.....	83
5.1.3.4.3 Mockups de favoritos y carrito.....	84
5.1.3.4.4 Mockup de solicitud de préstamo.....	86
5.1.3.4.5 Mockup de la pestaña de solicitudes	88
5.1.3.4.6 Mockup de la pestaña de asignaciones.....	90

5.1.3.4.7 Mockup de la pestaña de feedback.....	91
5.1.3.5 Construcción iterativa.....	93
5.1.3.5.1 Tarjeta del recurso.....	93
5.1.3.5.2 Pestaña Explorar.....	97
5.1.3.5.3 Pestaña Pre-solicitud.....	100
5.1.3.5.4 Pestaña Asignaciones.....	107
5.1.3.5.5 Pestaña Feedback.....	110
5.1.3.5.6 Pestañas de Favoritos y Carrito.....	113
5.1.3.5.7 Pestaña Validación.....	115
5.1.3.5.8 Pestaña de Parámetros.....	116
5.1.3.5.9 Pestaña Auditoria.....	118
5.1.3.6 Evaluación y ajustes.....	119
5.1.3.6.1 HU-01 Marcar recursos como favoritos.....	120
5.1.3.6.2 HU-02 Agregar recursos al carrito.....	122
5.1.3.6.3 HU-03 Registro de solicitud de préstamo de recurso.....	124
5.1.3.6.4 HU-04 Consulta del estado de una solicitud de préstamo.....	127
5.1.3.6.5 HU-05 Cancelación de una solicitud antes de su aprobación.....	129
5.1.3.6.6 HU-06 Filtrado de recursos disponibles.....	131
5.1.3.6.7 HU-07 Visualización de detalles de un recurso.....	133
5.1.3.6.8 HU-08 Seguimiento de disponibilidad de un recurso.....	135
5.1.3.6.9 HU-09 Notificaciones por cambios en una solicitud.....	136
5.1.3.6.10 HU-10 Evaluación de un recurso después del uso.....	137
5.1.3.6.11 HU-11 Evaluación del propietario del recurso.....	139
5.1.3.6.12 HU-12 Historial personal de solicitudes y asignaciones.....	140
5.1.3.6.13 HU-13 Validación de recursos propuestos por propietarios.....	142
5.1.3.6.14 HU-14 Gestión de categorías de recursos.....	144
5.1.3.6.15 HU-15 Reportes de auditoría del módulo.....	146
5.1.3.7 Implantación y seguimiento.....	149
6. Conclusiones.....	151
7. Recomendaciones.....	152
Referencias bibliográficas.....	154

Lista de Figuras

Figura 1. Vista de solicitudes recibidas	28
Figura 2. Vista para solicitar recursos.....	28
Figura 3. Vista para enviar un mensaje al propietario del recurso.....	29
Figura 4. Vista para asignar un recurso	29
Figura 5. Impacto en cascada que sufre una arquitectura fuertemente acoplada.....	31
Figura 6. Diagrama de secuencia para que un usuario marque o desmarque un recurso como favorito.....	34
Figura 7. Diagrama de secuencia para que un usuario marque o desmarque un recurso de su carrito	36
Figura 8. Diagrama de secuencia para el registro de una solicitud de préstamo	38
Figura 9. Diagrama de secuencia para consultar el estado de una solicitud de préstamo.....	40
Figura 10. Diagrama de secuencia para cancelar una solicitud	42
Figura 11. Diagrama de secuencia para el filtrado de recursos disponibles	45
Figura 12. Diagrama de secuencia para la visualización de los detalles de un recurso	47
Figura 13. Diagrama de secuencia para el seguimiento de la disponibilidad de un recurso....	49
Figura 14. Diagrama de secuencia para las notificaciones	51
Figura 15. Diagrama de secuencia para la evaluación de un recurso	53
Figura 16. Diagrama de secuencia para la evaluación del propietario del recurso.....	55
Figura 17. Diagrama de secuencia para el historial de solicitudes y asignaciones.....	57
Figura 18. Diagrama de secuencia para la validación de recursos	60
Figura 19. Diagrama de secuencia para la gestión de categorías.....	62
Figura 20. Diagrama de secuencia para los reportes de auditoría.....	65
Figura 21. Arquitectura MVC de referencia	66
Figura 22. Modelo entidad-relación del antiguo módulo de préstamo de recursos	70
Figura 23. Segunda versión del modelo entidad-relación pensado para el nuevo módulo.....	71
Figura 24. Tercera versión del modelo entidad-relación pensado para el nuevo módulo.	73
Figura 25. Cuarta versión del modelo entidad-relación pensado para el nuevo módulo.	74
Figura 26. Quinta versión del modelo entidad-relación pensado para el nuevo módulo.....	75
Figura 27. Diagrama de flujo de la solicitud cuando el usuario no ha interactuado previamente con los recursos.	77
Figura 28. Diagrama de flujo de la solicitud cuando el usuario ya ha interactuado previamente con los recursos.	78
Figura 29. Diagrama de flujo de la gestión de solicitudes.	80
Figura 30. Diagrama de flujo de las asignaciones.	82
Figura 31. Mockup del módulo de préstamo de recursos	83
Figura 32. Mockup de búsqueda y exploración de recursos.....	84
Figura 33. Mockup de la pestaña de favoritos del usuario.	85
Figura 34. Mockup de la pestaña de carrito del usuario.	86
Figura 35. Mockup de la pestaña de solicitud de préstamo.	88
Figura 36. Mockup de la pestaña de solicitudes.	89
Figura 37. Mockup de la pestaña de asignaciones.....	91
Figura 38. Mockup de la pestaña de feedback.....	93

Figura 39. Iconos interactivos de cada recurso.	94
Figura 40. Tarjeta del recurso para la pestaña Explorar.	94
Figura 41. Tarjeta del recurso para la pestaña Pre-solicitud.	95
Figura 42. Tarjeta del recurso para la pestaña Solicitudes.	96
Figura 43. Tarjeta del recurso para la pestaña Feedback.	96
Figura 44. Diseño del buscador de la pestaña Explorar.	97
Figura 45. Vista del filtro cuando en el criterio de búsqueda se especifica bodega.	98
Figura 46. Diseño de la bodega y sus respectivos combos.	98
Figura 47. Vista del filtro cuando en el criterio de búsqueda se especifica recurso.	99
Figura 48. Sistema de carga progresiva de recursos.	100
Figura 49. Diseño de la pestaña de Pre-solicitudes.	101
Figura 50. Diseño del formulario para la pre-solicitud de recursos.	102
Figura 51. Mensajes de retroalimentación de la pestaña Pre-solicitudes.	102
Figura 52. Vista de la pestaña Solicitudes cuando no se selecciona ningún filtro.	103
Figura 53. Vista de la información de una solicitud.	104
Figura 54. Vista de una solicitud rechazada.	105
Figura 55. Vista de una solicitud con recursos que cuentan individualmente con estado aprobado y en estudio.	105
Figura 56. Tarjetas de los recursos asociados a una solicitud con diferentes estados.	106
Figura 57. Vista de una solicitud recién asignada al usuario.	108
Figura 58. Vista de las dos opciones que tiene el usuario al momento de aprobar o rechazar una asignación.	108
Figura 59. Vista de la información de la asignación.	109
Figura 60. Vista de la información de una solicitud finalizada.	110
Figura 61. Vista de una solicitud finalizada.	111
Figura 62. Vista dentro de una solicitud finalizada de los recursos solicitados organizado por propietario.	111
Figura 63. Vista inicial que permite evaluar a un propietario.	112
Figura 64. Vista de la evaluación realizada por un solicitante a un recurso.	112
Figura 65. Chat de una solicitud entre el solicitante y el o los propietarios.	113
Figura 66. Vista de la pestaña Carrito.	114
Figura 67. Vista de la pestaña Favoritos.	114
Figura 68. Vista de un recurso que aún no se ha validado.	115
Figura 69. Vista individual de una categoría que el administrador puede editar o modificar.	117
Figura 70. Vista para agregar una nueva categoría.	117
Figura 71. Vista de una solicitud.	119
Figura 72. Vista de la pestaña Explorar donde se busca un recurso concreto y se desmarca como favorito.	121
Figura 73. Vista de la pestaña Explorar donde se busca un recurso concreto y se marca como favorito.	121
Figura 74. Vista de la pestaña Favoritos después de agregar un nuevo recurso.	122
Figura 75. Vista de la pestaña Explorar donde se busca un recurso concreto y se desmarca del carrito.	122

Figura 76. Vista de la pestaña Explorar donde se busca un recurso concreto y se marca del carrito.	123
Figura 77. Vista de la pestaña Carrito después de agregar un nuevo recurso.....	123
Figura 78. Vista desde diferentes pestañas para confirmar el uso del o los recursos.	124
Figura 79. Vista de la solicitud realizada desde la pestaña Explorar.....	125
Figura 80. Vista del formulario para realizar una solicitud.	125
Figura 81. Notificaciones del sistema cuando no se ingresa correctamente los datos.....	126
Figura 82. Solicitud creada previamente desde la pestaña Pre-solicitud.	126
Figura 83. Vista de la pestaña Solicitudes.	127
Figura 84. Vista de la información de una solicitud.	127
Figura 85. Vista de la pestaña Solicitudes con diferentes filtros de estado.	128
Figura 86. Vista de la pestaña Solicitudes, donde se muestra el ícono que permite cancelar una solicitud dependiendo de su estado.	129
Figura 87. Notificación del sistema para cancelar una solicitud.	130
Figura 88. Vista de la pestaña Auditoria donde se puede apreciar el nuevo estado de la solicitud.....	130
Figura 89. Vista de la pestaña Explorar donde se aprecia la búsqueda de un recurso.....	131
Figura 90. Vista de la pestaña Explorar para búsqueda por bodega, categoría y propietario	132
Figura 91. Vista de la tarjeta del recurso en la pestaña Explorar.....	133
Figura 92. Vista del calendario en modo lista.....	134
Figura 93. Vista del calendario en modo mes.....	134
Figura 94. Vista del calendario en modo semana.	135
Figura 95. Vista de la tabla TP_Correo.	136
Figura 96. Vista de la pestaña Feedback.....	137
Figura 97. Vista de la calificación realizada al recurso.	138
Figura 98. Vista del historial de calificaciones del recurso.	138
Figura 99. Vista de la calificación realizada al propietario del recurso.....	139
Figura 100. Vista del historial de calificaciones del propietario.	140
Figura 101. Vista de la información de una asignación.....	141
Figura 102. Vista de la pestaña Asignaciones con diferentes filtros de estado.	142
Figura 103. Vista de la pestaña Validación.	143
Figura 104. Vista de la tarjeta del recurso en cuestión y su respectivo icono de validación.	143
Figura 105. Vista de la pestaña Parámetros.....	144
Figura 106. Vista para activar y desactivar una categoría.	145
Figura 107. Vista de la pestaña Auditoria.....	147
Figura 108. Filtros de búsqueda de la pestaña Auditoria.....	148
Figura 109. Vista de las coincidencias encontradas luego de hacer una búsqueda.	149
Figura 110. Vista de los nuevos cambios agregados al formulario de pre-solicitud.	150

Lista de Abreviaturas

COMA: Comunidad Académica.

MVC: Modelo Vista Controlador.

HU: Historia de Usuario.

DAO: Data Access Object (Objeto de Acceso a Datos).

SoC: Separation of Concerns (Separación de Responsabilidades).

UCD: User Centered Design (Diseño Centrado en el Usuario).

JSP: JavaServer Pages.

JS: JavaScript.

SQL: Structured Query Language.

UIS: Universidad Industrial de Santander.

API: Application Programming Interface (Interfaz de Programación de Aplicaciones).

CRUD: Create, Read, Update, Delete (Crear, Leer, Actualizar, Eliminar).

Glosario

Actor solicitante: usuario del sistema que registra solicitudes de préstamo, realiza búsquedas de recursos, gestiona su carrito y consulta el estado de sus peticiones.

Administrador: usuario con permisos para validar recursos, administrar categorías, generar reportes de auditoría y supervisar el funcionamiento general del módulo.

Auditoría: proceso de verificación y registro de las operaciones realizadas en el sistema para asegurar la trazabilidad de las acciones.

Carrito de recursos: funcionalidad que permite agrupar varios recursos en una misma solicitud de préstamo antes de su envío.

Categoría de recurso: Clasificación que organiza los recursos disponibles según sus características o tipo de uso.

COMA: plataforma denominada Comunidad Académica, utilizada para la gestión de servicios y módulos internos de la institución.

CRUD: acrónimo de Crear, Leer, Actualizar y Eliminar, que representa las operaciones básicas en la manipulación de datos.

Historia de usuario (HU): descripción breve y estructurada de una funcionalidad desde la perspectiva del usuario final, empleada para definir requisitos en metodologías ágiles.

Metodología ágil: enfoque de desarrollo de software basado en iteraciones cortas, entregas incrementales y retroalimentación continua.

Modelo–Vista–Controlador (MVC): patrón de arquitectura de software que separa la lógica de negocio (modelo), la interfaz de usuario (vista) y el control de las interacciones (controlador).

Notificación automática: mensaje generado por el sistema para informar a los propietarios o solicitantes sobre cambios en el estado de las solicitudes.

Objeto de Acceso a Datos (DAO): componente de software que se encarga de la comunicación entre la aplicación y la base de datos, permitiendo consultas y actualizaciones.

Prototipado evolutivo: estrategia de desarrollo que consiste en construir un prototipo inicial del sistema e ir refinándolo a medida que se recibe retroalimentación.

Recurso: elemento físico o digital que puede ser solicitado en préstamo dentro de la plataforma (equipos, materiales, espacios, entre otros).

Validación de recurso: proceso mediante el cual un administrador revisa y aprueba la disponibilidad y las características de un recurso antes de que pueda ser solicitado.

Resumen

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

Autores: Angel Ronaldo Roa Prada e Iván Darío Cárdenas Velásquez**

Palabras clave: Gestión y préstamo de recursos. COMA. Experiencia de usuario. MVC. Metodologías ágiles.

Descripción: La gestión ineficiente de recursos en entornos académicos, caracterizada por procesos manuales y sistemas desarticulados, genera barreras de acceso y dificultades administrativas. El presente trabajo de grado aborda esta problemática mediante el diseño e implementación del módulo de préstamos para la plataforma COMA, enfocándose específicamente en los requerimientos del actor solicitante. El objetivo principal fue desarrollar una herramienta intuitiva que facilite el flujo de préstamo, desde la búsqueda hasta la devolución y evaluación. Para ello, se empleó una metodología ágil con prototipado evolutivo y diseño centrado en el usuario, permitiendo una validación continua de la interfaz. A nivel técnico, el sistema se construyó sobre una arquitectura Modelo-Vista-Controlador (MVC) utilizando Java (Servlets y JSP) junto con MySQL como motor de base de datos.

Como resultado, se obtuvo un módulo funcional que permite al usuario explorar el catálogo con filtros avanzados, gestionar recursos mediante listas de favoritos o un carrito de compras, además de registrar solicitudes con validación automática de disponibilidad y horarios. Adicionalmente, se implementaron mecanismos para el seguimiento de estados, cancelación de pedidos antes de su aprobación, así como un sistema de retroalimentación bidireccional para evaluar tanto los recursos como a sus propietarios. La integración de este módulo en la plataforma COMA reduce significativamente la curva de aprendizaje, mejora la transparencia en la asignación de activos y sienta las bases para una gestión de recursos más equitativa y trazable dentro de la institución.

*Trabajo de Grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Luis Ignacio González Ramírez

Abstract

Title: Design of the resource lending module in the COMA platform and development and deployment of requester-oriented requirements*

Autores: Angel Ronaldo Roa Prada and Iván Darío Cárdenas Velásquez**

Key Words: Resource management and lending; COMA; User experience; MVC; Agile methodologies.

Description: Inefficient resource management in academic environments, often characterized by manual processes and disjointed systems, creates access barriers and administrative difficulties. This thesis addresses this issue through the design and implementation of a lending module for the COMA platform, specifically focusing on the requirements of the requester actor. The main objective was to develop an intuitive tool to **facilitate** the lending flow, from search to return and evaluation. To achieve this, an agile methodology based on evolutionary prototyping and user-centered design was adopted, enabling continuous interface validation. On a technical level, the system was built upon a Model-View-Controller (MVC) architecture using Java (Servlets and JSP) with MySQL as the database engine.

As a result, a functional module was delivered that allows users to explore the catalog with advanced filters, manage resources through favorite lists or a shopping cart, in addition to registering requests with automatic validation of availability and schedules. Additionally, mechanisms were implemented for status tracking, order cancellation before approval, as well as a bidirectional feedback system to evaluate both the resources and their owners. The integration of this module into the COMA platform significantly reduces the learning curve, improves transparency in asset allocation, and lays the foundation for more equitable and traceable resource management within the institution.

*Trabajo de Grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Luis Ignacio González Ramírez

Introducción

En entornos académicos y de investigación, la disponibilidad y organización de recursos materiales son pilares esenciales para el avance del conocimiento y la formación profesional. No obstante, numerosas instituciones enfrentan obstáculos persistentes en la coordinación de estos activos, debido a métodos tradicionales de gestión que dependen de registros físicos, comunicación fragmentada y sistemas digitales desarticulados. Estas limitaciones no sólo ralentizan los procesos administrativos, sino que también generan inequidades en el acceso a herramientas críticas, afectando directamente la productividad de estudiantes, docentes e investigadores. En este marco, la implementación de soluciones tecnológicas intuitivas y especializadas emerge como una prioridad para transformar estos desafíos en oportunidades de mejora continua.

El proyecto “Diseño del módulo de préstamos de recursos en la plataforma COMA y desarrollo e implantación de los requerimientos orientados al actor solicitante” tiene como objetivo general diseñar e implantar un módulo que priorice las necesidades del solicitante, permitiéndole gestionar solicitudes de préstamo, realizar seguimiento a recursos, y evaluar tanto la calidad de los materiales como la eficiencia de los propietarios. Para ello, se desarrollarán interfaces intuitivas y flujos automatizados que simplifiquen la interacción del usuario con la plataforma, asegurando transparencia y accesibilidad. Paralelamente, se integrarán herramientas administrativas esenciales, como la validación de recursos registrados, la configuración de parámetros del sistema y la generación de registros auditables, garantizando así la operatividad, seguridad y escalabilidad del módulo.

La relevancia de esta iniciativa radica en su enfoque estratégico: al desglosar el desarrollo del módulo en dos etapas independientes —centradas en actores distintos pero complementarios—, se prioriza la profundidad sobre la amplitud, asegurando que cada

componente cumpla con estándares de usabilidad y eficiencia. Este trabajo, en particular, se concentra en empoderar al solicitante como agente activo del proceso, facilitando su capacidad para gestionar solicitudes, recibir retroalimentación y participar en la mejora continua del sistema. Además, al integrarse en COMA —una plataforma ya establecida en el entorno institucional—, se aprovecha la familiaridad existente entre los usuarios, reduciendo barreras de adopción y acelerando el impacto positivo en la comunidad.

Más allá de la automatización de tareas repetitivas, este proyecto representa un paso hacia la democratización de recursos académicos. Al eliminar fricciones en los procesos de préstamo, se promueve un uso más equitativo y transparente de los materiales, lo que, a su vez, fortalece la colaboración interdisciplinaria y el aprovechamiento de infraestructuras existentes. Mediante metodologías ágiles y un diseño centrado en el usuario, este desarrollo no solo resuelve problemáticas inmediatas, sino que sienta las bases para un ecosistema digital más cohesionado y alineado con las demandas de la educación moderna.

1. Planteamiento y Justificación del Problema

El desarrollo de este proyecto responde a la necesidad de mejorar la gestión de recursos dentro de las instituciones educativas, especialmente en laboratorios, áreas de investigación y otros espacios académicos claves. En la actualidad, la ausencia de herramientas digitales eficientes para los procesos de préstamo genera problemas como pérdida de tiempo, recursos mal gestionados y acceso limitado, lo que afecta negativamente el desarrollo académico. Por lo tanto, existe una necesidad urgente de implantar una solución funcional en el menor tiempo posible para garantizar una mayor eficiencia en estos procesos.

En este contexto, el módulo de préstamos de recursos en la plataforma COMA aborda las interacciones de tres actores principales: propietario, solicitante y administrador. Dado que el número de requerimientos para un prototipo funcional es alto, este proyecto se divide en dos trabajos de grado. Es importante destacar que, aunque ambos trabajos de grado abordan aspectos del mismo módulo, están claramente diferenciados: uno se centra en los requerimientos del solicitante y parte de los requerimientos del administrador, mientras que el otro aborda los aspectos restantes. Esta división asegura un enfoque más detallado y viable en el desarrollo de cada parte.

En ese orden de ideas, el presente proyecto se enfoca en las funcionalidades dirigidas al solicitante, quien puede realizar solicitudes de préstamo de recursos, gestionar dichas solicitudes, realizar seguimientos a recursos de interés y calificar tanto los recursos como a sus propietarios en función de la experiencia obtenida. Asimismo, se desarrollarán algunos requerimientos claves para el administrador, tales como la consulta y aval de recursos registrados, el acceso al módulo de configuración y la posibilidad de llevar una auditoría de los registros realizados. Estas funcionalidades son esenciales para garantizar el correcto funcionamiento del sistema.

La integración de este módulo en la plataforma COMA, ya conocida por los usuarios, facilitará una rápida adopción del sistema. Esto no solo mejorará la eficiencia en los procesos de préstamo, sino que también incrementará el uso de los recursos disponibles, mejorando así los procesos misionales de la academia

2. Objetivos

2.1 Objetivo General

Desarrollar e implantar los requerimientos funcionales del usuario solicitante en el módulo de préstamos de la plataforma COMA, abarcando la gestión y seguimiento de recursos y solicitudes de préstamo, así como la evaluación del propietario y sus recursos; adicionalmente, desarrollar e implantar casos específicos para el actor administrador que garanticen el correcto funcionamiento y operatividad del sistema

2.2 Objetivos Específicos

Se define como los objetivos específicos los siguientes:

- Diseñar, desarrollar e implementar las funcionalidades del actor solicitante, incluyendo:
 - Registro, cancelación y consulta de solicitudes de préstamo de recursos.
 - Exploración de recursos incluyendo métodos de búsqueda y filtrado de los mismos.
 - Etiquetar recursos como favoritos para futuras posibles solicitudes de préstamo.
 - Valoración de recursos y evaluación de los responsables de dichos recursos como parte de la experiencia del proceso de solicitud.
 - Comunicación con los propietarios de los recursos.
 - Consulta del historial de solicitudes propias.
- Diseñar, desarrollar e implementar parte de las funcionalidades del actor administrador, priorizando:
 - Consulta y validación de recursos cargados en la plataforma.
 - Configuración del módulo según necesidades operativas como lo son la creación y/o modificación de categorías y la supervisión del ciclo de vida de los préstamos,

abarcando su finalización, aprobaciones pendientes o cambios en el estado de los recursos.

- Validar el correcto funcionamiento del módulo, antes de su despliegue, mediante pruebas que se enfoquen en la experiencia del solicitante y del administrador
- Implantar el módulo de préstamos en la plataforma COMA, asegurando su despliegue en el entorno productivo.

3. Marco de Referencia

El desarrollo de aplicaciones web ha evolucionado notablemente en las últimas décadas; en ese sentido, la plataforma del Grupo Calumet (UIS) debe entenderse como el resultado de decisiones técnicas pragmáticas, tomadas en un contexto de entrega rápida y con la experiencia disponible en cada etapa. Este marco describe el recorrido desde las prácticas iniciales (como la mezcla de lógica y presentación en JSP) hasta las estrategias de evolución arquitectónica aplicadas más recientemente, así como el papel específico del módulo implementado por los autores.

3.1 Evolución tecnológica y de desarrollo

La plataforma del Grupo Calumet (UIS) ha experimentado un proceso continuo de transformación tecnológica y metodológica, en el que las decisiones iniciales, motivadas por la necesidad de entregar resultados rápidos, fueron refinadas con prácticas más estructuradas y sostenibles. Este apartado describe la progresión de las distintas capas del sistema, las herramientas y procesos de apoyo, así como los aprendizajes derivados de la deuda técnica

acumulada. Además, se exponen las buenas prácticas adoptadas, el papel del módulo implementado por los autores y las perspectivas futuras, con especial atención a la seguridad, accesibilidad y calidad de la solución.

3.1.1 Capa de presentación

La interfaz se construyó con énfasis en usabilidad y mantenibilidad, formalizando progresivamente criterios de calidad. En sus primeras versiones no siempre se respetó la separación entre presentación y lógica, situación que se corrigió de forma gradual. Se aplicaron estándares consolidados de HTML y CSS, al mismo tiempo que se adoptó *Elise.css* como guía para homogeneizar estilos. La modularización de hojas de estilo sustituyó reglas ad-hoc preexistentes.

En cuanto a la interacción, se pasó de scripting embebido en JSP a archivos JavaScript independientes, incorporando primero AJAX y posteriormente Fetch API con *async/await*, lo que mejoró la legibilidad y el control de las peticiones. Así, la lógica embebida se trasladó a ficheros JS y las llamadas al backend se organizaron de forma más controlada, reduciendo acoplamientos y facilitando pruebas.

3.1.2 Capa de servidor

La lógica de negocio se centralizó para asegurar la separación de responsabilidades, el control de acceso y la trazabilidad. En los inicios, la inserción de lógica y consultas en JSP generó acoplamientos que dificultaron la mantenibilidad y las pruebas. La transición hacia servlets y filtros permitió gestionar controladores y mecanismos transversales (autenticación, autorización, validaciones).

El código se organizó en capas bien definidas (servlets, beans, DAOs), lo que redujo la deuda técnica y clarificó el flujo de ejecución. MySQL se mantuvo como motor de persistencia por su robustez, y las consultas dispersas se centralizaron en DAOs para garantizar consistencia y control transaccional.

3.1.3 Herramientas y prácticas de desarrollo

La evolución tecnológica se apoyó en herramientas y procesos que fortalecieron la colaboración y la calidad. Durante la etapa de estabilización, entornos como NetBeans y editores como Atom facilitaron depuración y refactorización. La adopción de GitHub permitió coordinar cambios, mantener historial y realizar revisiones por pares.

La metodología de trabajo se disciplinó mediante prototipado evolutivo con entregables claros, revisiones de código y documentación constante, lo que redujo la reincidencia en prácticas deficientes.

3.1.4 Contexto histórico, evolución y deuda técnica

El origen de la plataforma respondió a la necesidad de ofrecer soluciones administrativas con plazos ajustados, lo que condujo a atajos técnicos que generaron deuda. Entre los principales problemas: mezcla de lógica y presentación en JSP, scripts embebidos, consultas dispersas, ausencia inicial de documentación y falta de convenciones, elevando la curva de aprendizaje para nuevos desarrolladores.

Las estrategias de mitigación incluyeron: extracción de lógica a servlets y servicios, centralización de acceso a datos en DAOs, externalización de activos estáticos, aplicación de filtros y controladores, establecimiento de normas de código y fortalecimiento de pruebas y

documentación. Como resultado, se redujo el acoplamiento, se mejoró la seguridad y se facilitó el mantenimiento.

3.1.5 Lecciones aprendidas y buenas prácticas adoptadas

El proceso consolidó prácticas que orientan la continuidad del proyecto:

- Mantener separación de responsabilidades (SoC).
- Evitar lógica en vistas.
- Establecer convenciones y revisiones periódicas.
- Ejecutar refactorizaciones incrementales.
- Documentar decisiones arquitectónicas.

Estas medidas permitieron transformar atajos pragmáticos en una base sostenible.

3.1.6 Legado, rol del módulo de los autores y perspectivas futuras

La plataforma constituye un activo técnico y pedagógico que ha servido como laboratorio para migraciones y mejoras. En este contexto, el módulo desarrollado por los autores se integra respetando las convenciones ya adoptadas: separación clara entre vista y lógica, uso de JS externo con AJAX/Fetch, interacción controlada con servlets y DAOs, y documentación/versionado en GitHub. Su aporte no radica únicamente en la funcionalidad entregada, sino también en la mejora incremental y en la correcta integración con el ecosistema existente. Estas prácticas facilitarán futuras migraciones y la adopción de arquitecturas más modulares.

3.1.7 Seguridad, accesibilidad y calidad

La eliminación progresiva de lógica en vistas y la validación centralizada reforzaron la seguridad; el uso de marcado semántico y estilos coherentes favoreció la accesibilidad; y la adopción de modularidad, revisiones y pruebas elevó la calidad operativa. En conjunto, la combinación de correcciones técnicas y gobernanza permitió convertir un sistema con atajos iniciales en una plataforma más mantenible, segura y alineada con buenas prácticas.

3.2 Fundamentos teóricos

El diseño e implementación del módulo de préstamos de recursos en la plataforma COMA se apoyó en un marco conceptual que integra la gestión estratégica de recursos, el diseño centrado en el usuario, la mejora de procesos, la usabilidad y las metodologías ágiles, complementado con enfoques para la adopción tecnológica y la mejora continua. Este conjunto de principios, fundamentado en literatura especializada y en la experiencia institucional, permitió que las decisiones técnicas y funcionales respondieran a necesidades reales, alineando el desarrollo con los objetivos de eficiencia, trazabilidad y satisfacción del usuario.

3.2.1 Gestión estratégica de recursos

La gestión de recursos académicos y educativos, entendida como un sistema de prácticas destinadas a garantizar la disponibilidad, control y pertinencia de los bienes institucionales (Brown & Duguid, 2000; Hoy & Miskel, 2008), constituye la base operativa del módulo. En COMA, este enfoque se materializa en un registro centralizado que ofrece visibilidad en tiempo real de la disponibilidad, características y estado de cada recurso, asegurando a la vez un historial completo de transacciones. Tal centralización, combinada con la posibilidad de adaptaciones locales, equilibra la uniformidad de criterios con la flexibilidad operativa.

3.2.2 Diseño centrado en el usuario

Sobre esta estructura se incorporó el Diseño Centrado en el Usuario, cuyo propósito es desarrollar sistemas que reflejen las necesidades, limitaciones y contextos de sus usuarios (Norman, 2019; Cooper et al., 2014). Las funcionalidades fueron organizadas para reducir pasos, ofrecer seguimiento claro de las solicitudes y mantener coherencia visual, incluso para usuarios con baja experiencia digital. El proceso se apoyó en prototipado incremental y ciclos de retroalimentación que permitieron ajustar tanto la interfaz como la lógica de interacción.

3.2.3 Mejora y automatización de procesos

La mejora y automatización de procesos se planteó no como una simple digitalización, sino como una oportunidad de rediseñar flujos, evitando la replicación de ineficiencias previas (Hammer & Champy, 1993; Davenport & Kirby, 2016). Así, el sistema realiza validaciones automáticas de disponibilidad, cambios de estado y generación de registros auditables, sin renunciar a rutas de excepción que permitan la intervención humana cuando sea necesario.

3.2.4 Usabilidad y experiencia del usuario

La usabilidad, entendida como la capacidad de un sistema para facilitar que los usuarios alcancen objetivos con eficacia, eficiencia y satisfacción (Nielsen, 1993), fue un principio transversal. Se aplicaron pautas reconocidas (Shneiderman et al., 2016) como la visibilidad de estados, la correspondencia con el mundo real y la prevención de errores, integrando elementos visuales que reducen la carga cognitiva y mejoran la experiencia global.

3.2.5 Metodologías ágiles

Para materializar estas directrices, el desarrollo se apoyó en metodologías ágiles, priorizando entregas incrementales, validaciones frecuentes y colaboración estrecha con los usuarios (Beck, 2004; Highsmith, 2009). Este enfoque permitió ajustar funcionalidades críticas

de forma temprana y evitar retrabajos costosos, manteniendo un producto en evolución constante.

3.2.6 Adopción tecnológica y mejora continua

Finalmente, la estrategia de adopción tecnológica se guió por modelos de difusión de innovaciones (Rogers, 2003; Moore, 1991), optando por un despliegue gradual que facilitara la transición desde los primeros adoptantes hacia la mayoría, mitigando resistencias y reforzando la utilidad percibida. La evaluación y mejora continua, basada en aspectos operativos, de experiencia y de gobernanza, asegura que el módulo no solo cumpla su propósito actual, sino que pueda adaptarse y evolucionar conforme cambien las necesidades institucionales y tecnológicas.

4. Metodología

Se siguió un proceso organizado y progresivo para el desarrollo, avanzando por etapas que permitieron conocer el contexto, definir los requerimientos, diseñar la solución, construirla y verificar su funcionamiento antes de la entrega.

4.1 Exploración y contextualización

Se recopiló y revisó la información disponible, incluyendo el funcionamiento del sistema existente, con el propósito de comprender el entorno, las necesidades de los usuarios y las condiciones de trabajo.

4.2 Definición de requerimientos

A partir del análisis previo se identificaron y registraron las funciones necesarias, estableciendo criterios básicos para orientar el diseño y la construcción.

4.3 Diseño progresivo

Se organizaron los requerimientos según su relevancia y se elaboraron prototipos iniciales que permitieron visualizar las principales pantallas y flujos, favoreciendo la revisión temprana.

4.4 Diseño y arquitectura

Se definió una estructura de trabajo que facilitó la organización de los componentes y la claridad en el desarrollo, considerando la continuidad del proyecto.

4.5 Construcción iterativa

Se desarrollaron las funcionalidades en etapas, realizando pruebas internas en cada avance para comprobar el correcto funcionamiento de los elementos incorporados.

4.6 Evaluación y ajustes

Se llevaron a cabo pruebas de uso y funcionamiento, tomando en cuenta las observaciones para realizar las correcciones necesarias antes de la implantación.

4.7 Implantación y seguimiento

Finalmente, se puso en marcha el módulo en el entorno correspondiente y se realizó un seguimiento inicial para verificar su operación y registrar cualquier situación que requiriera atención.

5. Desarrollo

El desarrollo de este trabajo de grado se guió por un enfoque iterativo y progresivo, orientado a la mejora continua y a la validación temprana de decisiones técnicas y funcionales. A través de ciclos de experimentación, pruebas y retroalimentación, se ajustaron los avances sin depender de fases rígidas, lo que permitió incorporar de forma oportuna observaciones tanto

del equipo técnico como de usuarios representativos. Este planteamiento coincide con las buenas prácticas de desarrollo ágil, que priorizan la adaptabilidad frente a cambios y la entrega temprana de valor (Beck et al., 2001). El uso de prototipos funcionales fue determinante para evaluar la viabilidad de soluciones, identificar limitaciones y fundamentar las decisiones sobre la evolución del sistema (Sommerville, 2016).

5.1 Fases del desarrollo

5.1.1 Exploración y contextualización

La primera etapa consistió en una revisión focalizada del intento previo de implementar el módulo de recursos y en un análisis detallado del artefacto existente (páginas JSP con *scriptlets*, accesos a datos, recursos estáticos y configuración). Esta exploración incluyó pruebas de usabilidad con perfiles representativos, lo que permitió evidenciar deficiencias significativas en la interfaz: una presentación visual poco consistente como se puede apreciar en la Figura 1 y 2, flujos confusos como se evidencia en la Figura 3, redundancia de pasos manuales y una sobrecarga cognitiva que dificultaba la interacción tal como se muestra en la Figura 4. Dichas limitaciones no solo comprometían la experiencia del usuario, sino que también suponían barreras a la adopción y riesgos de escalabilidad para la plataforma (Nielsen, 1994).

Figura 1

Vista de solicitudes recibidas

[Inicio](#) / [Comunicación](#) / [Préstamo de Recursos](#) / [Administrar Mis Recursos](#) / [Solicitudes Recibidas](#)

Administrar Mis Recursos

Solicitudes Recibidas

Ver Mis Recursos

Agregarme Recurso

Editar Mis Recursos

Asignar Mis Recursos

Ver Mis Calificaciones

Solicitar Recursos

Solicitar Recursos

Ver Mis Solicitudes

Calificar Usuario

Ver Solicitud

Generales

Nombre : null Estado : null

Nombre solicitante : Calificación solicitante : 0.0

Evento : null

Horario de solicitud

Fecha Inical : null Fecha Final : null

Hora Inical : null Hora Final : null

Observaciones

Comentario:

Figura 2

Vista para solicitar recursos

[Inicio](#) / [Comunicación](#) / [Préstamo de Recursos](#) / [Solicitar Recursos.](#) / [Solicitar Recursos](#)

Administrar Mis Recursos

Solicitudes Recibidas

Ver Mis Recursos

Agregarme Recurso

Editar Mis Recursos

Asignar Mis Recursos

Ver Mis Calificaciones

Solicitar Recursos

Solicitar Recursos

Ver Mis Solicitudes

Calificar Usuario

Escoger Recurso

Tipo de Recurso:

	NOMBRE DEL RECURSO	NOMBRE DEL PROPIETARIO	DESCRIPCION	DESCRIPCION
1	balon prestadi	David Ricardo Gómez Morgado	hoo	Elegir
2	suplemento a base de coca	Einer Steven Jaimes Mantilla	Descripción del Recurso: Agregue la descripción y si considera necesario agregue un número telefónico	Elegir
3	pa borrar	Abadmin Portal EISIWeb LuisIgnacio	descripción.	Elegir

Figura 3

Vista para enviar un mensaje al propietario del recurso

Asunto del mensaje:

Escriba su mensaje:

Fuente HTML **B** *I* ☰ ☰ ☰ ☰ ☰ ☰ ☰

NRO	NOMBRE DE USUARIO	CATEGORIA	CÓDIGO DE ESTUDIANTE	<input type="checkbox"/>
1	Einer Steven James Mantilla	Egresado	2171978	<input checked="" type="checkbox"/>

Figura 4

Vista para asignar un recurso

Ago 2025

Lu	Ma	Mi	Ju	Vi	Sa	Do
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Hoy

2025-08-08 12 : 25

SOLICITUD DE RECURSOS

Paso 1: Elija el día en el cual desea solicitar el recurso, si lo desea solicitar por varios días, seleccione la fecha desde la cual desea comenzar a utilizarlo.

Paso 2: Elija el horario en el cual desea utilizar el recurso, los horarios ya ocupados por otros usuarios no se podrán seleccionar, al elegir el horario tenga en cuenta que debe ser una franja de tiempo continua.

Nota: Si desea utilizar un recurso mas de un día, tenga en cuenta que debe ser en el mismo horario cada día, de lo contrario deberá realizar otra solicitud.

Paso 3: Toda la información sobre la solicitud aparecerá en la pantalla, y podrá elegir (si así lo desea) cuantos días desea repetir la solicitud del recurso.

HORARIO DE EL CACIQUE DE LA JUNTA PARA EL DIA VIERNES 08 DE AGOSTO DE 2025

HORA		<input type="checkbox"/>	HORA		<input type="checkbox"/>	HORA		<input type="checkbox"/>
05:00 a 05:30	<input type="checkbox"/>		11:00 a 11:30	<input type="checkbox"/>		17:00 a 17:30	<input type="checkbox"/>	
05:30 a 06:00	<input type="checkbox"/>		11:30 a 12:00	<input type="checkbox"/>		17:30 a 18:00	<input type="checkbox"/>	
06:00 a 06:30	<input type="checkbox"/>		12:00 a 12:30	<input type="checkbox"/>		18:00 a 18:30	<input type="checkbox"/>	
06:30 a 07:00	<input type="checkbox"/>		12:30 a 13:00	<input type="checkbox"/>		18:30 a 19:00	<input type="checkbox"/>	
07:00 a 07:30	<input type="checkbox"/>		13:00 a 13:30	<input type="checkbox"/>		19:00 a 19:30	<input type="checkbox"/>	
07:30 a 08:00	<input type="checkbox"/>		13:30 a 14:00	<input type="checkbox"/>		19:30 a 20:00	<input type="checkbox"/>	
08:00 a 08:30	<input checked="" type="checkbox"/>		14:00 a 14:30	<input type="checkbox"/>		20:00 a 20:30	<input type="checkbox"/>	
08:30 a 09:00	<input checked="" type="checkbox"/>		14:30 a 15:00	<input type="checkbox"/>		20:30 a 21:00	<input type="checkbox"/>	
09:00 a 09:30	<input checked="" type="checkbox"/>		15:00 a 15:30	<input type="checkbox"/>		21:00 a 21:30	<input type="checkbox"/>	
09:30 a 10:00	<input checked="" type="checkbox"/>		15:30 a 16:00	<input type="checkbox"/>		21:30 a 22:00	<input type="checkbox"/>	
10:00 a 10:30	<input type="checkbox"/>		16:00 a 16:30	<input type="checkbox"/>		22:00 a 22:30	<input type="checkbox"/>	
10:30 a 11:00	<input type="checkbox"/>		16:30 a 17:00	<input type="checkbox"/>		22:30 a 23:00	<input type="checkbox"/>	

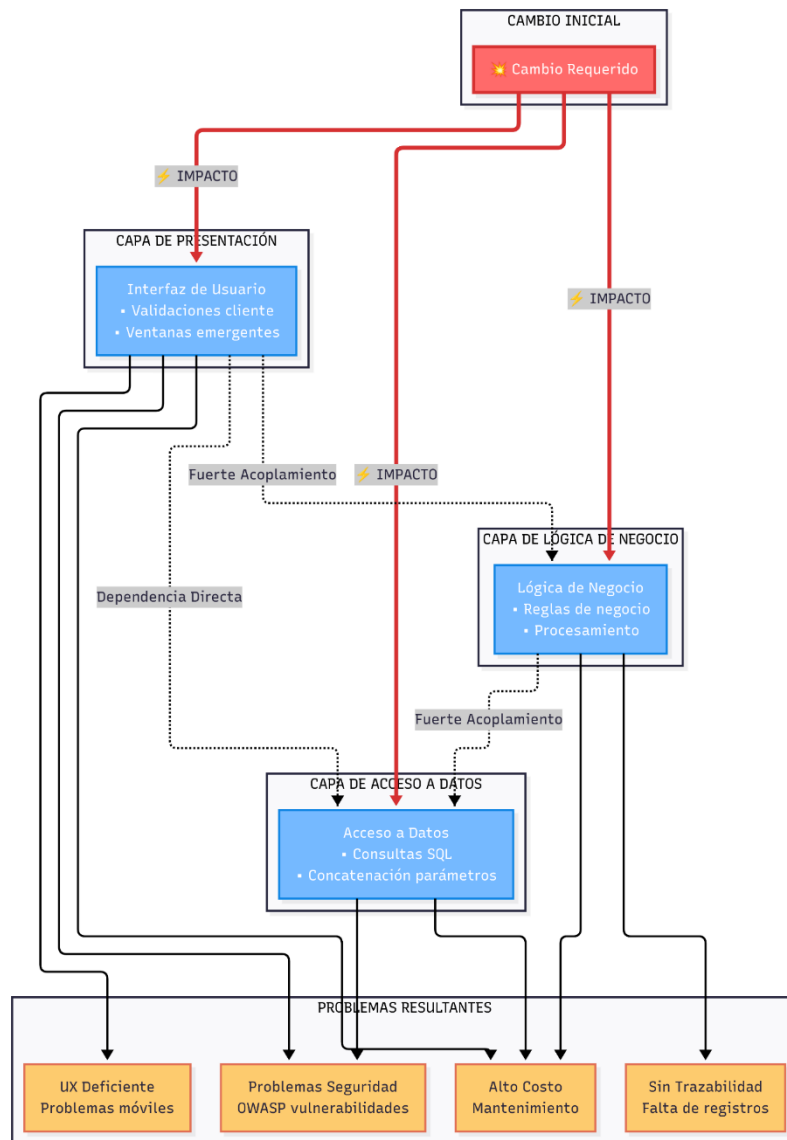
En este sentido, los hallazgos se estructuraron en categorías, acompañadas de prioridades relativas y ejemplos ilustrativos. Entre los aspectos más relevantes se identificaron:

- Acoplamiento entre presentación, lógica de negocio y acceso a datos, lo que obliga a modificar varios niveles de la aplicación ante cualquier cambio (Gamma et al., 1995).
- Prácticas inseguras en el manejo de datos, como validaciones limitadas al cliente y consultas construidas mediante concatenación de parámetros, contrarias a las recomendaciones de OWASP (2021).
- Ausencia de registros consistentes para trazabilidad e indicadores, lo que complica la gestión operativa (ISO, 2013).
- Dependencia de ventanas emergentes y procesos repetitivos, que reducen la fluidez de uso, especialmente en dispositivos móviles (Wroblewski, 2011).

La Figura 5 ilustra cómo el acoplamiento excesivo entre las capas de presentación, lógica de negocio y acceso a datos genera un efecto dominó donde cualquier cambio inicial se propaga inevitablemente a través de toda la arquitectura (flechas rojas). Las líneas punteadas representan dependencias problemáticas que violan principios de diseño limpio, resultando en cuatro consecuencias críticas: vulnerabilidades de seguridad por incumplimiento de estándares OWASP, ausencia de trazabilidad para auditoría, experiencia de usuario deficiente en dispositivos móviles, y alto costo de mantenimiento debido al esfuerzo multiplicado para implementar cambios. La visualización demuestra la necesidad urgente de refactorizar hacia un diseño desacoplado donde los cambios se contengan en capas específicas.

Figura 5

Impacto en cascada que sufre una arquitectura fuertemente acoplada



En consecuencia, estos factores combinados afectan la robustez percibida y la experiencia de usuario.

Es válido mencionar que la experiencia documentada en modernización de sistemas heredados sugiere que, en contextos similares, el desarrollo de un prototipo desde cero puede resultar más eficiente que realizar ajustes incrementales, especialmente si se adopta un enfoque de sustitución gradual como el patrón *strangler fig* (Fowler, 2004).

Bajo este criterio, se propuso como referencia técnica un prototipo mínimo que implementara únicamente los flujos críticos, con servlets como controladores, DAOs para el acceso a datos y JSP como capa de presentación. La base de datos actual se mantendría por dependencia de otros módulos, previendo compatibilidad y coordinación. El prototipado priorizaría validaciones en servidor, consultas parametrizadas, reducción de pasos y mejora para entornos móviles (Pressman & Maxim, 2020).

Las causas por las que el módulo previo no alcanzó un uso efectivo se mantienen como hipótesis: posibles requerimientos de estabilización superiores a lo estimado, incidentes de seguridad, baja aceptación por experiencia de uso o cambios en prioridades organizativas. Su validación requeriría entrevistas con actores involucrados.

5.1.2 Definición de requerimientos

A partir del diagnóstico, las necesidades se transformaron en requisitos verificables y priorizados mediante historias de usuario, siguiendo el formato “Como / Quiero / Para”, ampliamente utilizado en metodologías ágiles para mantener la trazabilidad entre la necesidad y su validación (Cohn, 2004). Estas describen funcionalidades, criterios de verificación, tareas, dependencias y evidencias mínimas para su comprobación, bajo la política de integridad de datos (los registros no se eliminan, sino que se modifica su estado para preservar su historial).

Se incluyeron mejoras de usabilidad (favoritos, carrito, filtros, tarjeta del recurso y calendario), flujos de gestión de solicitudes (registro, consulta, modificación y cancelación), validaciones y reportes, además de funciones administrativas para la gobernanza del catálogo. Las historias con mayor potencial de validación rápida fueron propuestas para prototipado, manteniendo un alcance flexible ajustable según los resultados de validaciones técnicas.

Cabe mencionar que, durante el desarrollo de las funcionalidades para el administrador, se reorientó el alcance del componente de configuración. En lugar de implementar la parametrización de avisos automáticos, se priorizó el desarrollo de una funcionalidad

de consulta y visualización consolidada de datos (HU-15). Esto permite al administrador acceder de forma centralizada a la información de los recursos, usuarios y sus respectivas calificaciones y comentarios, proporcionando una visión clara del estado de la información en el sistema antes de automatizar alertas sobre la misma.

5.1.2.1 HU-01 Marcar recursos como favoritos. Descripción: Como solicitante, quiero marcar y desmarcar recursos como favoritos para acceder rápidamente a los elementos que uso con frecuencia.

Criterios de verificación:

- Marcar y desmarcar disponible en toda sección que cuente con una tarjeta de recurso.
- Estado persistente por usuario y accesible en la pestaña *Mis favoritos*.
- No se podrá marcar como favorito un recurso que no esté disponible para el usuario.

Tareas principales:

- Crear relación usuario–recurso en la base de datos mediante cambio de estado;
- implementar endpoints para establecer, obtener y retirar favoritos;
- controles de interfaz con feedback inmediato;
- pruebas de persistencia y manejo de inactivos.

Dependencias: Esquema de usuarios y tabla de recursos.

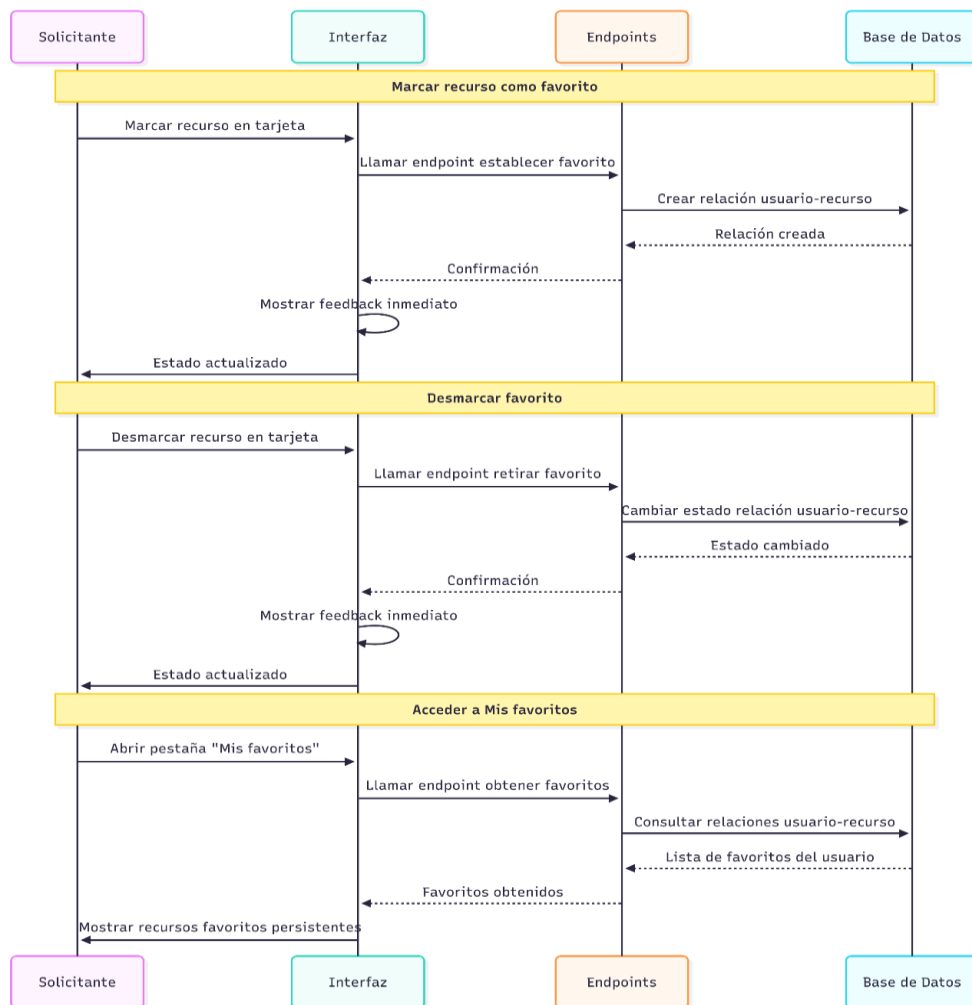
Evidencia mínima: Demo en entorno de prueba mostrando persistencia entre sesiones.

Diagrama de secuencia: La figura 6 muestra las interacciones temporales entre el solicitante y los componentes del sistema para cumplir con los criterios de verificación

establecidos. Ilustra cómo los controles de interfaz en las tarjetas de recursos invocan los endpoints correspondientes (establecer, obtener y retirar favoritos) que gestionan el cambio de estado en la relación usuario-recurso de la base de datos. Se destaca el feedback inmediato al usuario tras cada operación y la persistencia del estado accesible mediante la pestaña "Mis favoritos", garantizando que las preferencias del solicitante se mantengan entre sesiones según lo especificado en las tareas principales de implementación.

Figura 6

Diagrama de secuencia para que un usuario marque o desmarque un recurso como favorito



5.1.2.2 HU-02 Agregar recursos al carrito. Descripción: Como solicitante, quiero agregar y quitar recursos de un carrito persistente para reunir varios ítems y gestionarlos en una sola solicitud.

Criterios de verificación:

- La acción de agregar y quitar debe estar disponible en las vistas de recursos.
- El contenido del carrito debe persistir por usuario y ser accesible desde la pestaña *Mi carrito*.
- No se podrán agregar al carrito recursos que estén inactivos o que no tengan categorías activas en TR_CategoriaRecurso.

Tareas principales:

- Crear la relación usuario–recurso en la base de datos mediante TB_RecCarrito y TR_RecCarItems.
- Implementar endpoints para registrar, obtener y retirar recursos del carrito.
- Ajustar la interfaz para mostrar controles de agregado con *feedback* inmediato.
- Implementar validaciones que consulten TP_Recurso.Estado y TR_CategoriaRecurso.Estado antes de permitir la adición.
- Realizar pruebas de persistencia y recuperación de carritos entre sesiones.

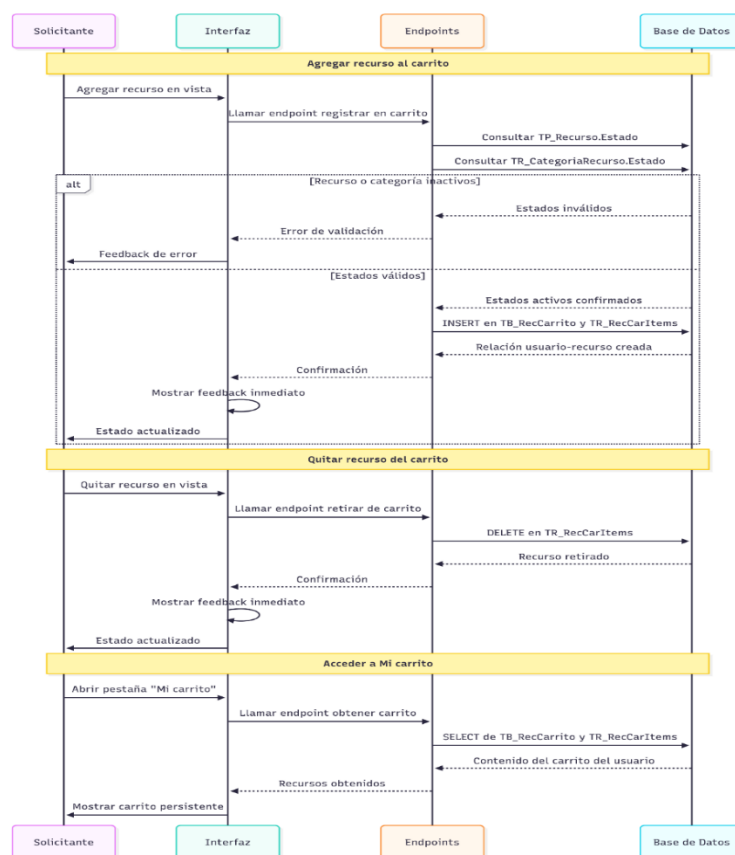
Dependencias: Tablas TP_Recurso, TR_CategoriaRecurso, TB_RecCarrito y TR_RecCarItems.

Evidencia mínima: Demo en entorno de prueba mostrando persistencia entre sesiones.

Diagrama de secuencia. La Figura 7 representa las interacciones entre el solicitante y el sistema para gestionar el carrito de recursos según los criterios establecidos. Se muestra cómo los controles de agregado en las vistas de recursos invocan los endpoints correspondientes (registrar, obtener y retirar) que validan los estados en TP_Recurso.Estado y TR_CategoriaRecurso.Estado antes de crear las relaciones en TB_RecCarrito y TR_RecCarItems. El flujo incluye el manejo de validaciones que previenen la adición de recursos o categorías inactivos, garantiza el feedback inmediato al usuario tras cada operación y demuestra la persistencia del contenido accesible mediante la pestaña "Mi carrito", cumpliendo con los requisitos de gestión centralizada de múltiples ítems para solicitudes agrupadas.

Figura 7

Diagrama de secuencia para que un usuario marque o desmarque un recurso de su carrito



5.1.2.3 HU-03 Registro de solicitud de préstamo de recurso. Descripción: Como solicitante, quiero registrar una solicitud de préstamo que incluya uno o varios recursos con fechas y datos del evento, para asegurar disponibilidad y evitar solapamientos.

Criterios de verificación:

- El registro de la solicitud debe poder iniciarse desde la pestaña Mi carrito, Mis favoritos o Explorar.
- El formulario de solicitud debe incluir: fecha de inicio, fecha de fin, nombre del evento y opción de repetición cuando aplique.
- Se debe validar la disponibilidad de los recursos en términos de cantidad y cruces de horarios antes de confirmar.
- La solicitud debe registrarse en la base de datos con estado inicial “En Estudio” en TR_UsoRecurso.

Tareas principales:

- Implementar formulario de registro de solicitudes con los campos requeridos.
- Desarrollar validaciones de cantidad y cruce de horarios con notificación clara en caso de conflicto.
- Crear el registro en TB_RecPedido y asociar recursos a la solicitud en TR_UsoRecurso.
- Establecer el estado inicial En Estudio para las solicitudes recién creadas.
- Generar una pantalla de resumen y confirmación tras el registro.

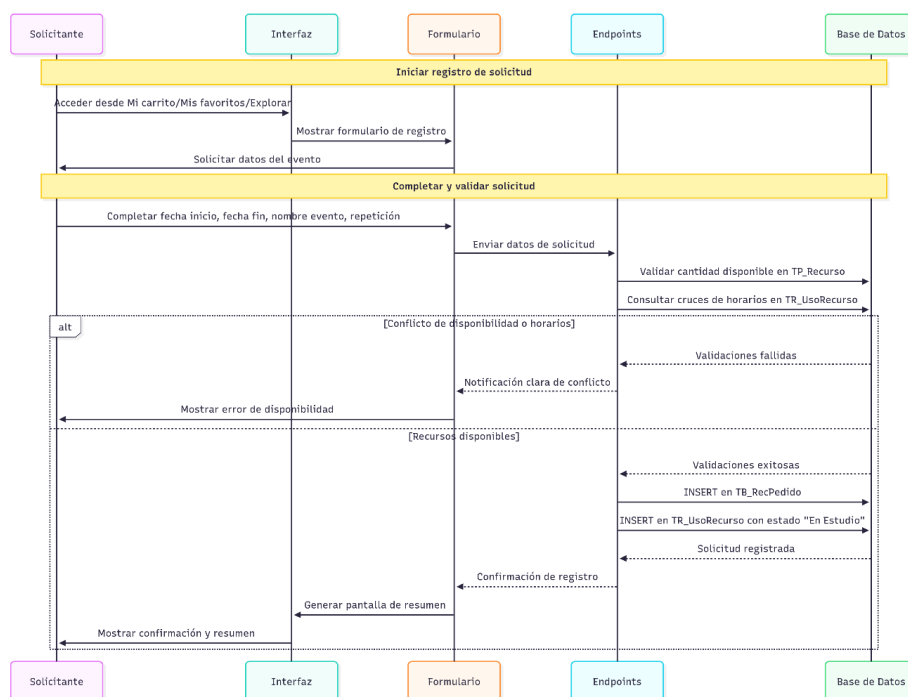
Dependencias: Tablas TP_Recurso, TR_CategoriaRecurso, TB_RecPedido, TR_UsoRecurso.

Evidencia mínima: Demo en entorno de prueba mostrando la creación de una solicitud completa con validaciones activas de disponibilidad y registro correcto en la base de datos.

Diagrama de secuencia: El diagrama ilustra el flujo completo de registro de solicitudes de préstamo desde los puntos de entrada establecidos (Mi carrito, Mis favoritos, Explorar) hasta la confirmación final. Se representa cómo el formulario de registro captura los campos requeridos (fecha inicio, fecha fin, nombre del evento, repetición) y los envía a los endpoints que ejecutan las validaciones críticas de cantidad disponible en TP_Recurso y cruces de horarios en TR_UsoRecurso. El proceso incluye el manejo de conflictos con notificación clara al solicitante, así como el registro exitoso en TB_RecPedido con la creación automática de registros en TR_UsoRecurso en estado inicial "En Estudio", culminando con la generación de una pantalla de resumen y confirmación que asegura al usuario el procesamiento correcto de su solicitud.

Figura 8

Diagrama de secuencia para el registro de una solicitud de préstamo



5.1.2.4 HU-04 Consulta del estado de una solicitud de préstamo. Descripción:

Como solicitante, quiero consultar en cualquier momento el estado de mis solicitudes para conocer si están en estudio, aprobadas, rechazadas o finalizadas.

Criterios de verificación:

- El acceso a las solicitudes debe estar disponible desde la sección *Mis solicitudes*.
- El listado debe mostrar identificador, fecha de registro, recursos asociados, rango de fechas y estado actual.
- Debe existir un filtro por estado y opción de acceder al detalle completo de la solicitud.
- Los estados deben reflejarse según los valores registrados en TR_UsosRecurso.Estado (p. ej., En Estudio, Asignado, Rechazado, Finalizado).

Tareas principales:

- Diseñar la interfaz de listados con filtros por estado.
- Implementar consultas a la base de datos filtradas por usuario solicitante.
- Construir la vista detalle con la información registrada en TB_RecPedido y los recursos vinculados en TR_UsosRecurso.
- Mostrar correctamente la trazabilidad del estado de cada solicitud.

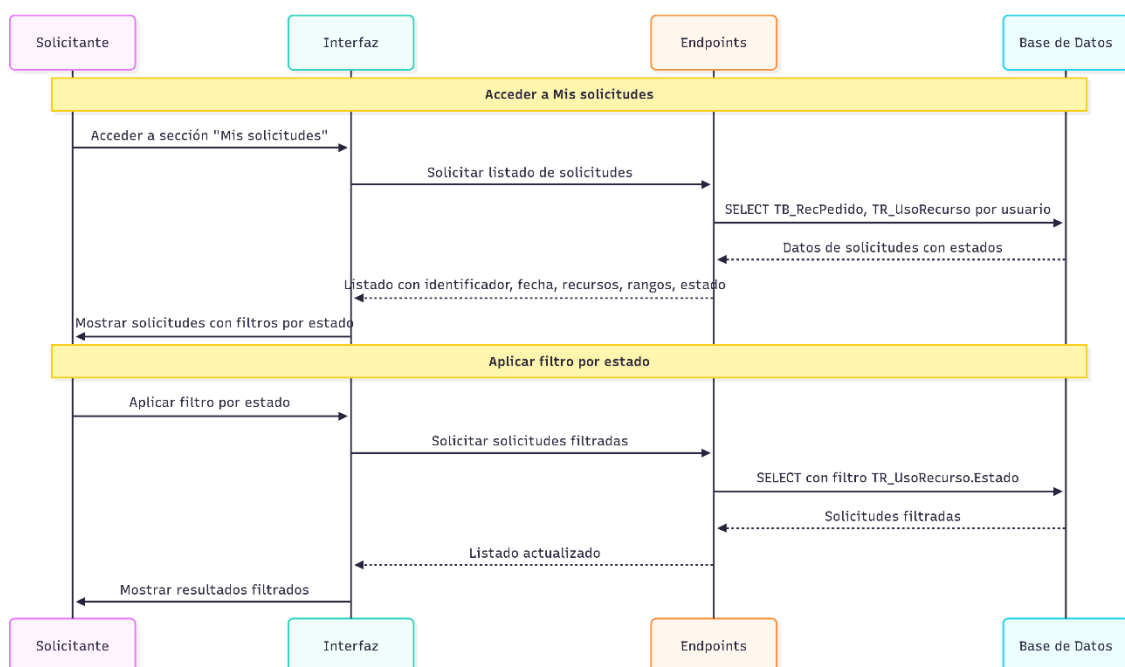
Dependencias: Tablas TB_RecPedido, TR_UsosRecurso, TP_Recurso.

Evidencia mínima: Demo en entorno de prueba mostrando la sección Mis solicitudes con datos reales, filtros por estado y acceso a la vista detalle de una solicitud con sus recursos y estados registrados.

Diagrama de secuencia. La Figura 9 representa el flujo de consulta de solicitudes desde la sección "Mis solicitudes" establecida en los criterios de verificación. Se ilustra cómo la interfaz consulta las tablas TB_RecPedido y TR_UsoRecurso filtradas por usuario solicitante para mostrar el listado con identificador, fecha de registro, recursos asociados, rango de fechas y estado actual según TR_UsoRecurso.Estado. El flujo incluye la funcionalidad de filtros por estado que actualiza dinámicamente el listado y la opción de acceder al detalle completo de cada solicitud, garantizando la correcta trazabilidad del estado y el acceso a toda la información registrada en las tablas dependientes según las tareas principales especificadas.

Figura 9

Diagrama de secuencia para consultar el estado de una solicitud de préstamo



5.1.2.5 HU-05 Cancelación de una solicitud antes de aprobación. Descripción:

Como solicitante, quiero cancelar una solicitud que aún no ha sido aprobada debido a que me arrepentí de solicitar dichos recursos.

Criterios de verificación:

- La cancelación solo estará permitida si el estado registrado en TR_UsoRecurso.Estado es En Estudio.
- El sistema debe solicitar confirmación explícita antes de ejecutar la cancelación.
- La solicitud debe quedar registrada con estado Cancelada en la base de datos, conservando el historial en TR_UsoRecurso.

Tareas principales:

- Implementar un botón de cancelación con confirmación en la vista detalle de solicitudes.
- Validar en backend que la cancelación solo se aplique sobre solicitudes en estado En Estudio.
- Actualizar el estado de la solicitud en TR_UsoRecurso.
- Garantizar que las solicitudes canceladas no aparezcan como activas en la sección Mis solicitudes, pero sí se mantengan registradas para auditoría.

Dependencias: Tablas TB_RecPedido, TR_UsoRecurso.

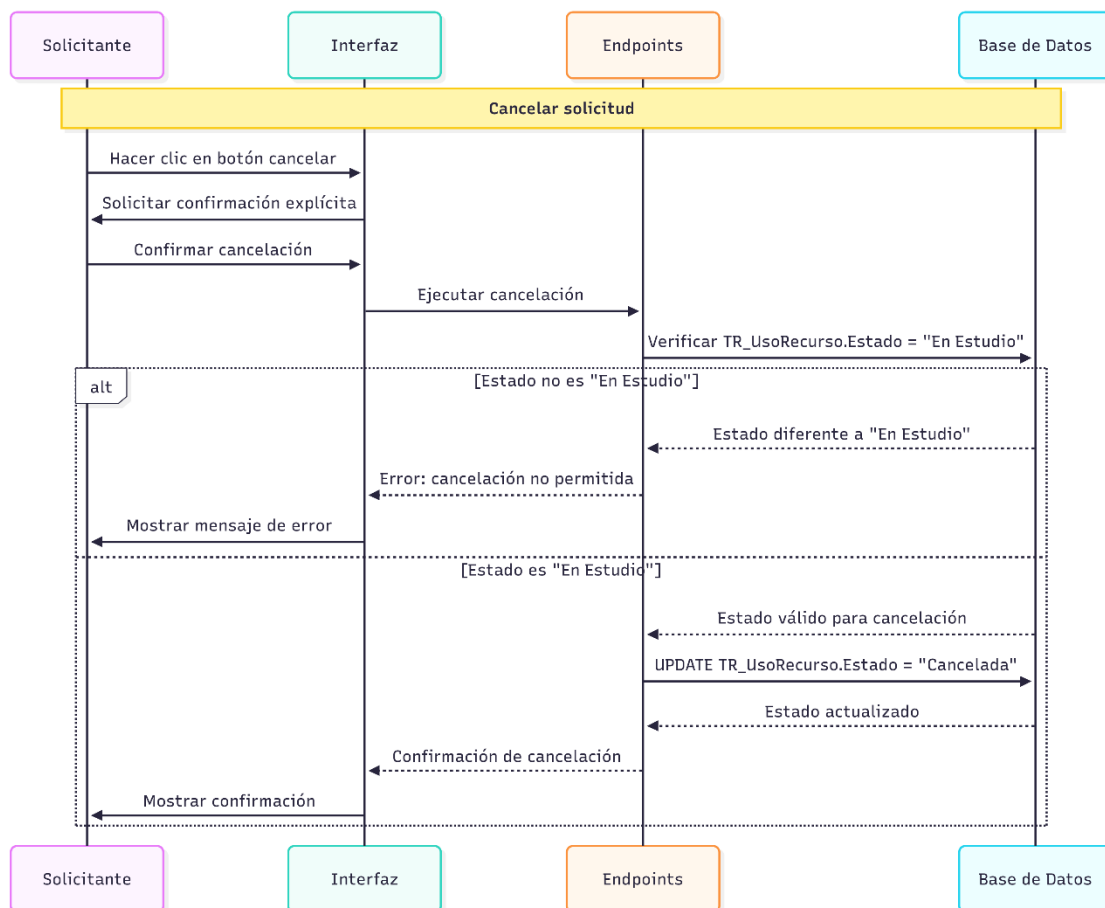
Evidencia mínima: Demo en entorno de prueba mostrando una solicitud en estado Cancelada que ya no se lista en Mis solicitudes activas, pero permanece en la base de datos con trazabilidad completa.

Diagrama de secuencia. El diagrama representa el proceso de cancelación de solicitudes desde el botón de cancelación hasta la actualización del estado en la base de datos. Se ilustra cómo el sistema solicita confirmación explícita del solicitante antes de proceder, seguido de la validación en backend que verifica el estado "En Estudio" en **TR_UsoRecurso.Estado** para

determinar si la cancelación está permitida. El flujo maneja tanto el caso de error cuando la solicitud ya no está en estado válido para cancelación, como el proceso exitoso que actualiza el registro a estado "Cancelada" en **TR_Usorecurso**, conservando el historial para auditoría según las tareas principales especificadas mientras la solicitud deja de aparecer como activa en las consultas principales.

Figura 10

Diagrama de secuencia para cancelar una solicitud



5.1.2.6 HU-06 Filtrado de recursos disponibles. Descripción: Como solicitante, quiero filtrar recursos por nombre, categoría, bodega o propietario para localizar con precisión lo que necesito.

Criterios de verificación:

- Solo se debe permitir aplicar un filtro activo a la vez entre las opciones disponibles.
- La búsqueda por nombre debe aceptar coincidencias parciales y no distinguir entre mayúsculas y minúsculas.
- La lista de recursos debe actualizarse inmediatamente al aplicar o quitar un filtro.
- Los resultados deben incluir únicamente recursos activos y con categorías activas en TR_CategoriaRecurso.

Tareas principales.

- Diseñar los controles de interfaz para aplicar filtros de nombre, categoría, bodega o propietario.
- Implementar la lógica que asegure que solo un filtro esté activo a la vez.
- Desarrollar consultas a la base de datos para cada tipo de filtro:
 - Nombre → búsqueda en TP_Recurso.Nombre y TP_Recurso.PalabrasClaves.
 - Categoría → validación en TR_CategoriaRecurso.
 - Bodega → relación en TR_RecBodItems con TB_Bodega.
 - Propietario → consulta directa al campo de propietario en TP_Recurso.

- Validar que los filtros respeten el estado del recurso (TP_Recurso.Estado) y de la categoría asociada (TR_CategoriaRecurso.Estado).

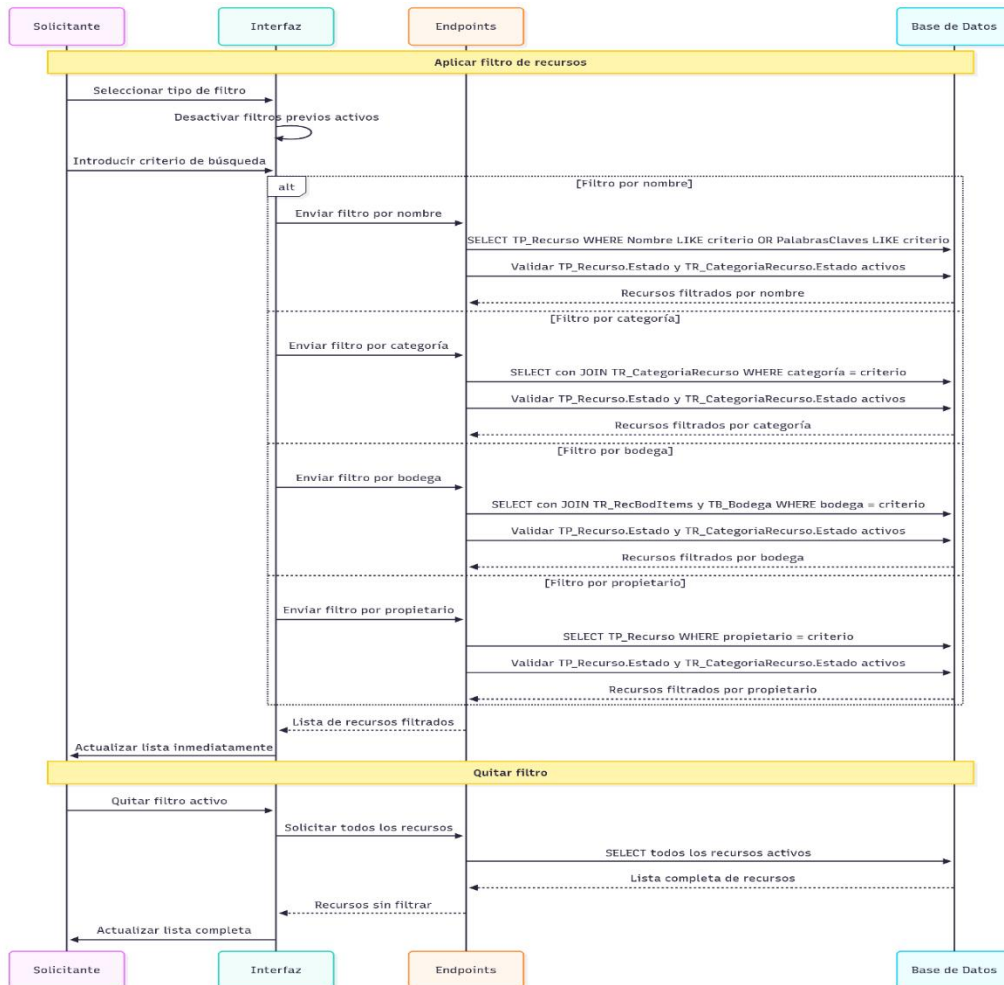
Dependencias: Tablas TP_Recurso, TR_CategoriaRecurso, TR_RecBodItems, TB_Bodega.

Evidencia mínima: Demo en entorno de prueba mostrando la aplicación de cada filtro y actualización dinámica de la lista de recursos.

Diagrama de secuencia: La Figura 11 ilustra el proceso de filtrado de recursos que garantiza la aplicación de un solo filtro activo a la vez mediante la desactivación automática de filtros previos. Se representan los cuatro tipos de filtro especificados: por nombre con búsqueda en TP_Recurso.Nombre y TP_Recurso.PalabrasClaves usando coincidencias parciales, por categoría mediante JOIN con TR_CategoriaRecurso, por bodega a través de la relación TR_RecBodItems con TB_Bodega, y por propietario consultando directamente el campo correspondiente en TP_Recurso. Cada consulta incluye las validaciones de estado requeridas en TP_Recurso.Estado y TR_CategoriaRecurso.Estado para mostrar únicamente recursos activos, con actualización inmediata de la lista tras aplicar o quitar filtros según los criterios de verificación establecidos.

Figura 11

Diagrama de secuencia para el filtrado de recursos disponibles



5.1.2.7 HU-07 Visualización de detalles de un recurso. Descripción: Como

solicitante, quiero ver la información de un recurso y que al hacer clic pueda visualizar el calendario de asignaciones para decidir si iniciar una solicitud.

Criterios de verificación:

- La tarjeta de recurso debe mostrar como mínimo el nombre y la imagen del recurso.

- Se debe integrar un calendario de solo lectura que muestre las asignaciones y préstamos confirmados de ese recurso.
- Toda la información debe obtenerse en tiempo real desde la base de datos.

Tareas principales:

- Diseñar la vista de un recurso con datos de TP_Recurso.
- Consultar en TR_UsoRecurso los rangos de fecha aprobados/confirmados asociados al recurso, para mostrarlos en el calendario.
- Implementar el calendario en modo solo lectura con bloques de tiempo reservados/ocupados.

Dependencias: Tablas TP_Recurso, TR_CategoriaRecurso, TR_RecBodItems, TB_Bodega, TR_UsoRecurso.

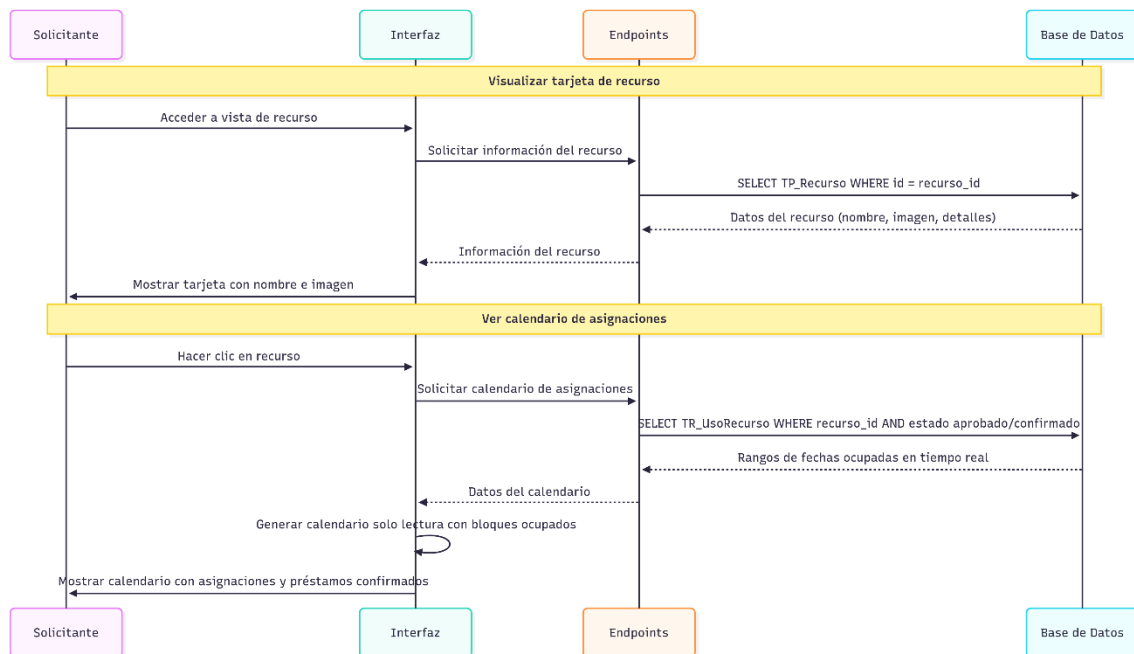
Evidencia mínima: Demo en entorno de prueba mostrando la ficha de un recurso con datos reales y un calendario donde se reflejen las asignaciones y reservas registradas.

Diagrama de secuencia: La Figura 12 representa la visualización de detalles completos de un recurso en dos fases según los criterios establecidos. Primero, muestra cómo se obtienen y presentan los datos básicos del recurso (nombre e imagen) desde TP_Recurso en la tarjeta inicial. Posteriormente, al hacer clic en el recurso, se consultan en tiempo real los rangos de fechas aprobados y confirmados desde TR_UsoRecurso para generar el calendario de solo lectura que muestra las asignaciones y préstamos confirmados con bloques de tiempo reservados u ocupados. El flujo garantiza que toda la información se obtenga en tiempo real desde la base de datos, permitiendo al solicitante visualizar la disponibilidad del recurso para

tomar decisiones informadas sobre el inicio de nuevas solicitudes según las tareas principales especificadas.

Figura 12

Diagrama de secuencia para la visualización de los detalles de un recurso



5.1.2.8 HU-08 Seguimiento de disponibilidad de un recurso. Descripción: Como solicitante, quiero consultar la disponibilidad de un recurso mediante un calendario para identificar con anticipación fechas libres.

Criterios de verificación:

- La vista de disponibilidad debe ser accesible al dar clic a la tarjeta del recurso.
- El calendario debe mostrar de forma clara los periodos ocupados (asignaciones aprobadas y préstamos en curso) y los libres.

- Esta vista es únicamente de consulta: no se podrán modificar reservas ni iniciar solicitudes directamente desde el calendario.
- La información debe obtenerse en tiempo real de las asignaciones registradas en la base de datos.

Tareas principales:

- Implementar un componente de calendario en la ficha del recurso para mostrar disponibilidad.
- Consultar en TR_UsoRecurso los periodos de uso aprobados o en curso asociados al recurso.
- Procesar y representar gráficamente en el calendario los intervalos ocupados vs. los libres.
- Garantizar que el calendario sea de solo lectura y no permita ninguna acción sobre las reservas.

Dependencias: Tablas TP_Recurso, TR_UsoRecurso.

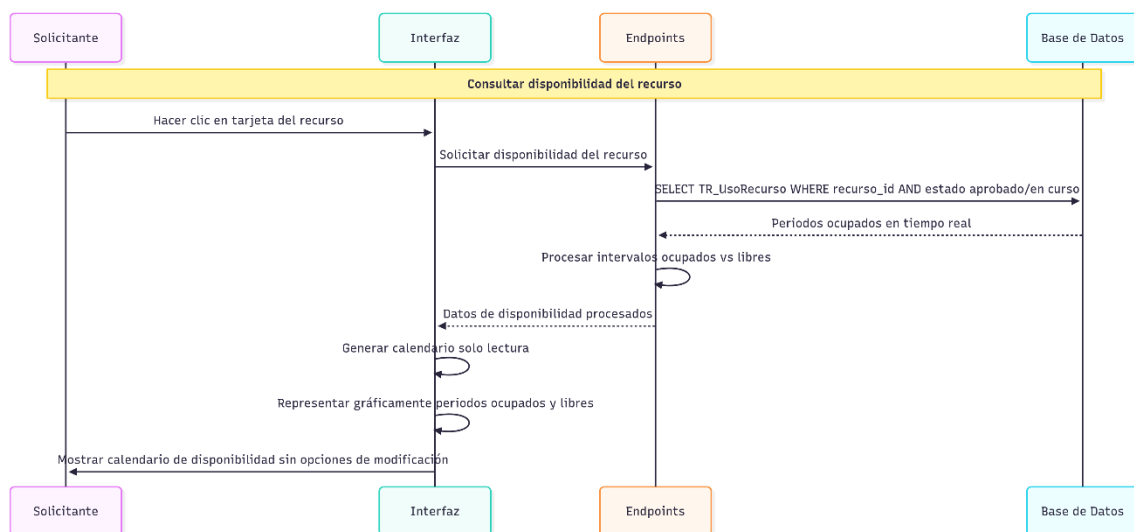
Evidencia mínima: Demo en entorno de prueba mostrando un calendario con los periodos ocupados y libres reflejados correctamente según los datos actuales en la base de datos.

Diagrama de secuencia: La Figura 13 representa el proceso de consulta de disponibilidad de un recurso mediante calendario de solo lectura. Se ilustra cómo al hacer clic en la tarjeta del recurso, el sistema consulta en tiempo real los periodos ocupados desde TR_UsoRecurso filtrando por asignaciones aprobadas y préstamos en curso, procesa los

intervalos para distinguir entre periodos ocupados y libres, y genera un calendario que representa gráficamente esta información de forma clara. El flujo garantiza que el calendario sea únicamente de consulta sin permitir modificaciones o acciones sobre las reservas, cumpliendo con los criterios de verificación que establecen esta funcionalidad como una herramienta informativa para que el solicitante pueda identificar con anticipación las fechas libres del recurso según los datos actuales de la base de datos.

Figura 13

Diagrama de secuencia para el seguimiento de la disponibilidad de un recurso



5.1.2.9 HU-09 Notificaciones por acciones del solicitante. Descripción: Como propietario de un recurso, quiero recibir una notificación por correo cuando un solicitante realice acciones sobre mis recursos (registro, modificación o cancelación de solicitudes), para estar informado y dar el trámite correspondiente.

Criterios de verificación:

- Cada vez que se registre, modifique o cancele una solicitud en TR_UsoRecurso, debe generarse un correo automático dirigido al propietario del recurso.

- El mensaje debe incluir como mínimo: identificador de la solicitud, nombre del evento (si aplica), recursos involucrados y la acción realizada por el solicitante.
- El correo debe enviarse de forma inmediata tras la acción y llegar a la dirección registrada en el perfil del propietario en TP_Usuario.

Tareas principales:

- Configurar disparador en backend que detecte acciones relevantes del solicitante en TR_UsoRecurso.
- Integrar el servicio de correo para generar y enviar notificaciones al propietario.
- Definir plantilla mínima de correo con datos clave de la solicitud y la acción.
- Pruebas de envío y recepción en diferentes escenarios (registro, modificación, cancelación).

Dependencias: Tablas TR_UsoRecurso (solicitudes), TP_Usuario (correo del propietario), servicio de correo.

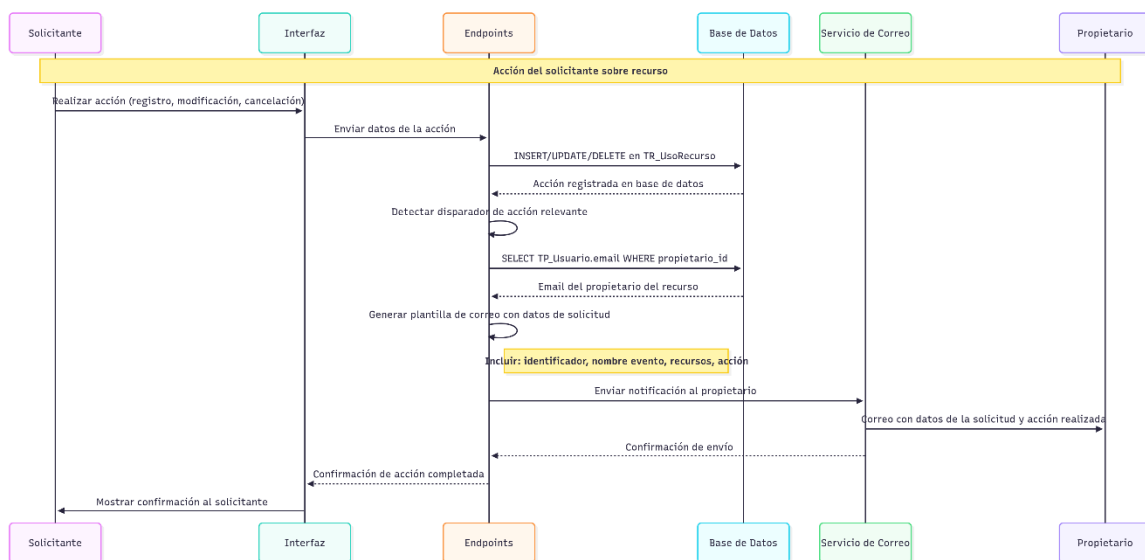
Evidencia mínima: En entorno de prueba, al registrar, modificar o cancelar una solicitud, el propietario recibe un correo en su cuenta con los datos de la solicitud y la acción realizada por el solicitante.

Diagrama de secuencia: La Figura 14 representa el flujo de notificaciones automáticas que se genera cuando un solicitante realiza acciones sobre recursos de otros propietarios. Se ilustra cómo cada operación en TR_UsoRecurso (registro, modificación o cancelación) activa un disparador en el backend que consulta el email del propietario desde TP_Usuario, genera una plantilla de correo con los datos mínimos requeridos (identificador de solicitud, nombre

del evento, recursos involucrados y acción realizada), y envía inmediatamente la notificación a través del servicio de correo. El flujo garantiza que el propietario reciba información en tiempo real sobre las acciones de los solicitantes en sus recursos, permitiéndole dar el trámite correspondiente según los criterios de verificación establecidos, mientras que el solicitante recibe confirmación de que su acción fue procesada correctamente.

Figura 14

Diagrama de secuencia para las notificaciones



5.1.2.10 HU-10 Evaluación de un recurso después del uso. Descripción: Como solicitante, quiero registrar una evaluación del recurso al finalizar el préstamo, para compartir mi experiencia y contribuir a su mantenimiento.

Criterios de verificación:

- La opción de evaluación debe estar disponible solo si el préstamo asociado en TR_UsorRecurso está en estado Finalizado.

- El formulario debe requerir una calificación obligatoria (ejemplo: escala 1 a 10) y permitir un comentario opcional.
- Solo debe aceptarse una evaluación del recurso por préstamo.
- Las evaluaciones deben quedar almacenadas y consultables en el historial del recurso.

Tareas principales:

- Implementar un formulario de evaluación con validaciones (obligatoriedad de la calificación, comentario opcional).
- Habilitar el acceso al formulario únicamente para préstamos en estado Finalizado y sin evaluación previa.
- Crear relación en la base de datos que vincule la evaluación con TR_UsoRecurso, TP_Recurso y TP_Usuario.
- Ajustar vistas de historial para mostrar evaluaciones registradas.

Dependencias: Tablas TR_UsoRecurso (préstamos), TP_Usuario (solicitante), TP_Recurso (recurso).

Evidencia mínima: Demo en entorno de prueba donde un préstamo finalizado permite evaluación, la calificación queda registrada y es visible en el historial del recurso.

Diagrama de secuencia: La Figura 15 representa el proceso de evaluación de recursos post-uso con validaciones estrictas de elegibilidad. Se ilustra cómo el sistema verifica que el préstamo esté en estado "Finalizado" en TR_UsoRecurso y que no exista una evaluación previa antes de habilitar el formulario, garantizando una evaluación única por préstamo. El flujo

incluye la validación de la calificación obligatoria y el manejo del comentario opcional, seguido del registro en base de datos vinculando la evaluación con TR_Usorecurso, TP_Recurso y TP_Usuario. Adicionalmente, se representa la funcionalidad de consulta del historial que permite visualizar las evaluaciones almacenadas del recurso, cumpliendo con los criterios de verificación que aseguran la integridad de las evaluaciones y su disponibilidad para contribuir al mantenimiento y mejora de los recursos según la experiencia compartida por los solicitantes.

Figura 15

Diagrama de secuencia para la evaluación de un recurso



5.1.2.11 HU-11 Evaluación del propietario del recurso. Descripción: Como

solicitante, quiero evaluar al propietario del recurso tras completar el préstamo, para reflejar la experiencia en atención y cumplimiento.

Criterios de verificación.

- La opción de evaluación estará disponible únicamente cuando el préstamo en TR_UsoRecurso esté en estado Finalizado.
- El formulario debe requerir una calificación obligatoria y permitir un comentario opcional.
- Solo podrá registrarse una evaluación por préstamo y propietario.
- La evaluación debe asociarse al propietario.

Tareas principales:

- Implementar formulario de evaluación del propietario con validaciones (calificación obligatoria, comentario opcional).
- Habilitar el acceso al formulario únicamente para préstamos finalizados.
- Crear relación en la base de datos que vincule evaluación con TR_UsoRecurso, TP_Usuario (solicitante) y el propietario registrado en TP_Recurso.

Dependencias: Tablas TR_UsoRecurso, TP_Recurso, TP_Usuario.

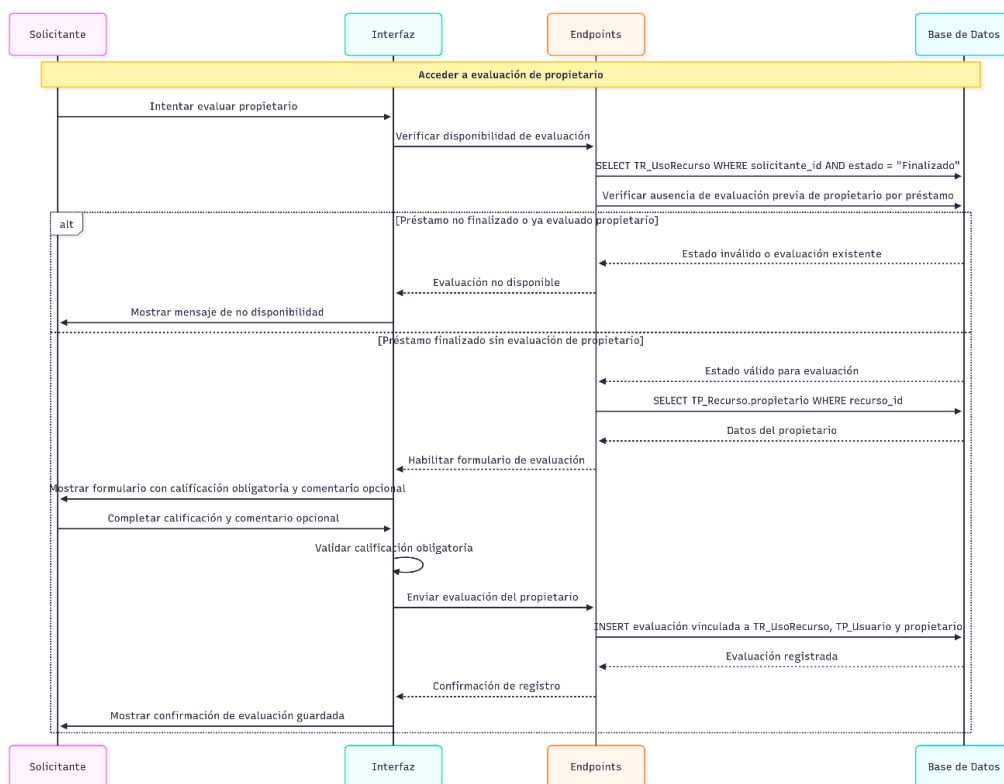
Evidencia mínima: En pruebas, un préstamo finalizado permite registrar una evaluación del propietario; esta queda almacenada y accesible para otros usuarios de acuerdo con permisos definidos.

Diagrama de secuencia: La Figura 16 representa el proceso de evaluación del propietario tras completar un préstamo con validaciones paralelas a las de evaluación del recurso. Se ilustra cómo el sistema verifica que el préstamo esté en estado "Finalizado" en

TR_UsoRecurso y que no exista una evaluación previa del propietario para ese préstamo específico, garantizando una evaluación única por préstamo y propietario. El flujo incluye la identificación del propietario desde TP_Recurso, la validación de la calificación obligatoria con comentario opcional, y el registro en base de datos vinculando la evaluación con TR_UsoRecurso, TP_Usuario (solicitante) y el propietario correspondiente. Esta funcionalidad permite a los solicitantes reflejar su experiencia en atención y cumplimiento del propietario, contribuyendo a un sistema de retroalimentación bilateral que mejora la confianza y calidad del servicio según los criterios de verificación establecidos.

Figura 16

Diagrama de secuencia para la evaluación del propietario del recurso



5.1.2.12 HU-12 Historial personal de solicitudes y asignaciones. Descripción: Como solicitante, quiero acceder a un historial consolidado de todas mis solicitudes y asignaciones, para revisar procesos pasados y contar con respaldo histórico.

Criterios de verificación:

- El usuario puede acceder al historial desde el menú principal.
- El listado incluye todos los registros en TR_UsoRecurso vinculados al solicitante, con al menos: identificador de la solicitud, fechas de registro e inicio/fin, recursos involucrados, propietario y estado final.
- El historial debe permitir aplicar filtros por estado (ej. En Estudio, Aprobada, Rechazada, Cancelada, Finalizada).

Tareas principales:

- Diseñar sección de historial con controles de filtro y ordenamiento.
- Implementar consultas a la base de datos que recuperen las solicitudes del usuario desde TR_UsoRecurso.
- Validar consistencia entre solicitudes, recursos y estados finales.

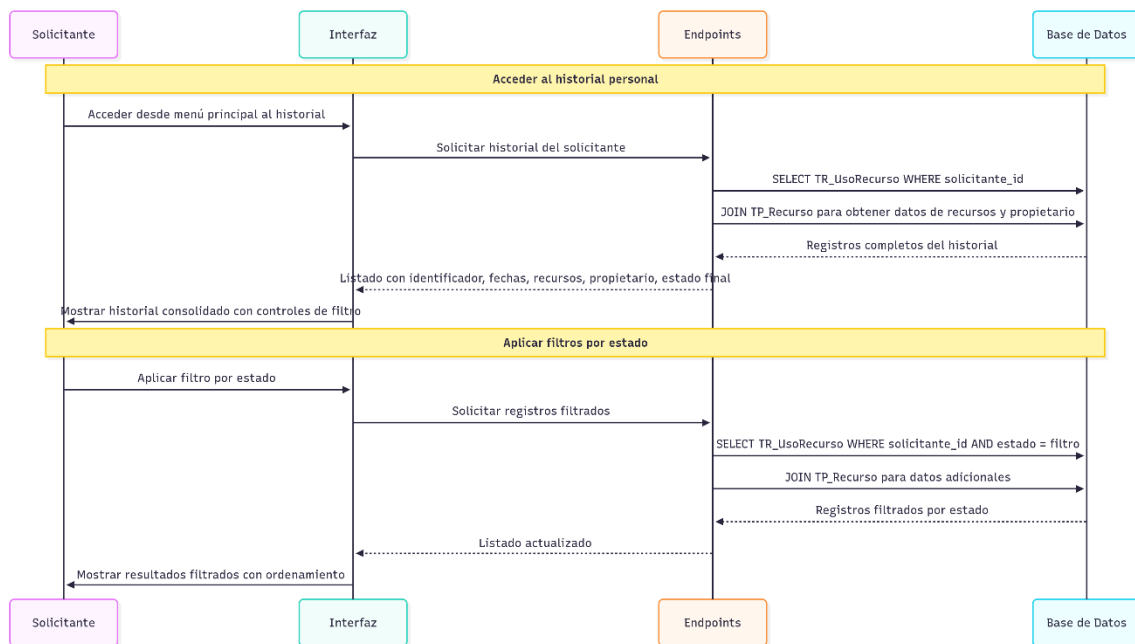
Dependencias: Tablas TR_UsoRecurso, TP_Recurso, TP_Usuario.

Evidencia mínima: En pruebas, el historial de un usuario carga registros reales, los filtros funcionan correctamente y se puede acceder al detalle con la trazabilidad de cada solicitud.

Diagrama de secuencia: La Figura 17 representa el acceso al historial personal consolidado desde el menú principal con recuperación completa de solicitudes y asignaciones del solicitante. Se ilustra cómo el sistema consulta todos los registros vinculados al usuario en TR_Usorecurso mediante JOIN con TP_Recurso para obtener la información completa requerida: identificador de solicitud, fechas de registro e inicio/fin, recursos involucrados, propietario y estado final. El flujo incluye la funcionalidad de filtros por estado que permite al solicitante refinar la consulta según criterios específicos (En Estudio, Aprobada, Rechazada, Cancelada, Finalizada) con actualización dinámica del listado. Esta funcionalidad proporciona al usuario un respaldo histórico completo con trazabilidad de todos sus procesos pasados, cumpliendo con los criterios de verificación establecidos para la revisión y consulta de su actividad histórica en el sistema.

Figura 17

Diagrama de secuencia para el historial de solicitudes y asignaciones



5.1.2.13 HU-13 Validación de recursos propuestos por propietarios. Descripción:

Como administrador, quiero marcar la validación de recursos propuestos por propietarios, de forma individual, para identificar cuáles han sido revisados sin alterar su disponibilidad.

Criterios de verificación:

- La validación se gestiona desde el área de administración.
- El listado de recursos (TP_Recurso) debe mostrar un campo de validación visible junto al resto de atributos.
- El administrador puede marcar o desmarcar la validación de manera individual, recurso por recurso.
- Los cambios de validación deben persistir en la base de datos y ser reflejados en futuras consultas.

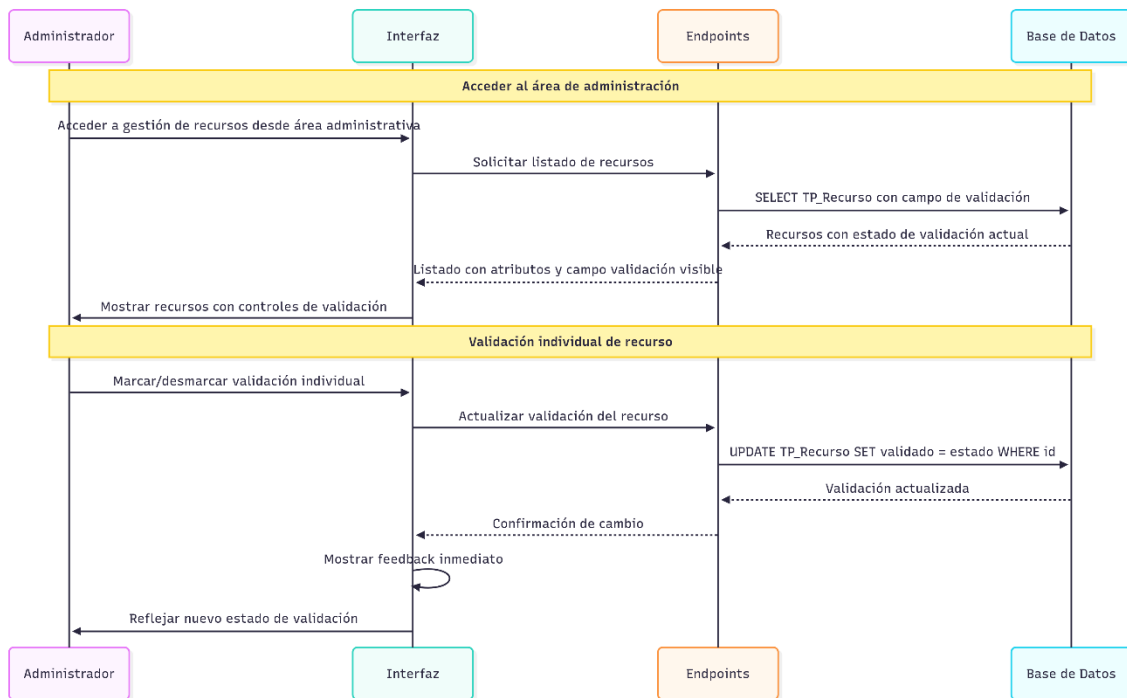
Tareas principales:

- Añadir un campo de validación en el modelo TP_Recurso (ej. Validado = Sí/No).
- Construir interfaz administrativa con controles individuales (ej. toggles o checkbox por recurso).
- Implementar endpoints para actualizar la validación de un recurso de manera individual.
- Garantizar persistencia en la base de datos y feedback inmediato en la interfaz.

Dependencias: Tablas TP_Recurso, TP_Usuario.

Evidencia mínima: En pruebas, el administrador marca o desmarca la validación sobre un recurso propuesto de manera individual, y los cambios quedan almacenados en la base de datos, visibles en listados posteriores.

Diagrama de secuencia: La Figura 18 representa la gestión administrativa de validación individual de recursos propuestos por propietarios desde el área de administración. Se ilustra cómo el administrador accede al listado de recursos en TP_Recurso con el campo de validación visible junto al resto de atributos, permitiendo identificar cuáles han sido revisados sin alterar su disponibilidad. El flujo se enfoca exclusivamente en la validación individual recurso por recurso mediante controles como toggles o checkbox, con actualización inmediata en la base de datos y feedback visual en la interfaz. Los cambios persisten y son reflejados en futuras consultas, cumpliendo con los criterios de verificación que permiten al administrador gestionar de manera precisa y controlada la revisión de cada recurso propuesto mientras mantiene un registro persistente del estado de validación individual.

Figura 18*Diagrama de secuencia para la validación de recursos*

5.1.2.14 HU-14 Gestión de categorías de recursos. Descripción: Como administrador, quiero gestionar categorías de recursos para mantener una clasificación ordenada que facilite búsquedas y administración.

Criterios de verificación:

- El administrador puede crear, editar y desactivar categorías mediante cambio de estado.
- El listado de categorías debe incluir nombre, descripción y estado (activo/inactivo).
- Las categorías inactivas permanecen en la base de datos y en el historial, pero no aparecen como opción en nuevos registros de recurso.

- El sistema debe prevenir nombres duplicados al crear o editar categorías.

Tareas principales:

- Implementar CRUD de categorías en la interfaz administrativa sobre la tabla TP_CategoriaRecurso.
- Añadir campo de estado (activo/inactivo) para controlar disponibilidad sin pérdida de historial.
- Validar unicidad de nombre en la base de datos.
- Ajustar vistas de recursos y formularios para que solo muestren categorías activas.

Dependencias: Tabla TP_CategoriaRecurso y todas las vistas/funciones que consumen categorías en TP_Recurso.

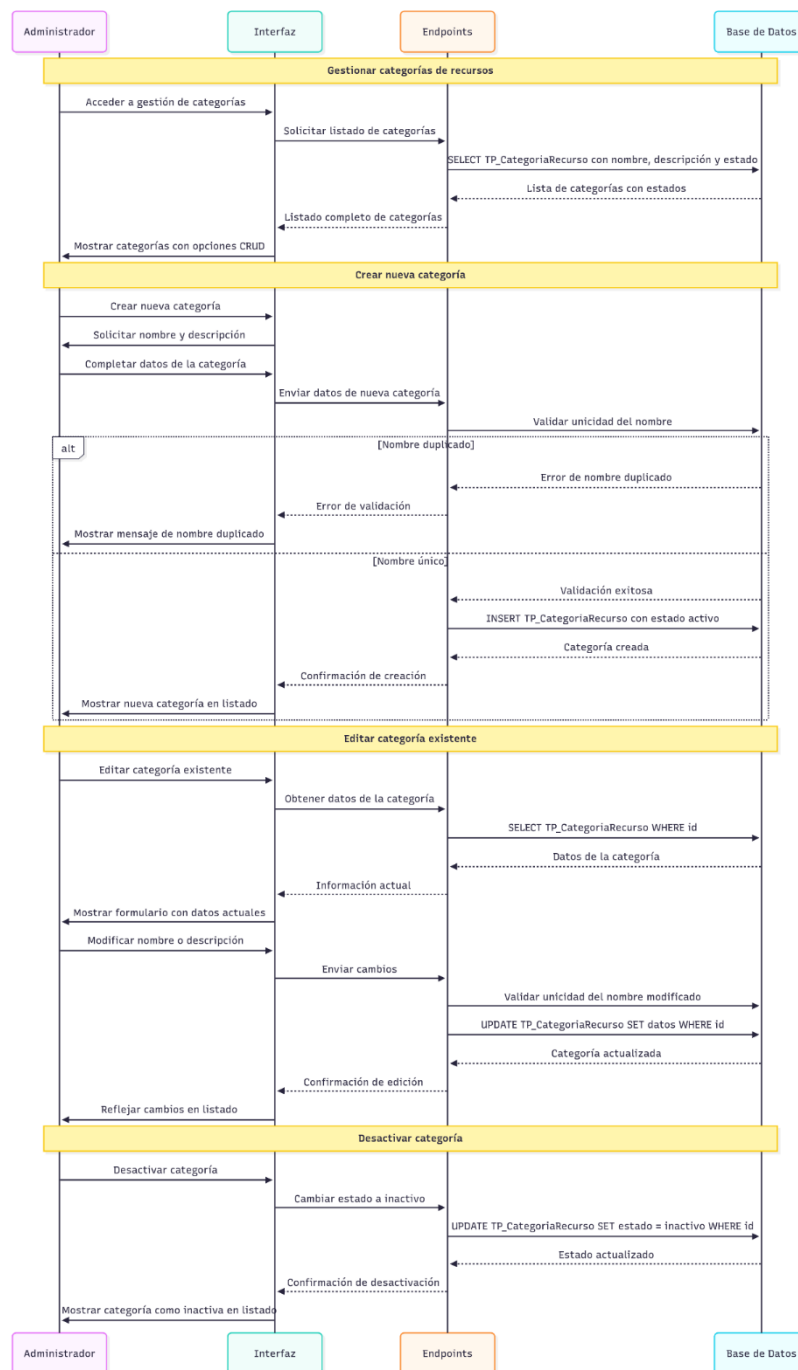
Evidencia mínima: En pruebas, el administrador puede crear, modificar y desactivar categorías; categorías inactivas no se muestran en formularios nuevos, pero sí permanecen visibles en registros históricos.

Diagrama de secuencia: La Figura 19 representa la gestión completa de categorías de recursos mediante operaciones CRUD desde la interfaz administrativa. Se ilustra cómo el administrador accede al listado de categorías en TP_CategoriaRecurso mostrando nombre, descripción y estado, seguido de los procesos de creación con validación de unicidad de nombres para prevenir duplicados, edición de categorías existentes manteniendo las mismas validaciones, y desactivación mediante cambio de estado que preserva el historial sin eliminar registros. El flujo garantiza que las categorías inactivas permanezcan en la base de datos para mantener la integridad referencial con registros históricos, pero no aparezcan como opciones

disponibles en nuevos registros de recursos, cumpliendo con los criterios de verificación que aseguran una clasificación ordenada y controlada que facilite búsquedas y administración del sistema.

Figura 19

Diagrama de secuencia para la gestión de categorías



5.1.2.15 HU-15 Reportes de auditoría del módulo. Descripción: Como administrador, quiero acceder a un reporte consolidado por solicitud que reúna la información clave registrada para entender el contexto de cada caso y facilitar el seguimiento administrativo.

Criterios de verificación:

- Acceso desde el área de administración a un reporte consolidado por cada solicitud registrada en TB_RecPedido.
- Cada registro incluye: fecha y hora, usuario solicitante, estado actual de la solicitud, y datos de los recursos vinculados desde TR_UsoRecurso y TP_Recurso.
- Se deben incluir también contenidos asociados como comentarios o calificaciones si existen.
- El reporte debe contar con varios filtros.

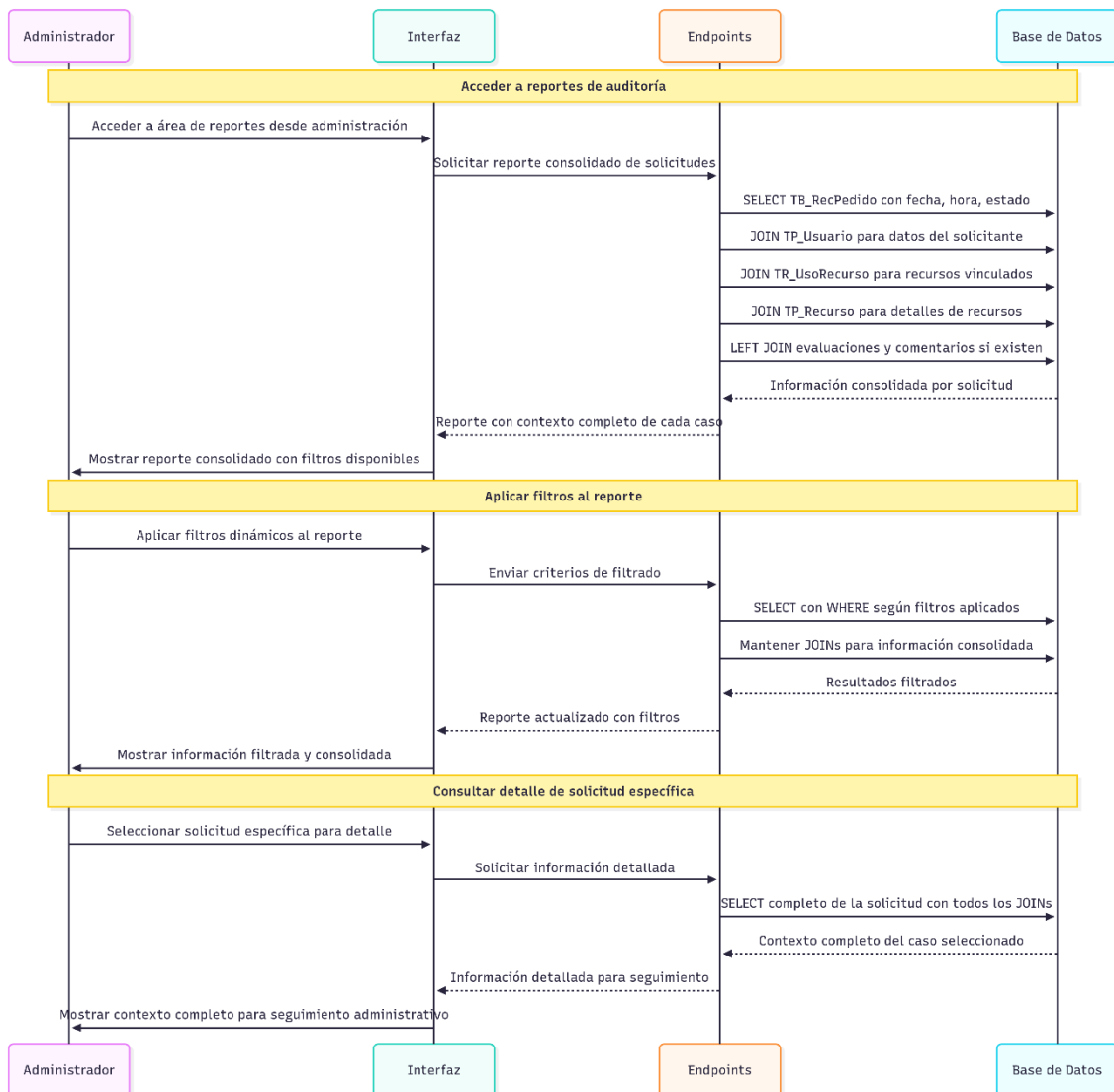
Tareas principales:

- Implementar vista de reporte de auditoría en el área administrativa.
- Consultas que consoliden información de TB_RecPedido, TR_UsoRecurso, TP_Recurso y TP_Usuario.
- Añadir filtros dinámicos.

Dependencias: Tablas TB_RecPedido, TR_UsoRecurso, TP_Recurso y TP_Usuario.

Evidencia mínima: En pruebas, el administrador accede al reporte, aplica filtros y visualiza información consolidada de cada solicitud, con datos coherentes a lo registrado en la base de datos.

Diagrama de secuencia: La Figura 20 representa el acceso a reportes de auditoría consolidados desde el área administrativa para facilitar el seguimiento de cada solicitud registrada. Se ilustra cómo el sistema consulta TB_RecPedido mediante múltiples JOINS con TP_Usuario, TR_UsosRecurso y TP_Recurso para consolidar información clave: fecha y hora, usuario solicitante, estado actual y datos de recursos vinculados, incluyendo contenidos asociados como comentarios o calificaciones cuando existen. El flujo incluye la aplicación de filtros dinámicos que permiten al administrador refinar la consulta según criterios específicos, manteniendo la integridad de la información consolidada, y el acceso al detalle completo de casos específicos para entender el contexto y realizar seguimiento administrativo efectivo, cumpliendo con los criterios de verificación establecidos para la auditoría y control del módulo.

Figura 20*Diagrama de secuencia para los reportes de auditoría*

5.1.3 Diseño progresivo

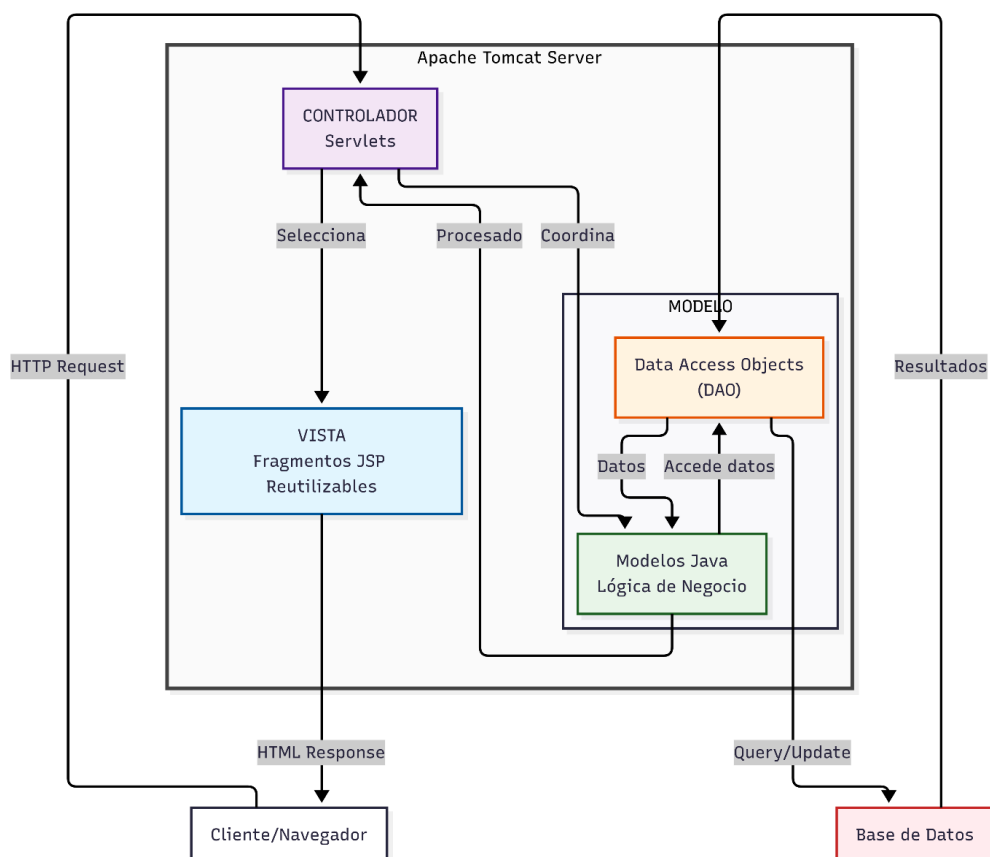
El diseño del sistema se abordó como un proceso adaptable y evolutivo, en coherencia con el enfoque iterativo planteado en la metodología general. En lugar de definir un modelo rígido y cerrado desde el inicio, se optó por elaborar un núcleo conceptual mínimo que garantizara coherencia y sirviera de base para incorporar mejoras de manera gradual. Esta

estrategia permitió mantener la alineación con los requerimientos priorizados y reducir en la medida de lo posible el tener que replantear componentes ya diseñados.

5.1.3.1 Arquitectura. La arquitectura de referencia se estructuró siguiendo el patrón Modelo–Vista–Controlador (MVC) ver Figura 21, el cual permite organizar de manera clara y modular los componentes del sistema. Este enfoque enfatiza la separación de responsabilidades, de modo que el Modelo gestiona la lógica de negocio y los datos, la Vista se encarga de la presentación y la interacción con el usuario, mientras que el Controlador actúa como intermediario coordinando la comunicación entre ambos. Gracias a esta división, se facilita la escalabilidad, la reutilización de componentes y el mantenimiento del sistema en el tiempo.

Figura 21

Arquitectura MVC de referencia



5.1.3.1.1 Modelo. En la capa de Modelo se implementaron los elementos encargados de representar el dominio del sistema y de gestionar la lógica de negocio. Esta capa se organizó en tres componentes principales: entidades, DAOs y capa de lógica de negocio.

En primer lugar, se definieron las clases de entidad, que reflejaban la estructura de las tablas de la base de datos. Estas clases servían como objetos de transferencia, permitiendo manipular la información en memoria de manera orientada a objetos y manteniendo una correspondencia clara entre el modelo lógico y el modelo físico de datos.

El acceso a la base de datos se resolvió mediante el patrón Data Access Object (DAO). Cada DAO encapsuló las operaciones CRUD (crear, leer, actualizar y eliminar) sobre una entidad específica, lo que permitió desacoplar los detalles de persistencia del resto de la aplicación. Gracias a esta abstracción, el código de la lógica de negocio no dependía de consultas SQL ni de la tecnología de persistencia utilizada, lo que favoreció la modularidad y la mantenibilidad.

Por encima de los DAOs se estableció una capa de lógica de negocio, encargada de coordinar la interacción entre la persistencia y el Controlador. En ella se consolidaron las reglas propias del dominio: se aplicaron validaciones, se ejecutaron cálculos y se orquestaron múltiples DAOs cuando una operación requería combinar información de varias entidades. De esta manera, el Controlador podía invocar métodos de alto nivel sin tener que gestionar directamente el acceso a datos ni las reglas específicas del dominio.

En conjunto, la organización del Modelo bajo esta estructura permitió mantener una separación clara entre persistencia, lógica de negocio y presentación, lo que garantizó un flujo

de información consistente, facilitó la escalabilidad y contribuyó a la reutilización de componentes en futuras extensiones del sistema.

5.1.3.1.2 Vista. La capa de Vista tuvo como propósito presentar la información al usuario y gestionar la interacción con la aplicación. Para su implementación se emplearon JavaServer Pages (JSPs), que permitieron generar contenido dinámico en función de los datos enviados desde la capa de Controlador.

Cada JSP estuvo asociado a una funcionalidad específica del sistema, mostrando formularios, tablas de resultados o mensajes informativos según correspondiera. El Controlador, tras procesar la solicitud y obtener los datos necesarios de la capa de lógica de negocio, los enviaba como atributos al JSP correspondiente, el cual se encargaba de representarlos de manera comprensible para el usuario final.

Con el fin de mantener la separación entre lógica y presentación, en las JSPs se evitó la inclusión de código complejo de negocio. En su lugar, se recurrió a etiquetas estándar de JSP y a expresiones simples para mostrar datos, dejando el procesamiento a cargo de las capas inferiores. Esta práctica favoreció la claridad del código y facilitó su mantenimiento a largo plazo.

Asimismo, la organización de las JSPs permitió una navegación estructurada y coherente dentro del sistema, asegurando una experiencia de usuario consistente y alineada con los objetivos de la arquitectura MVC.

5.1.3.1.3 Controlador. La capa de Controlador se implementó con el objetivo de gestionar la interacción entre la interfaz de usuario (Vista) y la lógica de negocio (Modelo). Para ello, se utilizaron Servlets, los cuales funcionaron como puntos de entrada a la aplicación

y fueron responsables de recibir las solicitudes HTTP, procesarlas y devolver una respuesta adecuada.

Cada Servlet se encargó de interpretar las peticiones provenientes de la Vista, extraer los parámetros necesarios y delegar la ejecución de la operación solicitada a la capa de lógica de negocio. De esta forma, se evitó que la Vista tuviera que interactuar directamente con los DAOs o con la persistencia, manteniendo la separación de responsabilidades definida por el patrón MVC.

Adicionalmente, los Servlets implementaron mecanismos de control de flujo, como la redirección o el reenvío hacia las páginas correspondientes según el resultado de la operación. Este manejo centralizado permitió uniformar la gestión de las solicitudes y simplificar la navegación dentro del sistema.

En conjunto, la capa de Controlador actuó como mediadora, asegurando que la comunicación entre la Vista y el Modelo se mantuviera clara y estructurada, lo que favoreció la mantenibilidad y la extensibilidad de la aplicación.

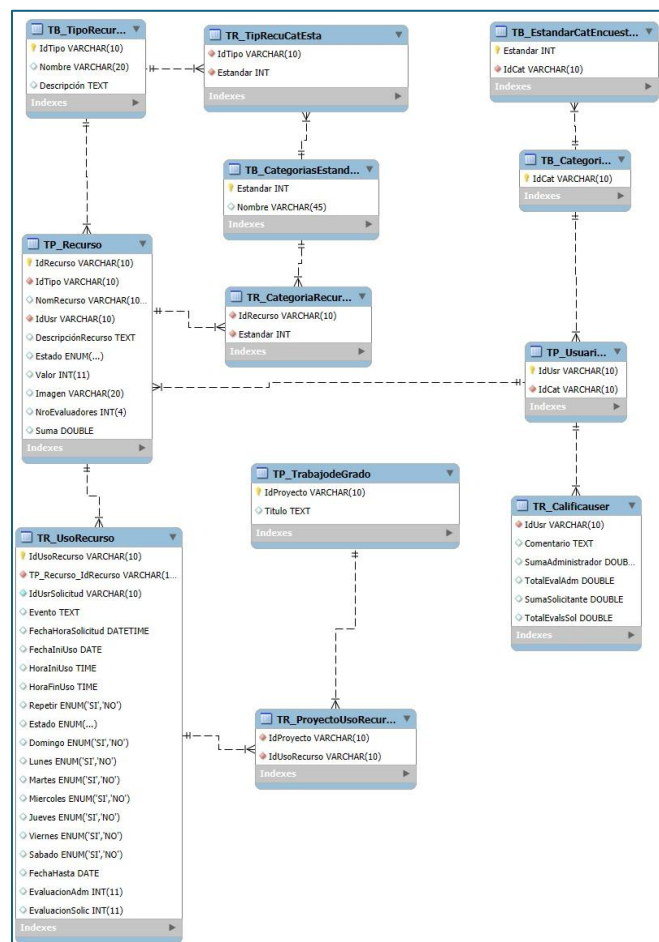
5.1.3.2 Modelo de Datos. El modelo de datos se consolidó a partir de sucesivas iteraciones sobre la base Diamante, manteniendo las estructuras críticas para garantizar la interoperabilidad con otros módulos institucionales y extendiéndolas de acuerdo con las necesidades identificadas en las historias de usuario. En este proceso se retomó un modelo entidad-relación previamente diseñado para soportar el intento previo del módulo de préstamo de recursos. Este esquema inicial sirvió como punto de partida, pero como se puede apreciar en la Figura 22 también evidenció limitaciones significativas: no contemplaba llaves foráneas que aseguraran la integridad referencial, mostraba rigidez frente a la incorporación de nuevas funcionalidades como favoritos, chat, carrito de recursos o foros de solicitudes, y contenía tablas con responsabilidades poco claras.

Un ejemplo de ello fue la tabla TB_TipoRecurso, concebida para clasificar los recursos, pero empleada además como referencia para definir visibilidad hacia distintos perfiles de usuario, tarea que ya era atendida por otra tabla específica. Esta superposición de funciones generaba inconsistencias conceptuales y dificultaba la evolución del modelo. En conjunto, estas restricciones revelaron la necesidad de un rediseño que no solo preservara la interoperabilidad institucional, sino que también aportara claridad en la definición de entidades y relaciones, mayor flexibilidad para extender el sistema y la capacidad de responder a los escenarios planteados en las historias de usuario.

Figura 22

Modelo entidad-relación del antiguo módulo de préstamo de recursos

Nota. Desarrollado por el grupo Calumet.

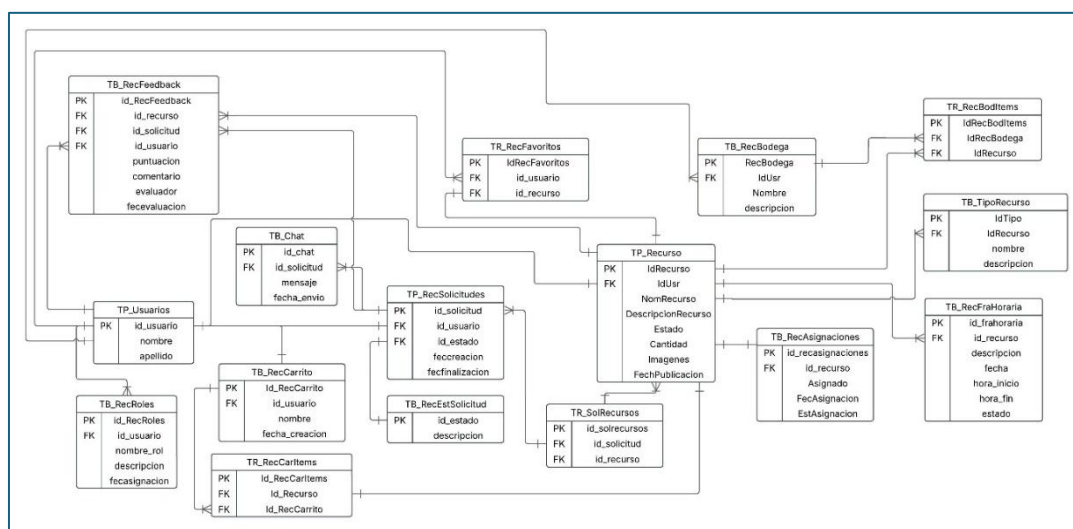


A partir de las limitaciones identificadas en el modelo heredado, se planteó un nuevo esquema entidad-relación que se puede apreciar en la Figura 23, fundamentado en buenas prácticas de diseño de bases de datos. Este modelo buscaba corregir las inconsistencias conceptuales, garantizar la integridad referencial desde su concepción y ofrecer la flexibilidad necesaria para incorporar funcionalidades como favoritos, carritos de recursos, foros de solicitudes y administración descentralizada mediante bodegas. En esencia, se trataba de un rediseño más cercano a lo que finalmente se consolidaría, con una estructura clara y relaciones bien definidas.

No obstante, tras ser evaluado en conjunto por el equipo y los responsables del sistema, se concluyó que su implementación no era viable. El principal reto radicaba en la compatibilidad con los demás componentes de la plataforma institucional, ya que la magnitud de los cambios requeridos implicaba un alto costo de integración y el riesgo de afectar estructuras críticas necesarias para la interoperabilidad. Por esta razón, aunque el modelo representaba una propuesta técnicamente sólida, fue descartado de manera consensuada.

Figura 23

Segunda versión del modelo entidad-relación pensado para el nuevo módulo



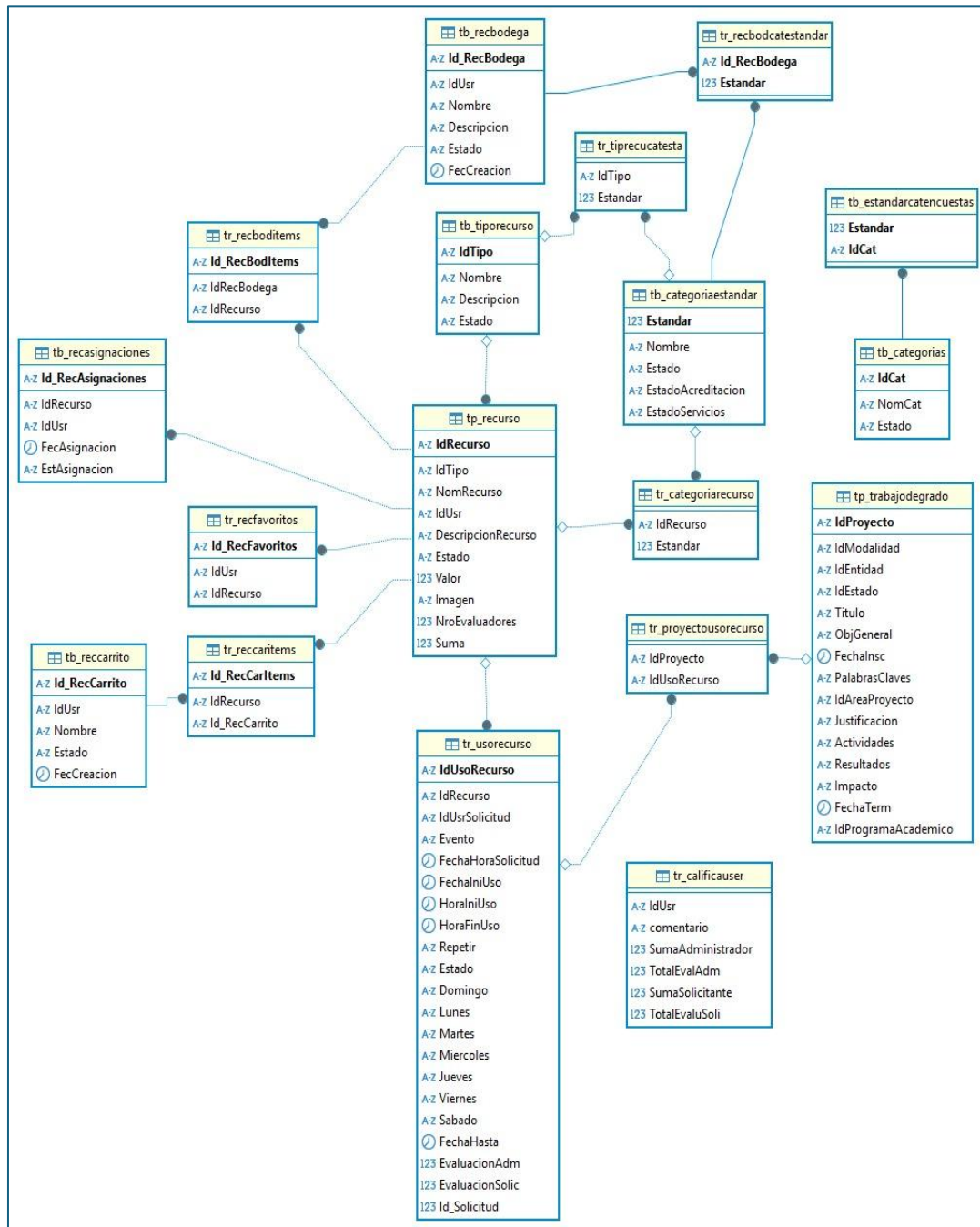
Tras el descarte del modelo alternativo, se decidió retomar el esquema inicial como base para un nuevo prototipo. La intención fue aprovechar las estructuras heredadas que ya garantizaban la interoperabilidad, pero al mismo tiempo extenderlas progresivamente para cubrir las necesidades levantadas en las historias de usuario. Desde el comienzo, el desarrollo del modelo se ha abordado mediante un prototipado evolutivo, en el que cada iteración introduce ajustes y ampliaciones a partir del análisis colectivo y de la validación con los distintos actores involucrados.

En el marco de estas iteraciones, en la Figura 24 se puede apreciar un prototipo más completo, que incorpora nuevas entidades para dar soporte a funcionalidades emergentes. Sin embargo, durante las sesiones de revisión con el equipo se advirtió que el modelo empezaba a mostrar un crecimiento mayor al esperado en el número de tablas. Aunque no se trataba de una proliferación excesiva, sí implicaba una complejidad adicional que se buscaba evitar. Algunas de estas tablas asumían funciones muy específicas o duplicaban en parte responsabilidades ya contempladas en otras estructuras, lo que podía comprometer la claridad y la simplicidad deseadas para el modelo. Este hallazgo llevó a replantear el diseño con un criterio de equilibrio: mantener la riqueza funcional necesaria para cubrir los casos de uso, pero evitando un nivel de fragmentación que dificultara la implementación y el mantenimiento futuro.

En este proceso también se prestó especial atención al fortalecimiento de las relaciones, lo que implicó un trabajo constante en la definición de llaves foráneas como mecanismo para asegurar coherencia e integridad en la estructura general.

Figura 24.

Tercera versión del modelo entidad-relación pensado para el nuevo módulo.

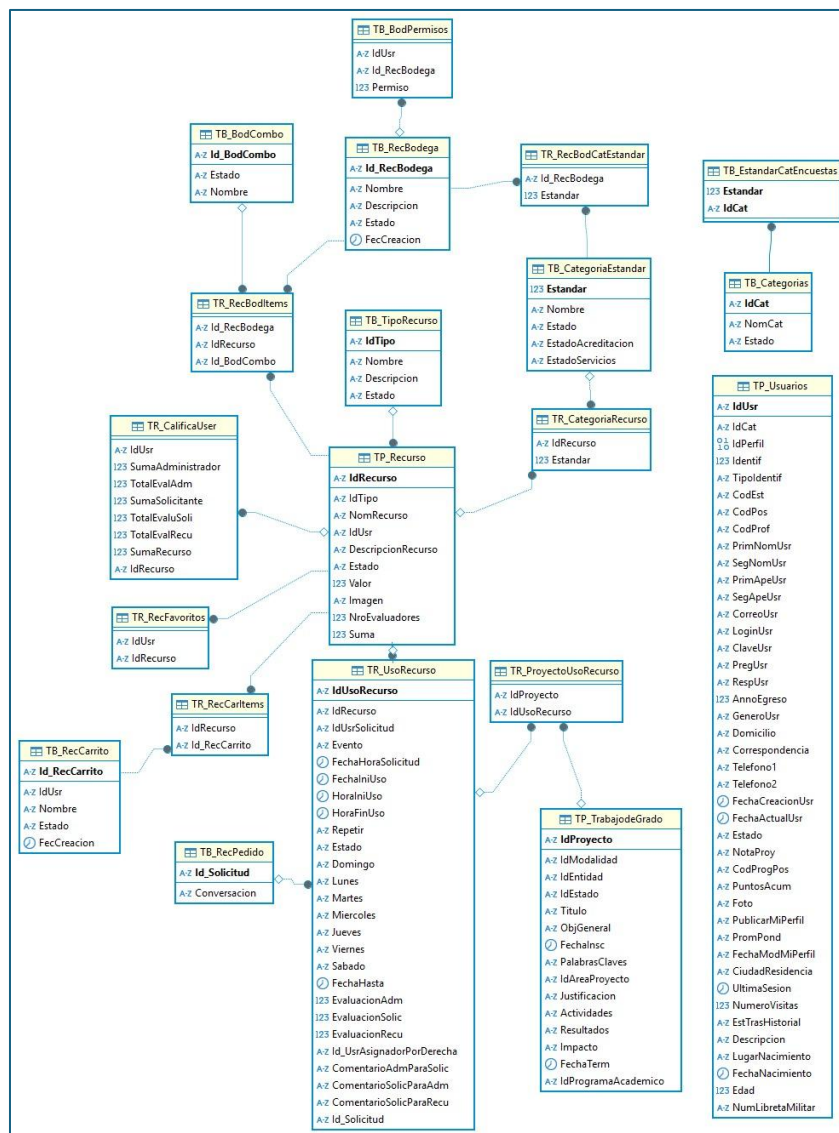


Como resultado de este ejercicio de ajuste y depuración, en la Figura 25 se puede apreciar un modelo intermedio que permitió avanzar en la construcción del prototipo. Este esquema ofrecía una base lo suficientemente estable para implementar funcionalidades clave y validar la

consistencia de las relaciones entre entidades, al tiempo que mantenía el alineamiento con las estructuras institucionales preexistentes. Sin embargo, se trataba aún de una versión transitoria, concebida principalmente como un punto de apoyo para la experimentación y la validación de casos de uso, más que como un diseño definitivo. En esta etapa, el modelo cumplió un papel fundamental en la exploración de alternativas y en la preparación del terreno para la consolidación del esquema final.

Figura 25

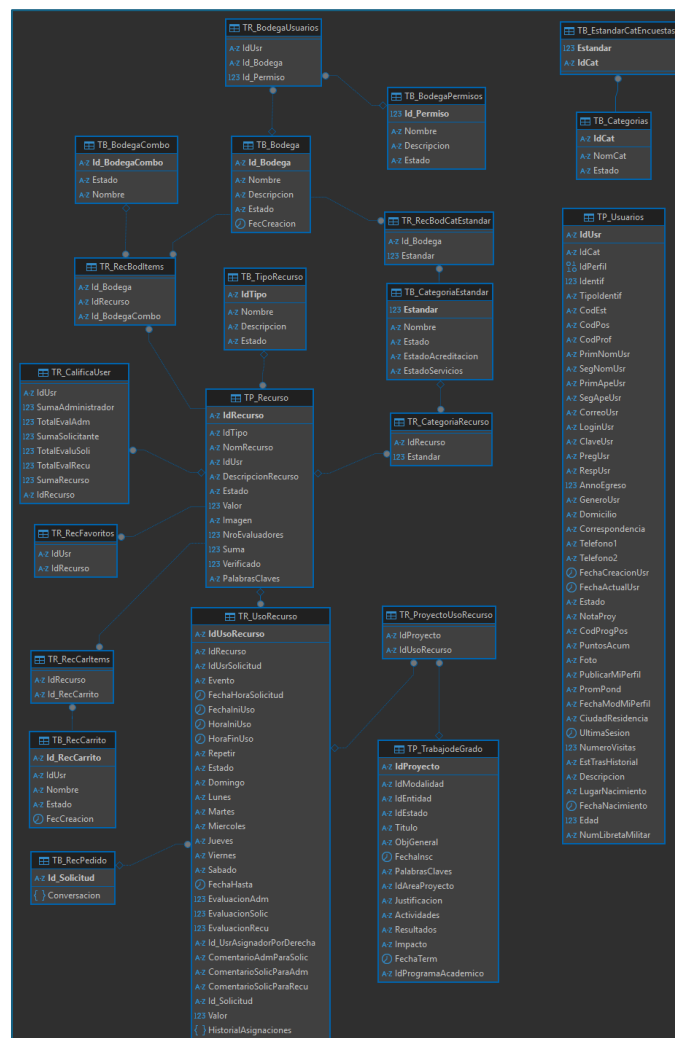
Cuarta versión del modelo entidad-relación pensado para el nuevo módulo.



El trabajo sobre el modelo se fue consolidando mediante una evolución constante, en la que cada ajuste respondía a la necesidad de mejorar la consistencia y la capacidad de representación. Más que transformaciones abruptas, el proceso estuvo marcado por la incorporación progresiva de variables y relaciones que enriquecieron lo ya existente, manteniendo un equilibrio entre simplicidad y detalle. De esta manera, la Figura 26 donde se aprecia la última iteración del modelo, no se entendió como una estructura estática, sino como un sistema en crecimiento, que fue afinándose a lo largo de cada iteración para dar soporte a los distintos escenarios planteados.

Figura 26

Quinta versión del modelo entidad-relación pensado para el nuevo módulo.

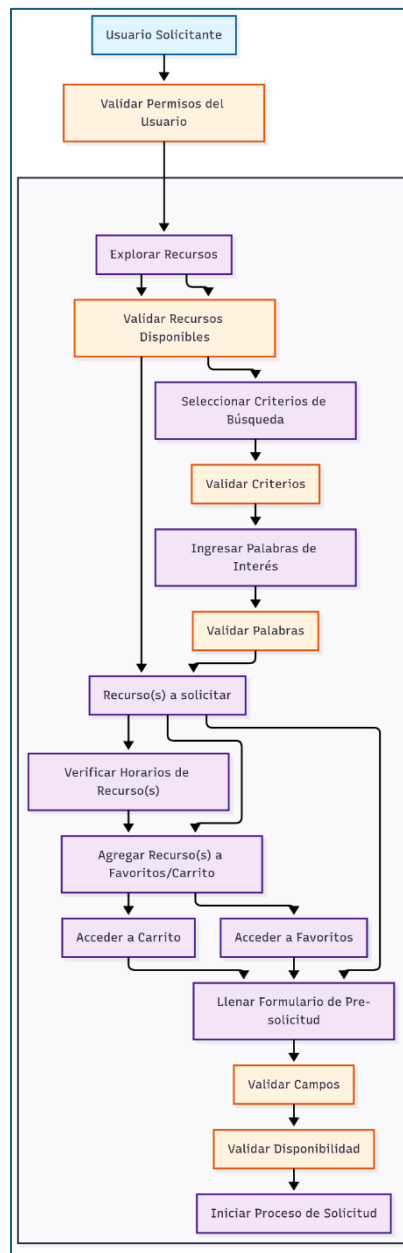


5.1.3.3 Diagramas de interacción. En relación con la experiencia de uso, se documentaron diagramas de interacción que describen la navegación entre vistas y las validaciones de negocio asociadas a cada operación. Se priorizó la reducción de pasos manuales y la claridad de los procesos, aspectos identificados en fases anteriores como determinantes para la aceptación por parte de los usuarios.

5.1.3.3.1 Solicitud. La Figura 27 representa el proceso completo de solicitud de recursos para usuarios que no cuentan con elementos previamente guardados en favoritos o carrito. El proceso inicia con la validación de permisos del usuario solicitante, seguido de la exploración de recursos disponibles donde se pueden aplicar criterios de búsqueda y palabras clave para filtrar las opciones. Una vez identificados los recursos de interés, el usuario puede o no verificar horarios de disponibilidad y opcionalmente agregarlos a favoritos o carrito para futuras consultas. El flujo culmina con el llenado del formulario de pre-solicitud, donde se validan tanto los campos ingresados como la disponibilidad final de los recursos, para finalmente iniciar el proceso formal de solicitud. El diagrama incluye múltiples puntos de validación (representados en naranja) que aseguran la integridad del proceso y casos de uso (en morado) que definen las acciones principales disponibles para el usuario.

Figura 27.

Diagrama de flujo de la solicitud cuando el usuario no ha interactuado previamente con los recursos.

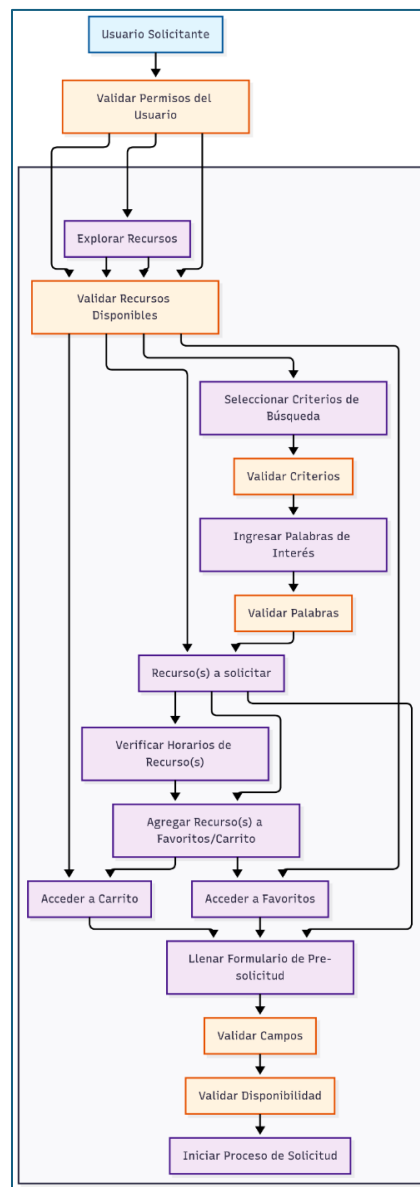


La Figura 28 muestra el flujo cuando el usuario ya cuenta con recursos en favoritos o carrito. El proceso se simplifica significativamente ya que el usuario puede acceder directamente a sus favoritos o carrito, saltándose los pasos de búsqueda y filtrado de criterios. Una vez que selecciona los recursos, procede directamente al llenado del formulario de pre-

solicitud, donde se validan los campos ingresados y la disponibilidad de los recursos. Posteriormente se verifica si existen conflictos de horario entre los recursos seleccionados antes de iniciar la solicitud formal. Este flujo alternativo reduce considerablemente los pasos necesarios para completar una solicitud, mejorando la experiencia del usuario que ya que este último ha identificado y guardado previamente los recursos de su interés.

Figura 28.

Diagrama de flujo de la solicitud cuando el usuario ya ha interactuado previamente con los recursos.



5.1.3.3.2 Gestión de solicitudes. La Figura 29 representa la gestión de solicitudes que incorpora un sistema de validaciones de seguridad y autorización para garantizar la integridad de las operaciones. El flujo inicia cuando el usuario solicitante accede a la sección de solicitudes, momento en el cual el sistema ejecuta una primera validación que verifica la relación legítima entre el usuario autenticado y las solicitudes que intenta consultar, previniendo accesos no autorizados a información de terceros. Una vez superada esta validación inicial, el sistema procede a cargar exclusivamente las solicitudes del usuario que mantengan estados válidos y activos en el sistema.

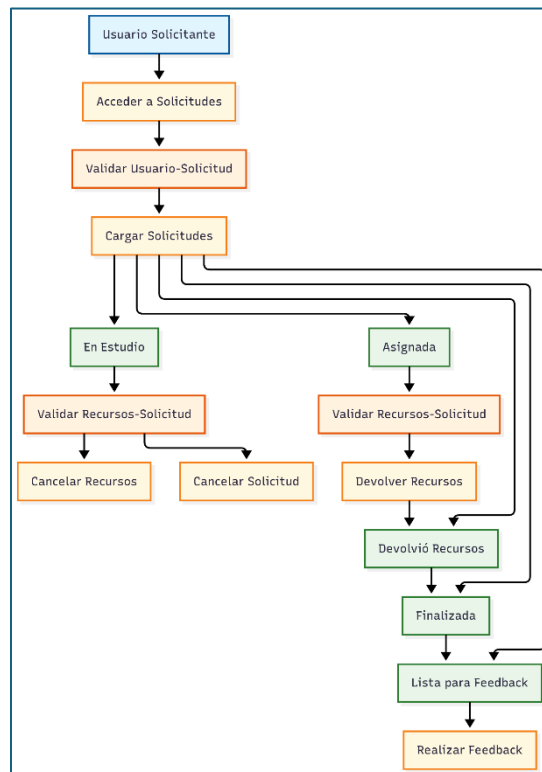
El diagrama contempla cinco estados distintos que reflejan el ciclo completo de vida de una solicitud. En el estado "En Estudio", correspondiente a solicitudes recién enviadas al propietario, el usuario mantiene control total sobre su petición, pudiendo ejecutar cancelaciones granulares de recursos específicos dentro de la solicitud o proceder con la cancelación completa de toda la solicitud, sin que esta última genere la posibilidad de realizar feedback posterior. El estado "Asignada" indica que el propietario ha aprobado la solicitud y los recursos han sido asignados al solicitante, habilitando automáticamente las funcionalidades de devolución una vez que el usuario haya utilizado los recursos solicitados.

Cuando el usuario procede a devolver los recursos, el sistema registra este evento en el estado "Devolvió Recursos", que representa la confirmación por parte del solicitante de que ha entregado de vuelta todos los elementos utilizados. Este estado progresa automáticamente hacia "Finalizada" cuando el propietario valida y confirma la recepción satisfactoria de todos los recursos devueltos, cerrando formalmente el ciclo transaccional del préstamo. Finalmente, el estado "Lista para Feedback" se activa tras la finalización, habilitando un sistema de evaluación bidireccional donde tanto el solicitante como el propietario pueden calificarse mutuamente y

agregar comentarios detallados sobre la experiencia del préstamo, fomentando la mejora continua del servicio y la confianza entre usuarios.

Figura 29.

Diagrama de flujo de la gestión de solicitudes.



5.1.3.3 Asignaciones. La Figura 30 representa el sistema completo de gestión de asignaciones, donde el usuario propietario (asignador) toma la iniciativa de asignar recursos específicos a usuarios solicitantes determinados, creando internamente una solicitud que debe ser gestionada por el usuario solicitante (asignado). El flujo contempla las perspectivas de ambos actores con sus respectivas validaciones de seguridad.

El usuario propietario (asignador) accede a sus asignaciones creadas tras validar su condición de propietario de los recursos, pudiendo visualizar seis estados: "Pendiente" mientras espera la respuesta del usuario solicitante; "Aceptada" cuando el solicitante ha confirmado

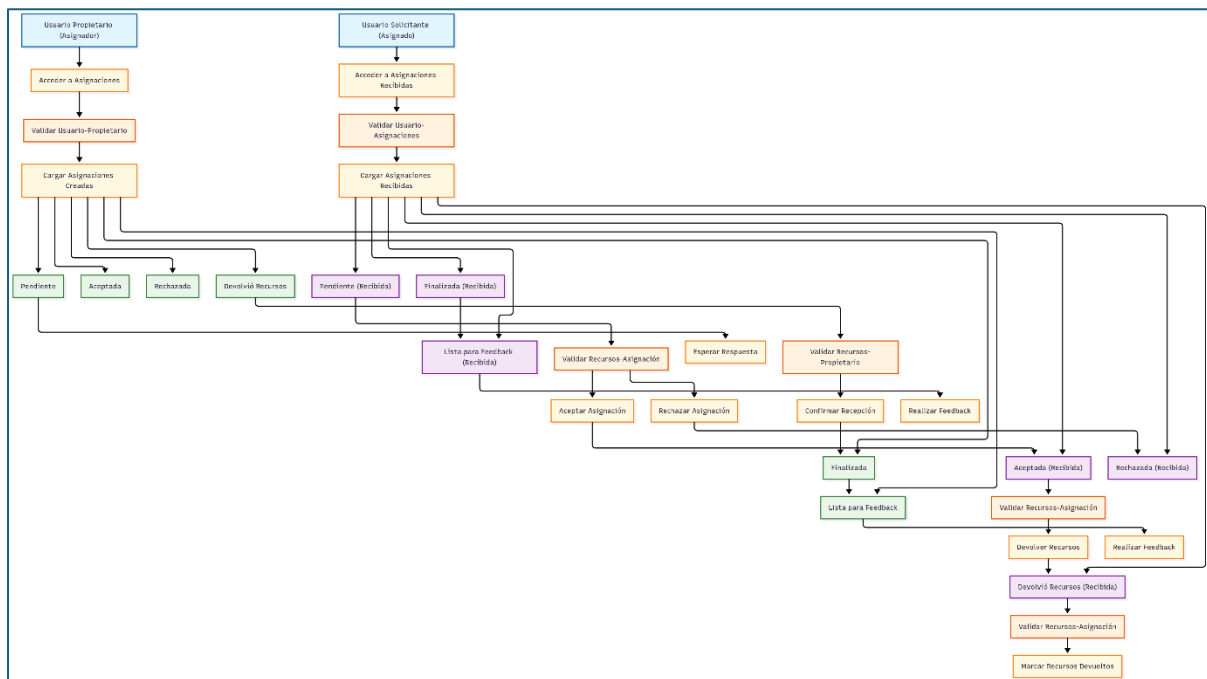
recibir la asignación; "Rechazada" si el solicitante declina la asignación; "Devolvió Recursos" cuando el solicitante ha marcado la devolución; "Finalizada" tras confirmar la recepción de los recursos devueltos; y "Lista para Feedback" para la evaluación mutua.

Desde la perspectiva del usuario solicitante (asignado), el sistema valida la relación usuario-asignaciones para confirmar que las asignaciones recibidas le corresponden legítimamente antes de cargarlas. En el estado "Pendiente (Recibida)" puede aceptar o rechazar la asignación tras validar que los recursos están correctamente asociados a la asignación; "Aceptada (Recibida)" le permite devolver los recursos cuando corresponda; "Rechazada (Recibida)" indica su negativa a la asignación; "Devolvió Recursos (Recibida)" confirma que ha entregado los recursos al propietario; "Finalizada (Recibida)" señala el cierre exitoso de la asignación; y "Lista para Feedback (Recibida)" habilita la calificación del propietario.

El sistema mantiene sincronización automática entre ambos flujos: la aceptación del solicitante se refleja como "Aceptada" para el propietario, el rechazo como "Rechazada", la devolución aparece para que el propietario confirme la recepción, y la finalización habilita el feedback bidireccional entre propietario y solicitante. Las validaciones multinivel garantizan que solo usuarios autorizados puedan gestionar asignaciones que les corresponden, asegurando la integridad y trazabilidad del proceso donde el propietario asigna proactivamente recursos a solicitantes específicos.

Figura 30

Diagrama de flujo de las asignaciones.



5.1.3.4 Interfaces. Como parte del proceso de diseño de la plataforma, se elaboraron diversos mockups que representaron propuestas visuales y funcionales para los módulos principales. Estos prototipos sirvieron para alinear criterios entre el equipo multidisciplinario, facilitar la retroalimentación de usuarios y identificar oportunidades de mejora antes de la implementación. Su revisión conjunta estableció las bases conceptuales que orientaron el desarrollo posterior del sistema.

5.1.3.4.1 Mockup del módulo de préstamo de recursos. El diseño inicial propuso una organización centrada en facilitar la identificación de funciones clave mediante una arquitectura de información jerárquica. Como se puede apreciar en la Figura 31, se planteó un panel lateral de navegación (20% de la pantalla) que incluiría el título "PRÉSTAMO DE RECURSOS" y la sección "Solicitante" con dos accesos principales: "Búsqueda y Solicitud de recursos" y "Gestión de préstamos y asignaciones". El área de trabajo ocuparía el 80% restante, buscando equilibrio visual entre navegación y espacio operativo.

Esta distribución respondía a principios de usabilidad web establecidos, donde el contenido principal debía dominar visualmente sin sacrificar la accesibilidad a las opciones de navegación. La propuesta visual contrastaba el panel de navegación en tonos azul marino con un área principal en colores neutros, siguiendo una estructura en rejilla con tipografía consistente y espaciado uniforme.

Figura 31.

Mockup del módulo de préstamo de recursos



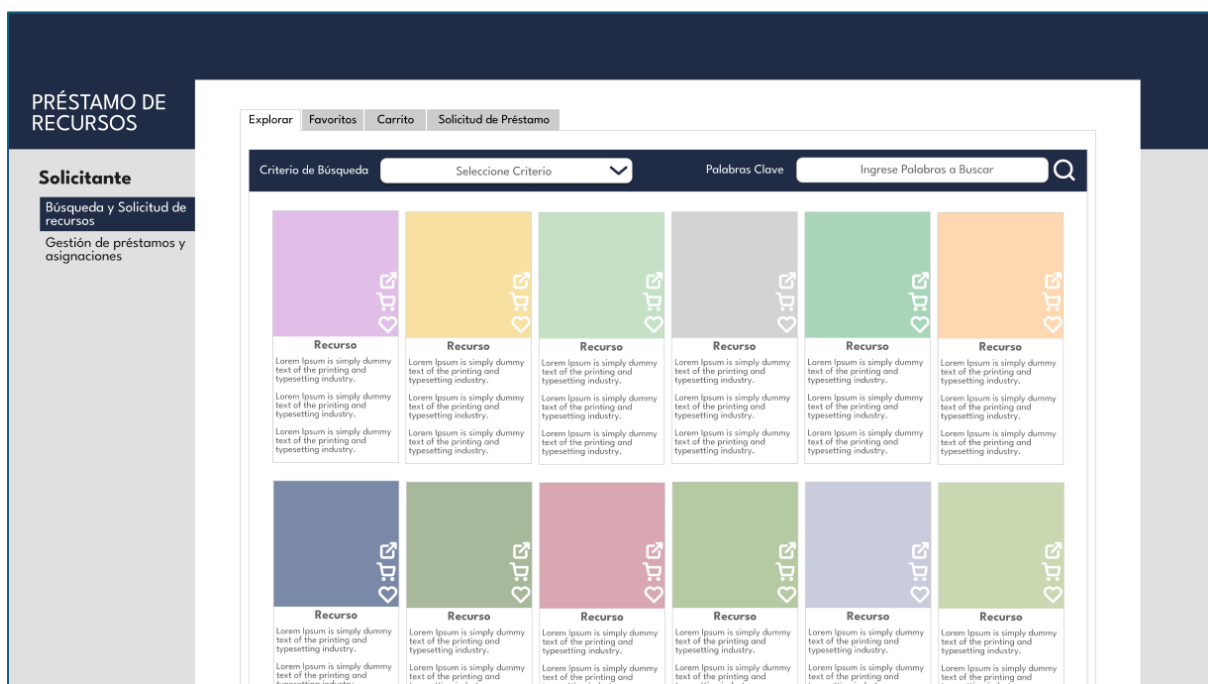
5.1.3.4.2 Mockup de búsqueda y exploración de recursos. La pestaña Explorar se diseñó con un sistema de búsqueda dual para atender diferentes patrones de uso. Como se puede apreciar en la Figura 32, se propuso combinar un selector desplegable con filtros predefinidos y un campo de texto libre para búsquedas por palabras clave, con un botón de acción claramente identificado.

Los resultados se organizarían en una rejilla responsiva que mostraría hasta doce recursos simultáneos, favoreciendo la comparación visual y aprovechando eficientemente el espacio disponible. Esta configuración buscaba mantener tiempos de navegación fluidos mediante paginación del contenido.

Cada recurso se presentaría en tarjetas estandarizadas que incluirían imagen representativa, título descriptivo, descripción breve y controles de acción (favoritos, carrito). En esta etapa del prototipo se utilizaron bloques de color pastel como sustitutos de imágenes, texto de ejemplo para las descripciones e iconografía básica para los controles. La propuesta contemplaba que en la implementación se integrarían imágenes reales, metadatos precisos del inventario y controles interactivos con retroalimentación visual.

Figura 32.

Mockup de búsqueda y exploración de recursos.



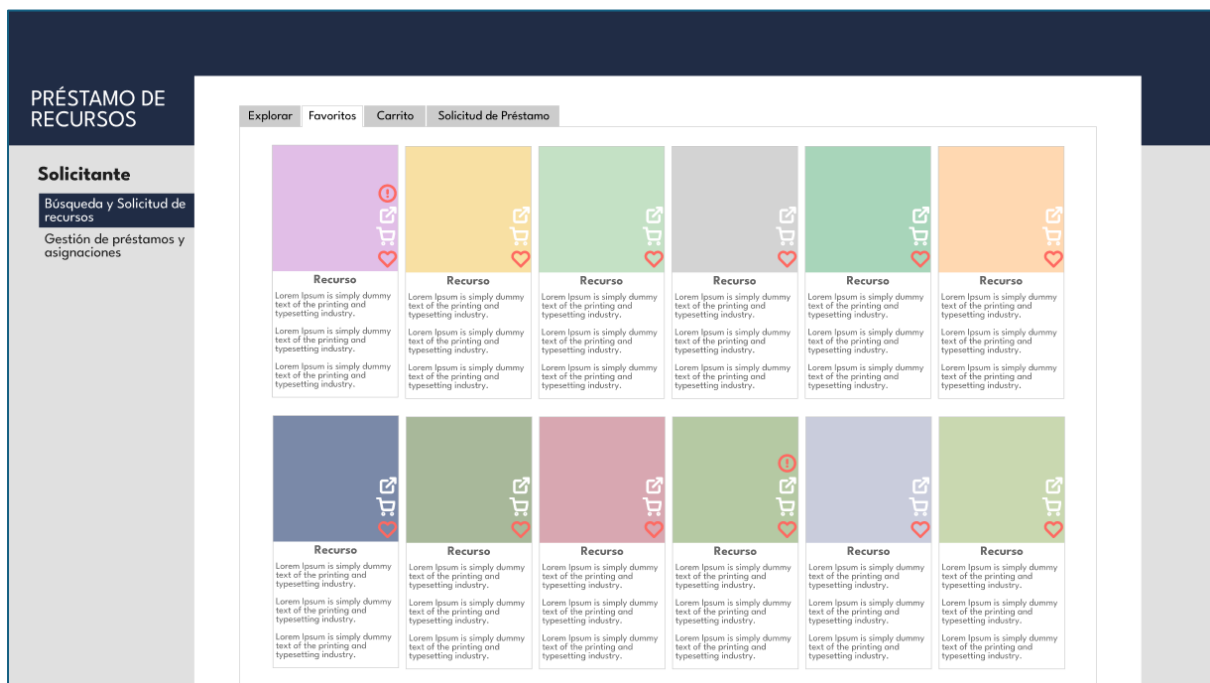
5.1.3.4.3 Mockups de favoritos y carrito. Como se puede observar en la Figura 33, la pantalla de Favoritos se conceptualizó como un espacio personal donde los usuarios

organizarían los recursos de interés antes de tomar decisiones definitivas. El diseño propuso una grilla 2x6 elementos con iconos de corazón activados para confirmar visualmente el estado de cada recurso.

Se planteó incluir controles contextuales para eliminar de favoritos, agregar al carrito, acceder a detalles o proceder directamente a solicitud. Durante las fases tempranas se consideró incorporar iconos de alerta para notificar cambios en disponibilidad, aunque esta funcionalidad fue descartada en revisiones posteriores.

Figura 33.

Mockup de la pestaña de favoritos del usuario.



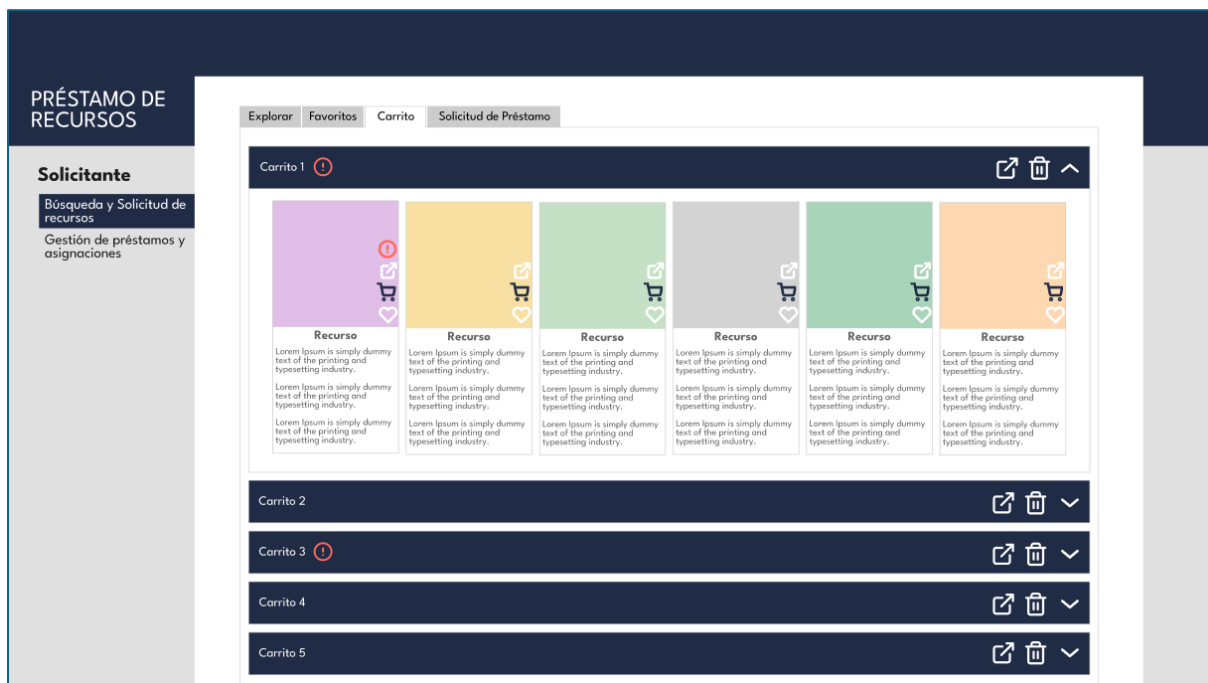
Como se puede observar en la Figura 34, la pantalla de Carrito se diseñó para concentrar los recursos seleccionados para préstamo en una interfaz funcionalmente similar a Favoritos. Tras evaluar la posibilidad de múltiples carritos por usuario, se optó conceptualmente por un carrito único para simplificar la experiencia de usuario.

El mockup mantuvo la grilla de tarjetas con iconos de carrito activados y controles para eliminar, revisar detalles o proceder a solicitud. Se propuso que el flujo se centrara en la preparación final antes de formalizar las solicitudes.

Ambos mockups utilizaron recursos de prueba con colores y textos simulados. La propuesta contemplaba implementar persistencia entre sesiones y validaciones en tiempo real para la versión funcional. Los principios de diseño aplicados priorizaban usabilidad, consistencia visual, escalabilidad y accesibilidad, posicionando estos módulos como filtros eficientes dentro del flujo conceptual de solicitud de préstamos.

Figura 34.

Mockup de la pestaña de carrito del usuario.



5.1.3.4.4 Mockup de solicitud de préstamo. Como se puede apreciar en la Figura 35, la pantalla de Solicitud de Préstamo fue conceptualizada como el espacio donde los usuarios formalizarían sus peticiones de recursos, cerrando el flujo iniciado en los módulos de exploración y carrito. En la versión original del mockup solo se gestionaba una solicitud a la

vez, pero en revisiones posteriores se incorporó la posibilidad de manejar múltiples pre-solicitudes de manera simultánea, permitiendo que los usuarios prepararan y organizaran varias solicitudes activas antes de ser enviadas.

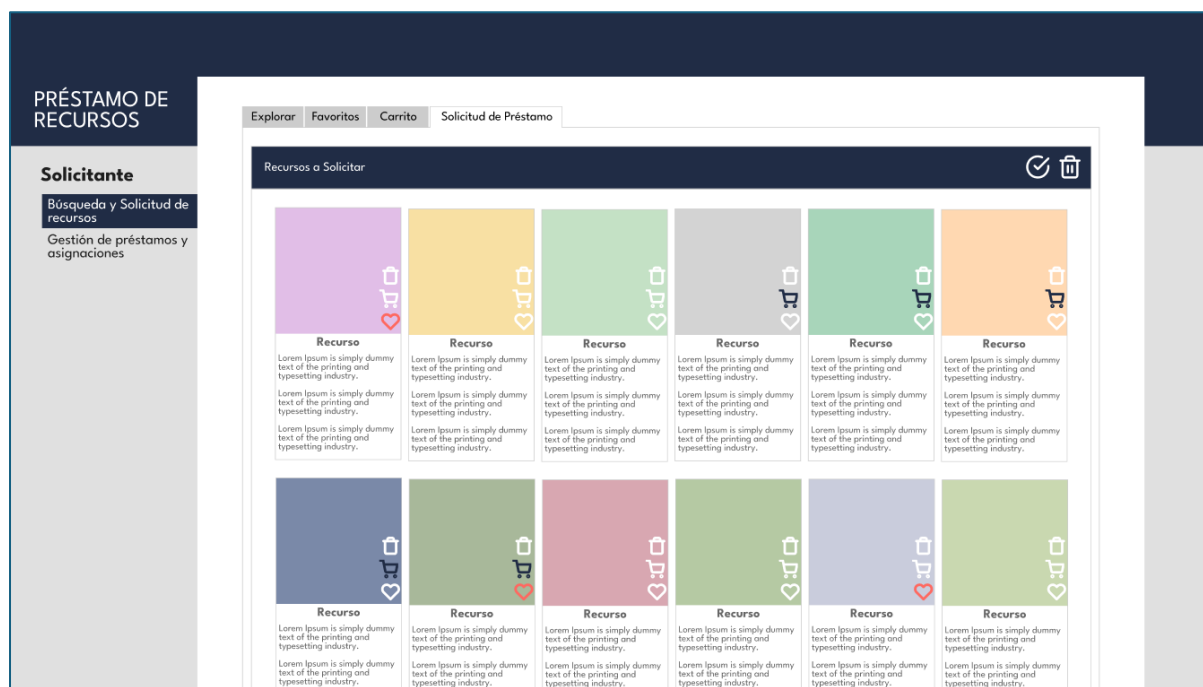
Visualmente, este módulo adoptó una estructura basada en tarjetas muy similar a la presentada en el mockup inicial del carrito, en la que cada pre-solicitud se representaba como una tarjeta independiente con opciones para confirmar o eliminar la solicitud según fuera necesario.

Durante el proceso de iteración se identificó que aún faltaba un formulario específico para el registro de los datos requeridos en cada solicitud. Este mecanismo se consideró para ser incluido en fases futuras del prototipo, garantizando que la recolección de información se alineara con los requerimientos institucionales y de usuario.

Este enfoque permitió dotar al módulo de mayor flexibilidad y adaptabilidad, manteniendo coherencia visual y operativa dentro de la plataforma y respondiendo a las necesidades complejas surgidas durante el proceso de diseño.

Figura 35.

Mockup de la pestaña de solicitud de préstamo.



5.1.3.4.5 Mockup de la pestaña de solicitudes. Como se puede observar en la Figura 36, la pestaña "Solicitudes" presenta el seguimiento detallado de las peticiones iniciadas por el usuario hacia los propietarios de recursos. El elemento más destacado del diseño fue la implementación de un indicador de progreso visual que muestra el flujo completo del proceso de solicitud. Este componente presenta cinco etapas claramente definidas: "Solicitud Realizada", "Enviada a Propietario", "Solicitud Aprobada", "Solicitud Finalizada" y "Lista para Feedback". Cada etapa se representa mediante iconografía específica y un sistema de conexión lineal que utiliza círculos rellenos para estados completados y círculos vacíos para estados pendientes.

La visualización de recursos asociados a cada solicitud adopta el mismo sistema de grilla utilizado en módulos anteriores, manteniendo las tarjetas de colores pastel con iconografía de carrito de compras y corazones. Sin embargo, en este contexto, las tarjetas

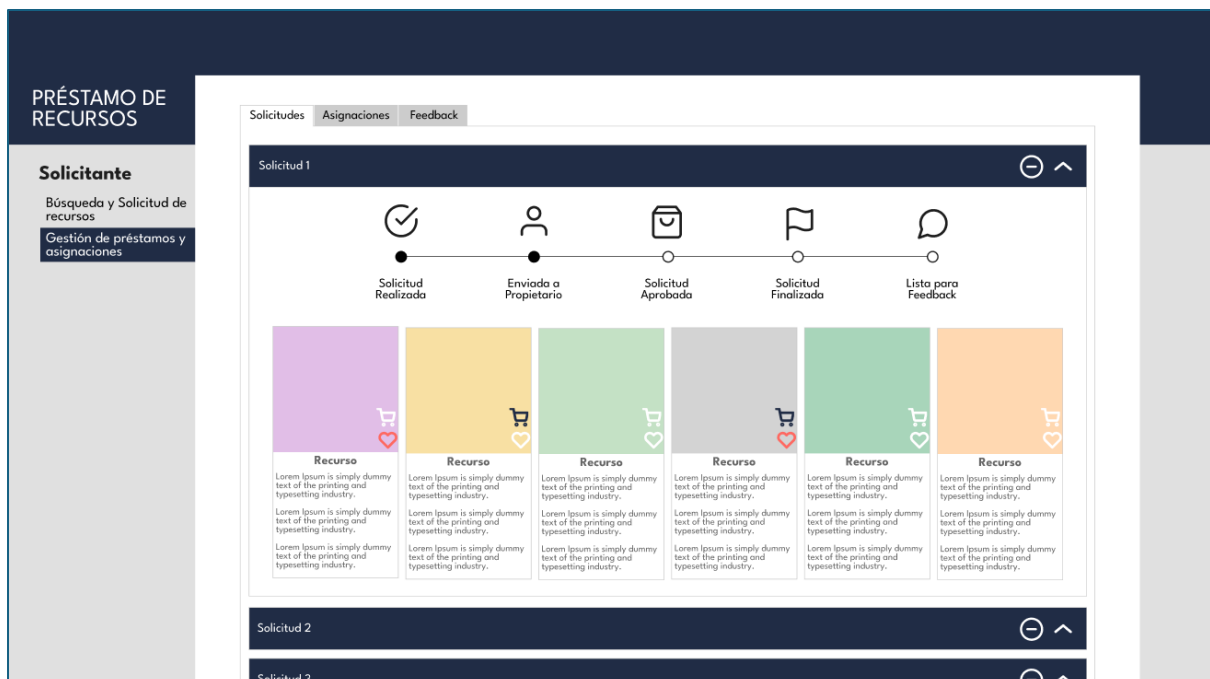
funcionan como elementos informativos que muestran el estado actual de los recursos dentro del flujo de préstamo.

El mockup incorpora un sistema de solicitudes múltiples colapsables, donde cada solicitud puede expandirse o contraerse mediante controles ubicados en la esquina superior derecha. Este enfoque permite a los usuarios gestionar varias solicitudes simultáneamente sin saturar visualmente la interfaz, respondiendo a la necesidad identificada durante las iteraciones de diseño de manejar múltiples procesos de préstamo de forma eficiente.

La implementación de este sistema de seguimiento visual representa una evolución significativa respecto a los mockups de exploración, transformando la plataforma de un sistema de búsqueda y selección hacia una herramienta completa de gestión del ciclo de vida de los préstamos de recursos.

Figura 36.

Mockup de la pestaña de solicitudes.



5.1.3.4.6 Mockup de la pestaña de asignaciones. Como se puede apreciar en la Figura 37, la pestaña "Asignaciones" representa un flujo inverso al proceso tradicional de solicitud, donde los propietarios de recursos toman la iniciativa de asignar directamente recursos a usuarios específicos sin que estos los hayan solicitado previamente.

En esta interfaz, el solicitante visualiza que "El usuario Pepito Pérez le ha asignado los siguientes recursos", donde Pepito Pérez actúa como propietario que decidió proactivamente ofrecer estos recursos al usuario actual. Este flujo permite que los propietarios gestionen sus inventarios de manera activa, identificando usuarios que podrían beneficiarse de recursos específicos.

Los controles de "Aprobar" y "Rechazar" otorgan al receptor la decisión final sobre si acepta o declina los recursos que le fueron asignados directamente. Esta mecánica invierte la dinámica tradicional, convirtiendo al solicitante en el punto final de aprobación de una transacción iniciada por el propietario.

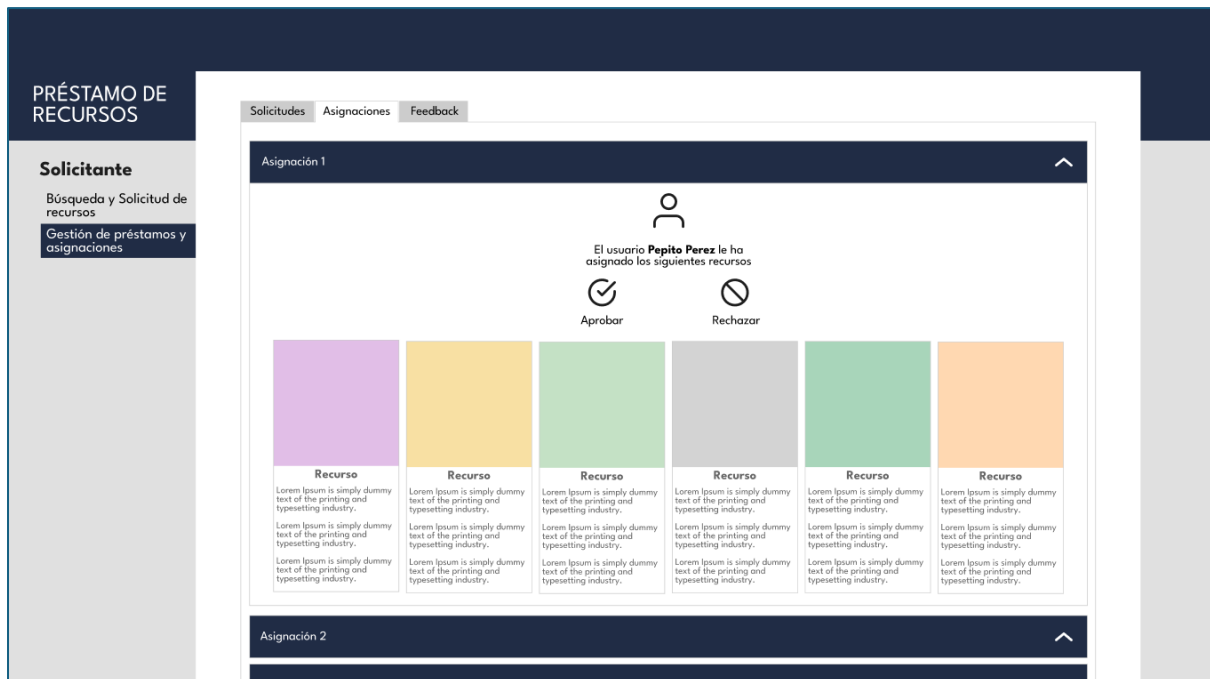
La grilla de recursos muestra los elementos que el propietario seleccionó específicamente para este usuario, basándose en criterios como historial de préstamos, necesidades identificadas, disponibilidad de inventario o políticas de distribución proactiva. Esta visualización permite al receptor evaluar la pertinencia de los recursos ofrecidos.

El sistema de asignaciones múltiples permite que un usuario reciba ofertas independientes de diferentes propietarios de manera secuencial o simultánea, donde cada asignación representa una transacción completa e independiente con un solo propietario. Cada asignación requiere una decisión individual del receptor, manteniendo la autonomía en la aceptación de recursos.

Este modelo complementa el flujo tradicional de solicitud, creando un sistema híbrido donde tanto la demanda (solicitudes) como la oferta (asignaciones proactivas) pueden iniciar el proceso de préstamo de recursos.

Figura 37.

Mockup de la pestaña de asignaciones.



5.1.3.4.7 Mockup de la pestaña de feedback. Como se puede observar en la Figura 38, la pestaña "Feedback" representa la fase final del ciclo de préstamo de recursos, donde los usuarios pueden evaluar y comentar sobre su experiencia con los recursos utilizados y el proceso general de préstamo. Esta interfaz cierra el circuito de retroalimentación del sistema, proporcionando información valiosa tanto para propietarios como para la mejora continua de la plataforma.

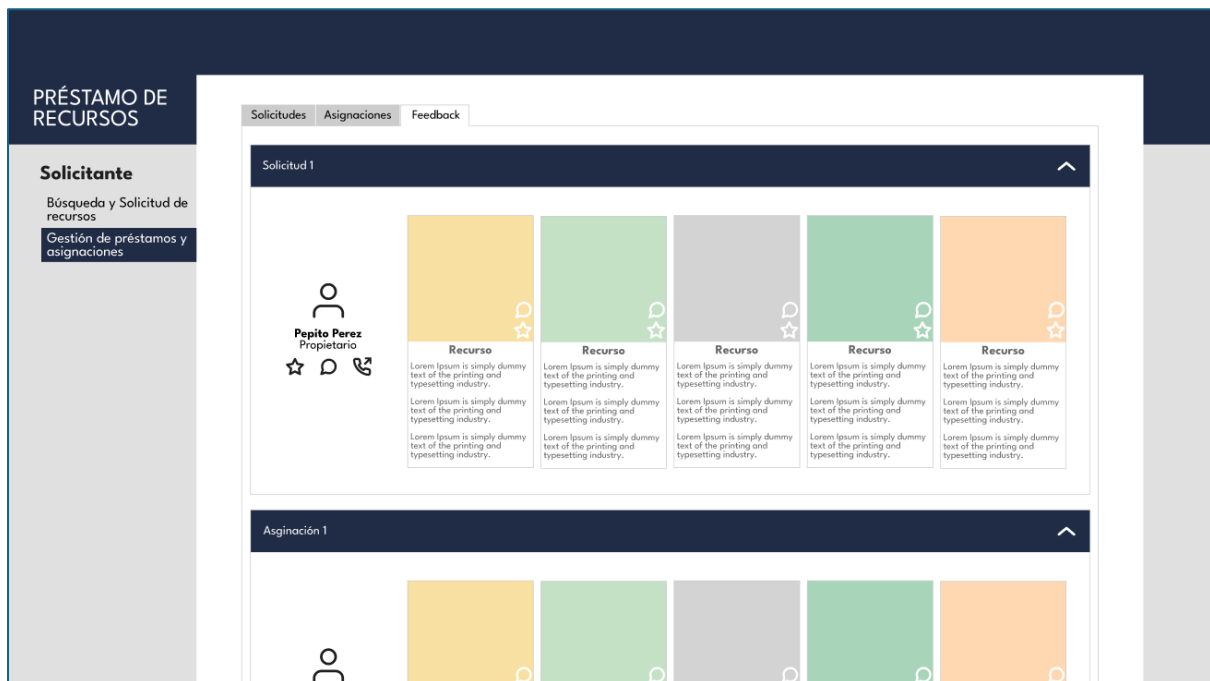
El diseño mantiene la estructura colapsible establecida en las pestañas anteriores, permitiendo que los usuarios gestionen múltiples procesos de feedback de manera organizada.

Cada sección corresponde a una transacción completada, ya sea originada por solicitud del usuario o por asignación proactiva del propietario.

La identificación del propietario "Pepito Pérez" en la sección superior establece el contexto de la evaluación, permitiendo que el feedback se asocie correctamente con el responsable de los recursos. Esta trazabilidad es fundamental para que los propietarios puedan recibir retroalimentación específica sobre su gestión y los recursos bajo su responsabilidad.

La grilla de recursos mantiene la iconografía de estrella, que en este contexto funciona como sistema de calificación, permitiendo a los usuarios valorar individualmente cada recurso utilizado. Esta granularidad en la evaluación proporciona información detallada que puede utilizarse para mejorar la calidad del inventario y identificar recursos problemáticos o especialmente valorados.

El sistema de feedback múltiple permite evaluar diferentes transacciones de manera independiente, reconociendo que cada experiencia de préstamo puede tener características únicas que requieren evaluación específica. Esta separación facilita análisis posteriores sobre patrones de satisfacción por propietario, tipo de recurso o período de préstamo.

Figura 38.*Mockup de la pestaña de feedback*

5.1.3.5 Construcción iterativa. A partir del diseño aprobado, el desarrollo se llevó a cabo de manera progresiva, priorizando primero los elementos fundamentales y avanzando después hacia aquellos de carácter complementario. En cada etapa se verificó el funcionamiento de lo implementado y se aplicaron ajustes cuando fue necesario, lo que permitió mantener un control sobre la calidad del producto. Este enfoque favoreció la detección oportuna de incidentes y redujo el riesgo de que incorporaciones posteriores comprometieran lo ya establecido, contribuyendo a conservar la estabilidad general del sistema.

5.1.3.5.1 Tarjeta del recurso. El componente fue diseñado como una pieza adaptable, capaz de mantener coherencia visual y funcional en todas las pestañas del sistema, a la vez que incorporaba variaciones según el contexto de uso. La versión inicial reunía los elementos básicos (imagen, nombre y breve descripción) dentro de un diseño limpio con bordes redondeados, sombras discretas y transiciones suaves que aportaban fluidez en la interacción.

Como se puede observar en la Figura 39, uno de los rasgos más distintivos del componente fue el uso de iconografía interactiva. Corazones que cambiaban de estado para marcar favoritos, indicadores de carrito, sistemas de calificación y otros iconos que variaban de color según su condición aportando dinamismo sin sobrecargar la tarjeta. Estos recursos no solo hicieron más intuitiva la interacción, sino que además se reutilizaron en distintas pestañas bajo la misma lógica visual, reforzando la consistencia del sistema.

Figura 39.

Iconos interactivos de cada recurso.



En la pestaña Explorar y, como se puede apreciar en la Figura 40, la tarjeta incorporó un botón para solicitar préstamos de forma directa. Aunque se trataba de una adición simple, su integración junto a los iconos mantuvo la armonía visual y transformó la tarjeta en un punto de acción, no únicamente de consulta.

Figura 40.

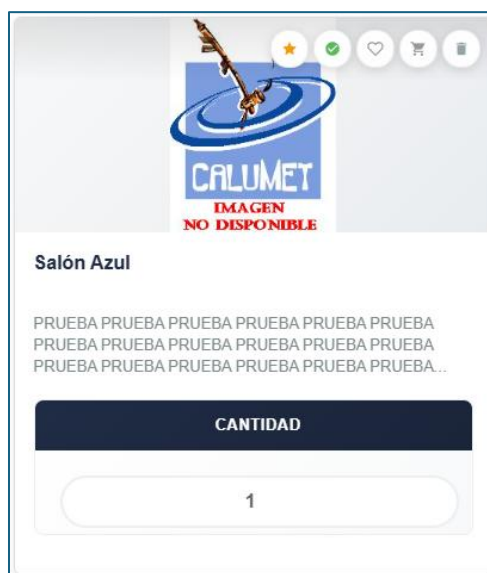
Tarjeta del recurso para la pestaña Explorar.



En Pre-solicitud y, como se puede apreciar en la Figura 41, la tarjeta pasó a integrarse dentro de un formulario en el que se incluían controles para indicar cuánto se deseaba solicitar de cada recurso. Estos campos se complementaban con retroalimentación visual mediante iconos, que ayudaban a confirmar las acciones realizadas y a mantener claridad sobre el estado de la tarjeta. También se añadió un botón para eliminar recursos de la pre-solicitud, lo que daba al usuario la posibilidad de ajustar sus decisiones con facilidad. La noción de *cantidad solicitada* introducida en esta pestaña se mantuvo como base en las siguientes, evolucionando según el contexto de cada etapa.

Figura 41.

Tarjeta del recurso para la pestaña Pre-solicitud.



En la pestaña de Solicitudes se profundizó en la complejidad informativa. Como se puede apreciar en la Figura 42, las tarjetas reutilizaron la estructura de cantidades, ahora vinculadas con la cantidad solicitada de recursos, y desplegaron estados diferenciados mediante iconos que cambiaban de color: en estudio, asignado, rechazado o finalizado. Este sistema no solo facilitó la lectura rápida, sino que también se consolidó como un patrón reutilizable que reapareció en la etapa posterior.

Figura 42.

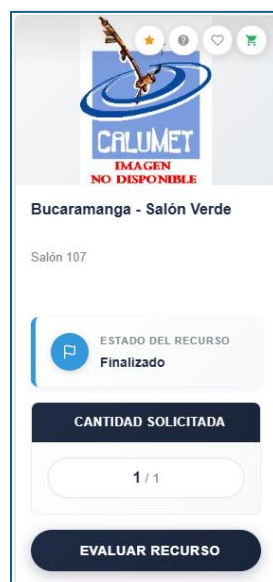
Tarjeta del recurso para la pestaña Solicitudes.



Finalmente, en la pestaña Feedback y, como se puede apreciar en la Figura 43, la tarjeta retomó la estructura de Solicitudes, reutilizando tanto los estados como el componente de cantidad solicitada. Sobre esta base se añadió un botón destacado para acceder a la evaluación, reforzando la importancia de la retroalimentación dentro del ciclo del sistema.

Figura 43.

Tarjeta del recurso para la pestaña Feedback.

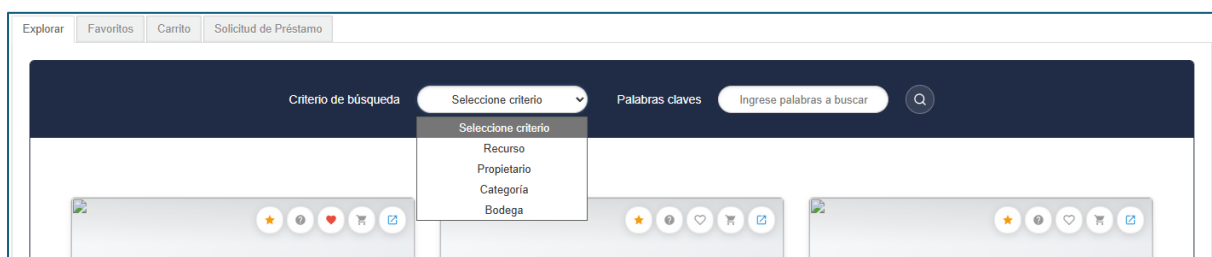


La evolución de las tarjetas no consistió en crear funciones aisladas para cada pestaña, sino en reutilizar y ampliar componentes ya validados, como los iconos, los estados o los controles de cantidad. Esto permitió que el diseño creciera de manera orgánica, reduciendo la complejidad de desarrollo y ofreciendo a los usuarios una experiencia coherente y predecible. En conjunto, la tarjeta se consolidó como un componente flexible y estable, capaz de guiar la interacción del usuario desde la exploración inicial hasta la valoración final, sin perder equilibrio entre claridad, funcionalidad y consistencia visual.

5.1.3.5.2 Pestaña Explorar. La pestaña Explorar se consolidó como el núcleo de navegación y descubrimiento del sistema, pensada para ofrecer una experiencia clara y flexible que permitiera recorrer el inventario de distintas maneras sin perder la coherencia visual definida desde el inicio. Como se puede observar en la Figura 44, el buscador es el punto de partida de esta interacción y está compuesto por un campo de texto para palabras clave y un botón con icono, integrados en una barra de herramientas centrada y adaptable que permite consultar por recurso, propietario, categoría o bodega.

Figura 44.

Diseño del buscador de la pestaña Explorar.

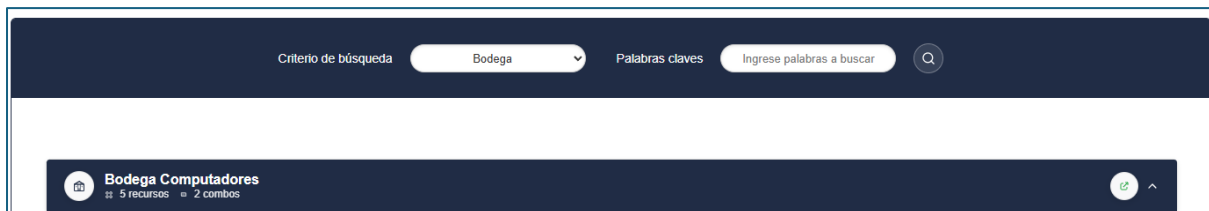


La presentación de resultados cambia según el filtro aplicado. En ese orden de ideas, cuando la búsqueda se realiza por bodegas, la interfaz organiza la información en torno a ellas como contenedores principales. Como se puede observar en la Figura 45, cada bodega aparece en una tarjeta con cabecera en color corporativo, iconos alusivos a almacenes, un

indicador del número de recursos y combos disponibles, además de una breve descripción que contextualiza su función. En el encabezado de estas tarjetas se añadió también un botón para solicitar directamente los recursos asociados, facilitando la interacción.

Figura 45.

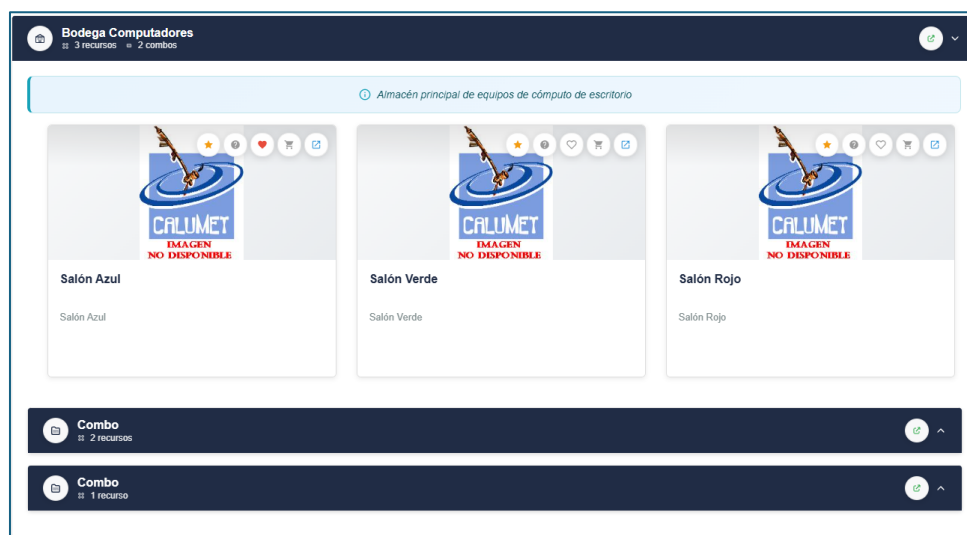
Vista del filtro cuando en el criterio de búsqueda se especifica bodega.



Dentro de las bodegas, los combos se representan con iconos de carpetas y mantienen la misma lógica visual. Como se puede observar en la figura 46, al desplegarlos, se revelan los recursos que los conforman, siempre bajo una estructura jerárquica clara. Al igual que en las bodegas, los combos cuentan en su cabecera con un botón que permite solicitar todos los recursos que incluyen, simplificando el proceso y reforzando la continuidad en la experiencia.

Figura 46.

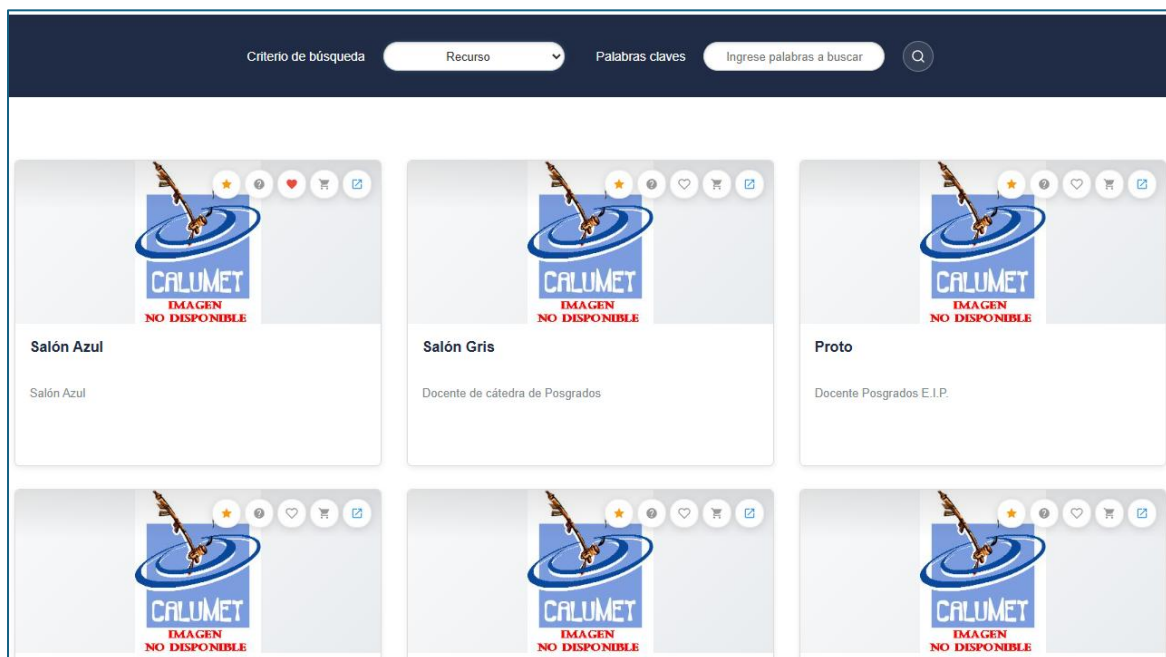
Diseño de la bodega y sus respectivos combos.



Como se puede apreciar en la Figura 47, cuando la búsqueda se realiza por recurso, propietario o categoría, el sistema omite la jerarquía de bodegas y muestra directamente los elementos que cumplen con la condición. Para ello se utiliza un componente reutilizable que garantiza uniformidad en la presentación tanto dentro como fuera de las bodegas, asegurando consistencia en cualquier contexto.

Figura 47.

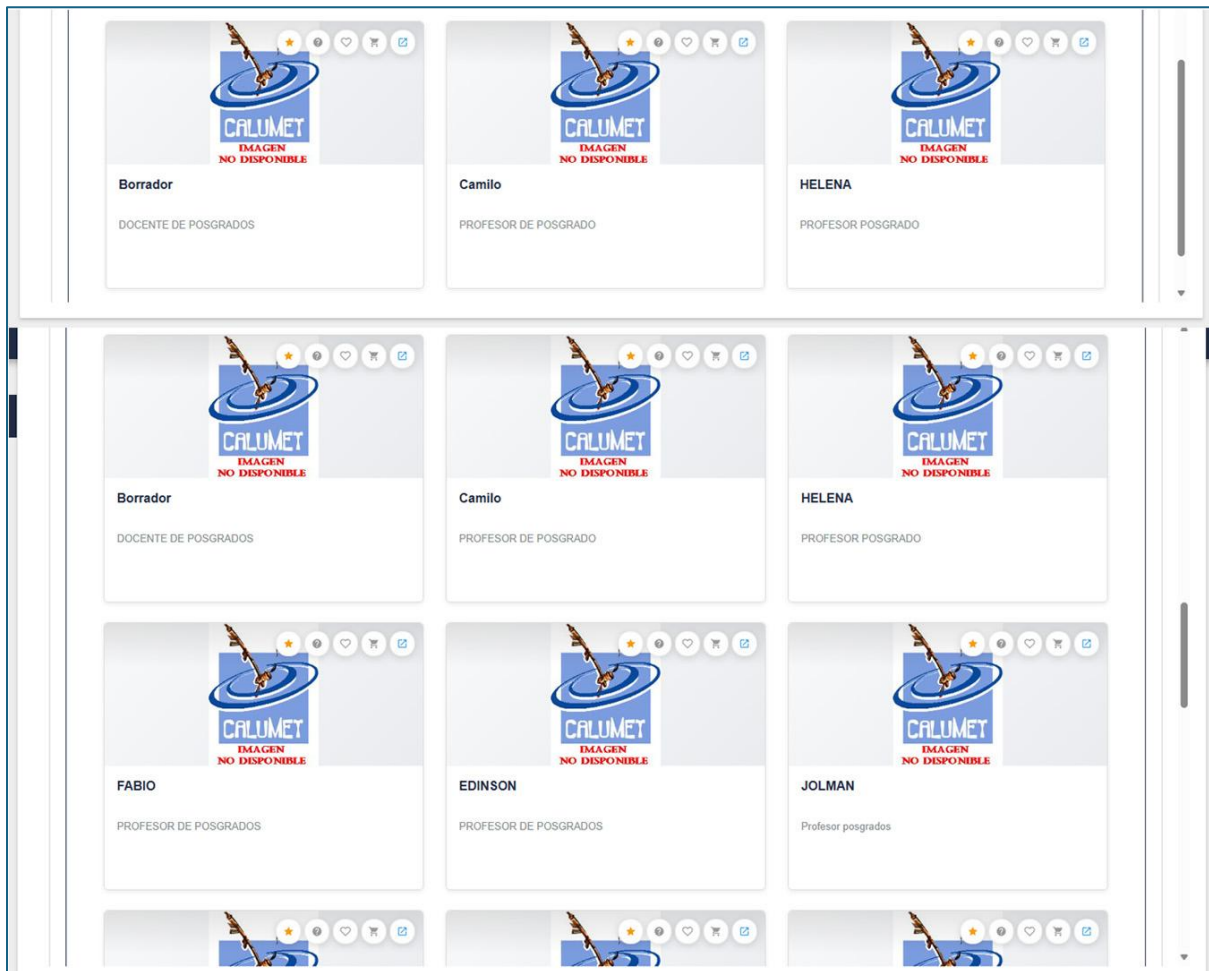
Vista del filtro cuando en el criterio de búsqueda se especifica recurso.



Como se aprecia en la Figura 48, la pestaña incorpora un sistema de carga progresiva que añade resultados conforme el usuario se aproxima al final de la página, con el fin de mantener la fluidez en inventarios extensos.

Figura 48.

Sistema de carga progresiva de recursos.



Esto evita interrupciones y permite una exploración continua y ordenada, ya sea en vistas basadas en bodegas y combos o en la presentación directa de recursos.

5.1.3.5.3 Pestaña Pre-solicitud. La pestaña de Pre-solicitud se configuró como el eje operativo del sistema, encargada de organizar de manera estructurada el proceso de solicitud y garantizar que cada recurso pueda asignarse sin conflictos. Como se puede observar en la Figura 49 y, desde su diseño inicial, se planteó un espacio donde los usuarios pudieran revisar y gestionar sus solicitudes pendientes, presentadas en acordeones expandibles que facilitan la navegación incluso cuando existen múltiples registros activos.

Figura 49.

Diseño de la pestaña de Pre-solicitudes.



Para preservar la coherencia de la información, se incorporaron validaciones que garantizan el registro completo de los datos obligatorios y la correcta relación entre parámetros como fechas, horarios y configuraciones de repetición como se puede observar en la Figura 50. A ello se añade una verificación de disponibilidad que controla la capacidad de los recursos solicitados: impide sobrepasar los límites en aquellos de cantidad múltiple y previene conflictos de horario en los de uso único. De forma complementaria, un proceso de detección de solapamientos revisa las solicitudes existentes para anticipar posibles choques, aplicando reglas específicas según el tipo de recurso y su configuración. Cuando el usuario requiere solicitudes recurrentes, la pestaña ofrece controles que verifican la consistencia entre los días seleccionados, los rangos de fecha y el patrón de repetición definido.

Figura 50.

Diseño del formulario para la pre-solicitud de recursos.

The screenshot shows a web form titled "Solicitud ID: RP10" with a sub-header "1 recurso". The form contains several input fields:

- Nombre del evento:** A text input field with the placeholder "Ingrese nombre del evento".
- Fecha de inicio:** A date picker field showing "mm/dd/aaaa".
- Hora de inicio:** A time picker field showing "--:-- --".
- Fecha de finalización:** A date picker field showing "mm/dd/aaaa".
- Hora de finalización:** A time picker field showing "--:-- --".

Below the input fields, there is a section titled "¿Cómo funciona la repetición?" with a sub-header "El botón «Repetir» permite reservar el recurso solo en los días que elijas dentro del rango de fechas. Si no lo activas, se asumirá que necesitas el recurso todos los días y durante todo el día." This section includes a "Repetir" button and a checkbox labeled "Activar evento recurrente".

Finalmente, cada interacción se acompaña de mensajes de retroalimentación que confirman las asignaciones exitosas o señalan con claridad los ajustes necesarios cuando se detecta un error como se puede observar en la Figura 51. Esto no solo brinda seguridad al usuario, sino que también contribuye a la eficiencia y confiabilidad del sistema en su conjunto.

Figura 51.

Mensajes de retroalimentación de la pestaña Pre-solicitudes.

The screenshot displays four distinct feedback messages arranged in a 2x2 grid:

- Error de validación:** A red-bordered box with a red 'X' icon. The message reads: "La cantidad para el recurso RC114 debe estar entre 1 y 1. Valor ingresado: 0." Below the message is a red "Entendido" button.
- Campos incompletos:** An orange-bordered box with an orange triangle icon. The message reads: "Por favor completa todos los campos requeridos." Below the message is an orange "Entendido" button.
- ¿Estás seguro?:** An orange-bordered box with an orange triangle icon. The message reads: "Se cancelarán 1 recurso(s) de la solicitud." Below the message is an orange "Sí, Cancelar" button.
- ¡Validación exitosa!** A green-bordered box with a green checkmark icon. The message reads: "Solicitud realizada correctamente." Below the message is a green "Aceptar" button.

5.1.3.5.4 Pestaña Solicitudes. La pestaña de Solicitudes constituye el centro de seguimiento y gestión del sistema, donde los usuarios pueden consultar el estado de sus peticiones y actuar sobre ellas de forma directa. Su construcción avanzó de manera progresiva: primero se establecieron las funciones básicas de visualización y, con el tiempo, se incorporaron mecanismos más completos que dieron lugar a una gestión integral. Como se puede apreciar en la Figura 51, desde el inicio se adoptó un formato tipo acordeón, en el que cada tarjeta puede expandirse o contraerse según la necesidad. Esto ofrece una visión general en una sola pantalla, pero también permite profundizar en los detalles de cada caso. Dentro de cada tarjeta se muestran de inmediato datos clave como el identificador único, el estado actual y la cantidad de recursos involucrados, con un diseño adaptable a distintos tamaños de pantalla para mantener legibilidad y facilidad de uso.

Figura 52.

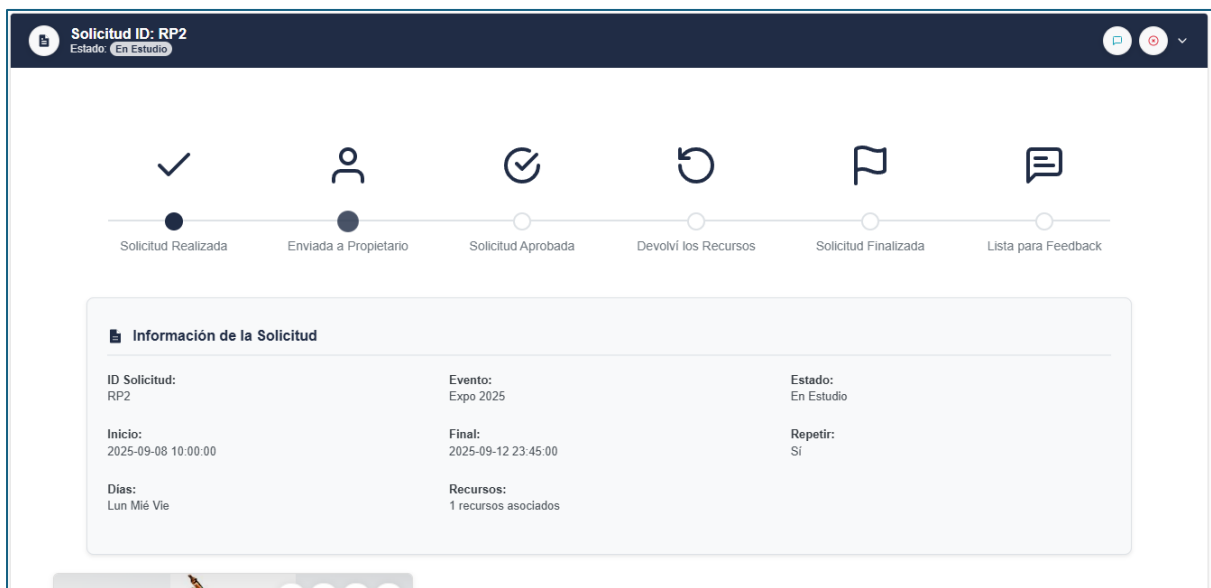
Vista de la pestaña Solicitudes cuando no se selecciona ningún filtro.



Como se puede apreciar en la Figura 53, uno de los rasgos más distintivos de esta pestaña es el sistema visual de seguimiento, que representa el recorrido de cada solicitud a través de una línea de progreso con iconos. Este esquema contempla etapas como la creación de la solicitud, el envío al propietario, la aprobación, la asignación, la devolución, el cierre y la retroalimentación final. Los pasos completados aparecen confirmados con marcas de verificación, el actual se resalta y los futuros se muestran atenuados.

Figura 53.

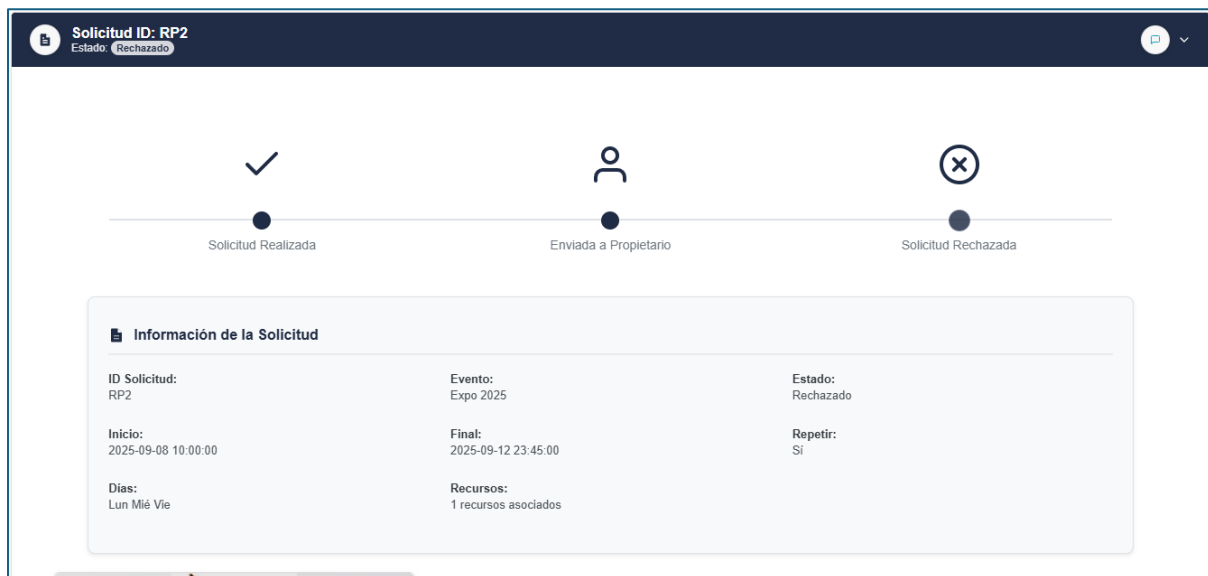
Vista de la información de una solicitud.



En casos excepcionales, como solicitudes canceladas o rechazadas, la ruta se adapta para dejar claro que el proceso tomó un curso diferente como se puede apreciar en la Figura 54.

Figura 54.

Vista de una solicitud rechazada.



Como se puede apreciar en la Figura 55, el diseño también reconoce que no todas las solicitudes avanzan de manera lineal. Cuando una petición involucra varios recursos en estados distintos, el sistema resume la situación con etiquetas como Aprobada parcialmente, sin ocultar el detalle de cada componente. Esta lógica compara los estados de todos los recursos asociados y comunica el resultado de forma clara, evitando confusiones.

Figura 55.

Vista de solicitud con recursos con estado aprobado y en estudio.



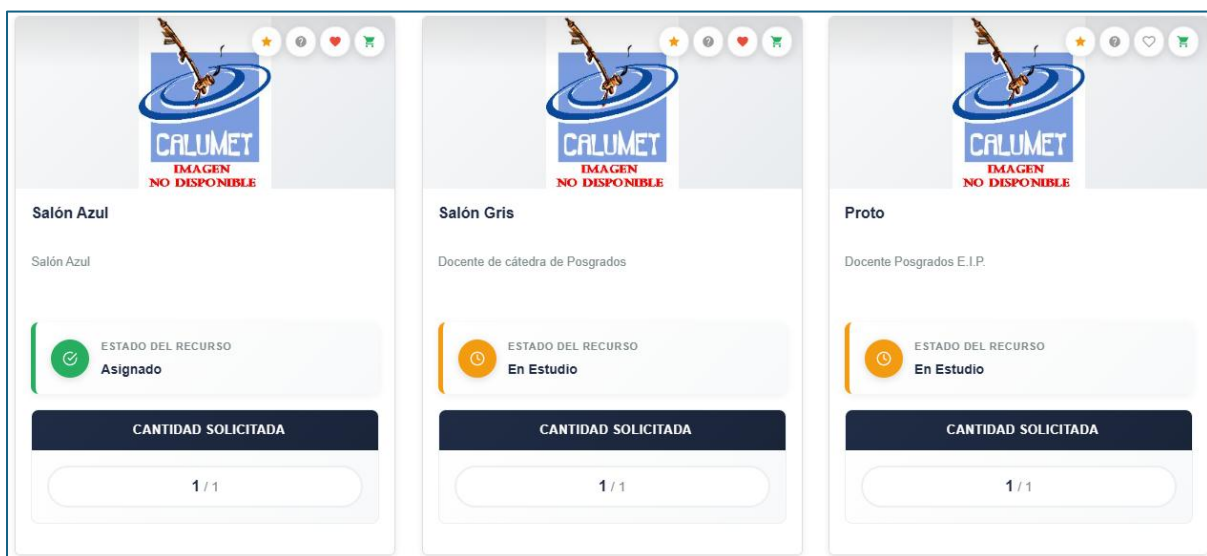
Cada tarjeta ofrece además botones de acción que aparecen únicamente cuando resultan pertinentes. Así, el usuario encuentra siempre opciones útiles y no se enfrenta a botones inactivos o irrelevantes. Por ejemplo, la opción de confirmar devolución solo aparece cuando los recursos han sido asignados, y la de cancelar únicamente mientras la solicitud sigue en

curso. Estas acciones se acompañan de respuestas visuales inmediatas que refuerzan su carácter interactivo.

Como se puede apreciar en la Figura 56, la información de los recursos asociados se muestra utilizando los mismos componentes de tarjeta empleados en otras partes del sistema, lo que garantiza uniformidad en la experiencia. Cada recurso aparece con su imagen, descripción, estado individual y notas relevantes, de modo que las diferencias entre recursos aprobados, rechazados o pendientes se perciben de inmediato.

Figura 56.

Tarjetas de los recursos asociados a una solicitud con diferentes estados.



Además, se conservaron elementos como los favoritos y el historial de carrito, de manera que la información marcada previamente se refleja también dentro de las solicitudes, manteniendo la coherencia en todo el recorrido del usuario.

Para evitar que la navegación se vuelva lenta en casos con múltiples solicitudes, la carga de información se diseñó de forma progresiva: se muestran primero los registros más recientes y se ofrecen controles para visualizar más a medida que el usuario lo necesita. Cuando ocurren

problemas, los mensajes se presentan con un lenguaje directo, acompañado de iconos y colores que orientan sin generar alarma; si no existen solicitudes, se muestra un aviso explicativo que informa la situación y sugiere los pasos siguientes. A medida que la pestaña evolucionó, se aplicaron mejoras orientadas a conservar la fluidez: mejoras en la carga de solicitudes y transiciones visuales suaves que mantienen la experiencia estable en distintos tipos de dispositivos.

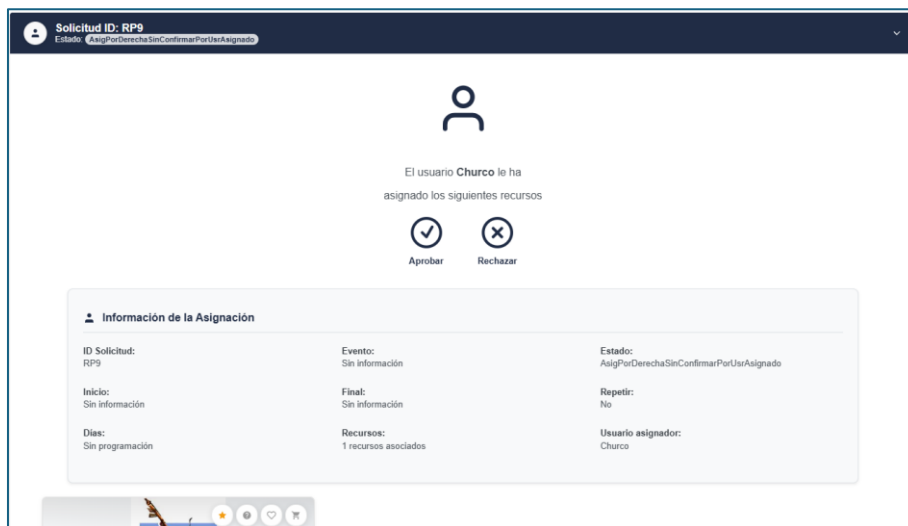
La evolución de esta pestaña ejemplifica cómo la construcción iterativa permitió articular un espacio complejo pero estable, que combina claridad visual, flexibilidad y retroalimentación constante para los usuarios.

5.1.3.5.4 Pestaña Asignaciones. La pestaña de Asignaciones se concibió como una extensión del sistema para manejar situaciones en las que ciertos usuarios con privilegios pueden asignar directamente recursos a otras personas. A diferencia del esquema tradicional de solicitudes, aquí el flujo parte de un asignador autorizado (como un profesor, coordinador o administrador) que distribuye recursos de manera proactiva. La construcción comenzó definiendo con claridad los roles y permisos que delimitan quién puede asignar a quién, un aspecto clave en entornos educativos u organizacionales donde la gestión centralizada es frecuente.

Como se puede apreciar en la Figura 57, una de las particularidades de esta pestaña es la aprobación bidireccional. No basta con que un recurso sea asignado: el usuario receptor debe confirmar si acepta o rechaza la asignación. Al recibirla, encuentra una interfaz centrada en la decisión, con un mensaje que indica quién hizo la asignación, qué recurso está involucrado y dos botones destacados para responder.

Figura 57.

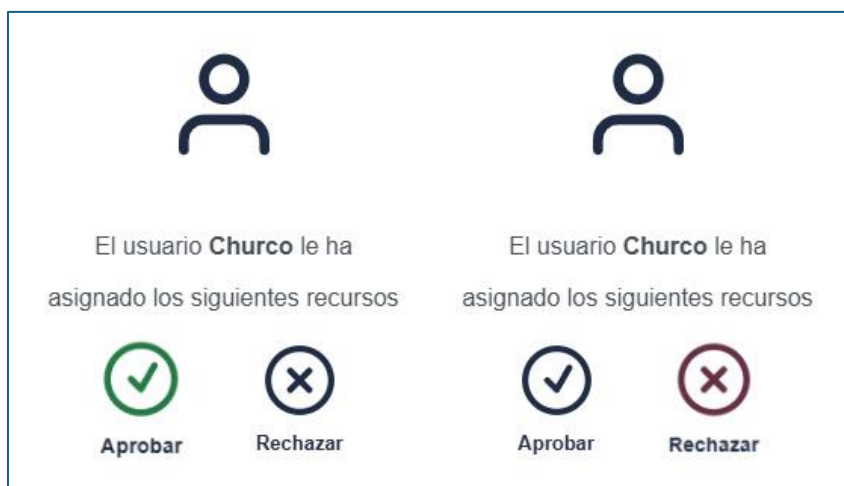
Vista de una solicitud recién asignada al usuario.



Como se puede apreciar en la Figura 58, estos botones ofrecen una retroalimentación visual inmediata: verde para aprobar, rojo para rechazar, lo que ayuda a que la acción sea clara y sin ambigüedades. Este doble control evita malentendidos y asegura que nadie cargue con responsabilidades sin haberlas aceptado.

Figura 58.

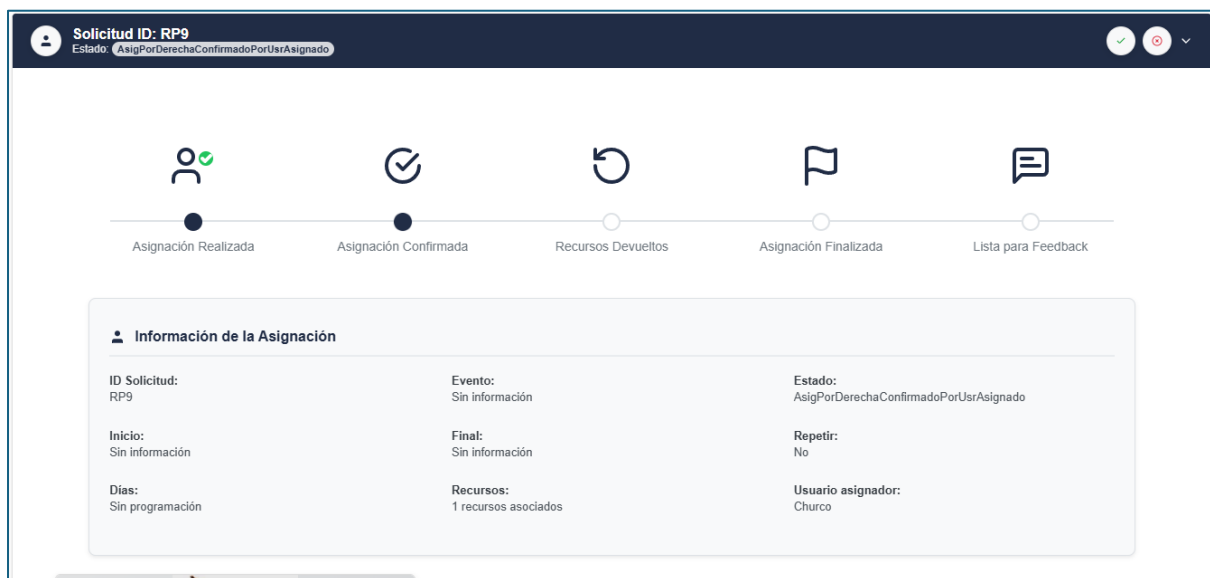
Vista de opciones que tiene el usuario al aprobar o rechazar una asignación.



Cuando la asignación se confirma, el sistema adapta la visualización al nuevo contexto. Como se puede apreciar en la Figura 59, el seguimiento mantiene la lógica de progreso, pero con un camino particular: asignación realizada, confirmación del receptor, devolución de recursos, cierre y retroalimentación final.

Figura 59.

Vista de la información de la asignación.



En caso de rechazo, la decisión se muestra de manera transparente sin implicar penalizaciones, y el asignador conserva la opción de reasignar los recursos a otra persona. También se contempló la posibilidad de cancelar asignaciones ya aceptadas, brindando flexibilidad al receptor si cambian sus circunstancias.

Cada tarjeta de asignación muestra además información contextual sobre quién realizó la acción inicial, lo que aporta transparencia y facilita la comunicación si es necesario resolver dudas o hacer ajustes. La visualización de los recursos asignados se mantiene consistente con la del resto del sistema: se presentan en tarjetas con sus descripciones, estados e indicadores habituales, respetando las preferencias del usuario como favoritos o elementos en carrito. Esto

asegura continuidad en la experiencia y evita que el usuario deba aprender un funcionamiento nuevo.

En paralelo, se incorporó lógica para prevenir conflictos entre asignaciones y solicitudes regulares, garantizando que no se sobrepasen capacidades ni se crucen horarios. Esta coordinación resultó esencial para evitar sobre reservas en escenarios donde múltiples tipos de operaciones recaen sobre los mismos recursos. Durante el desarrollo, la construcción iterativa permitió identificar y ajustar estos posibles choques desde etapas tempranas, lo que evitó problemas posteriores y reforzó la estabilidad general.

5.1.3.5.5 Pestaña Feedback. La pestaña Feedback se consolidó como el espacio destinado a recopilar la evaluación de las solicitudes procesadas, permitiendo a los usuarios calificar su experiencia y aportar comentarios sobre el uso de los recursos. Como se puede apreciar en la Figura 60, su diseño mantuvo la coherencia visual y funcional con el resto del sistema, privilegiando la claridad en la presentación de información y la sencillez en la interacción.

Figura 60.

Vista de la información de una solicitud finalizada.

Solicitud ID: RP9
Estado: Finalizado

Información de la Solicitud

ID Solicitud: RP9	Evento: Sin información	Estado: Finalizado
Inicio: Sin información	Final: Sin información	Repetir: No
Días: Sin programación	Recursos: 1 recursos asociados	

A continuación encontrará los usuarios propietarios y sus respectivos recursos.

Como se puede observar en la Figura 61, las solicitudes se muestran en tarjetas expandibles que agrupan los recursos asociados y exponen la información esencial: identificador, estado de la solicitud e icono para conversar con el o los propietarios. En los casos donde algunos recursos permanecen en proceso de cierre, la pestaña alerta al usuario de manera clara para conservar la trazabilidad de cada estado.

Figura 61.

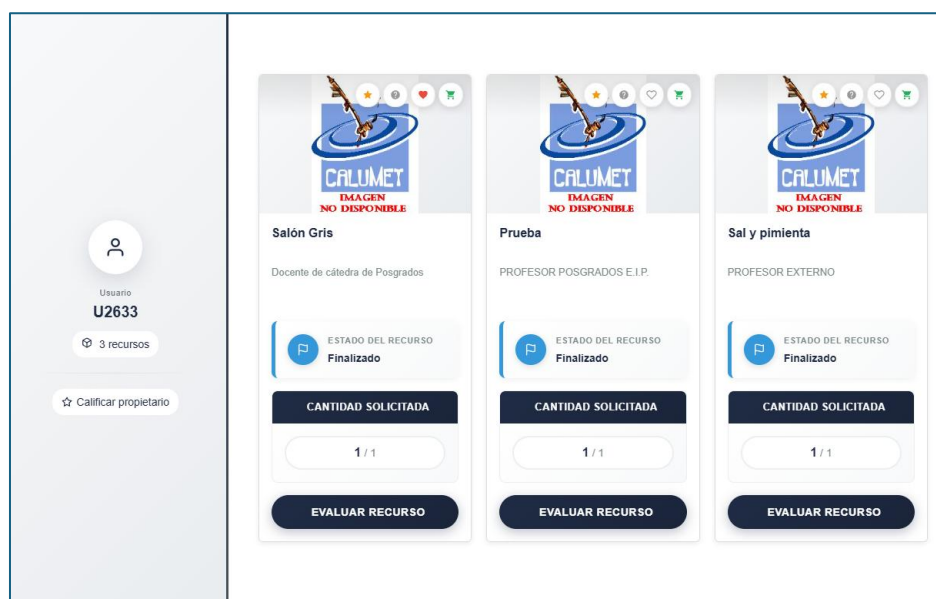
Vista de una solicitud finalizada.



Como se puede observar en la Figura 62, la información se organiza por propietario de recurso, lo que facilita una evaluación más precisa y permite asociar cada calificación directamente con la solicitud y con el usuario responsable.

Figura 62.

Vista de una solicitud finalizada de los recursos solicitados organizado por propietario.



A partir de esta estructura se integró un mecanismo de calificación que fortalece la construcción de un sistema de reputación confiable y transparente, el cual se puede apreciar tanto en la Figura 63 como en la Figura 64.

Figura 63.

Vista inicial que permite evaluar a un propietario.



The screenshot shows a web form titled "Evaluar Propietario" with the subtitle "Comparte tu experiencia para ayudar a otros usuarios". The form is divided into two main sections. The first section, "Calificación", prompts the user to "Selecciona una calificación del 1 al 10" and features a row of ten star icons numbered 1 through 10. The second section, "Comentario (Opcional)", prompts the user to "Comparte tu experiencia sobre el propietario..." and includes a large text input area. At the bottom of the form is a dark button labeled "Guardar Evaluación".

Figura 64.

Vista de la evaluación realizada por un solicitante a un recurso.

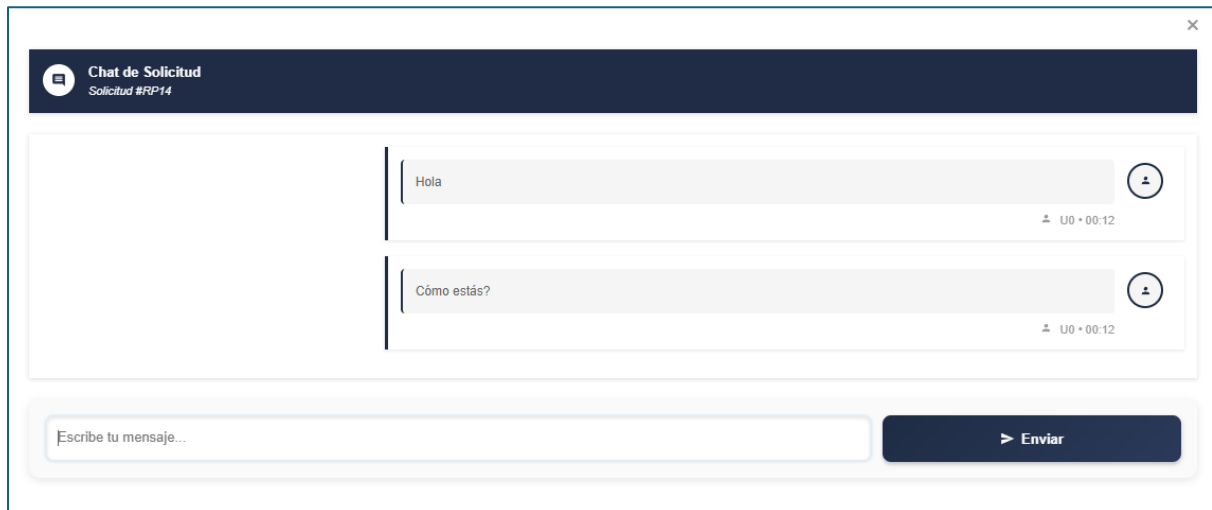


The screenshot shows a confirmation page titled "Tu Evaluación realizada al Recurso" with the subtitle "Calificación registrada exitosamente". The page features a large circular graphic on the left containing the number "10" and "/ 10", with ten yellow stars below it. To the right, there are two rectangular boxes: the first contains the number "10" and the word "CALIFICACIÓN", and the second contains a green checkmark icon and the word "EXCELENTE". Below this, a dark header section titled "Tu Comentario" with the subtitle "Tu experiencia registrada" is visible, followed by a text input field containing the word "Excelente".

Como se puede apreciar en la Figura 65, junto a las calificaciones se habilitó un canal de comunicación contextual a través de un chat vinculado a cada solicitud, lo que refuerza la interacción entre usuarios y propietarios manteniendo la trazabilidad de las conversaciones en un mismo lugar.

Figura 65.

Chat de una solicitud entre el solicitante y el o los propietarios.



Para preservar la fluidez incluso con grandes volúmenes de información, la pestaña implementa un sistema de carga progresiva que despliega solo lo necesario en cada momento y permite continuar explorando mediante paginación incremental. Esta mejora mantiene tiempos de respuesta ágiles y evita sobrecargar la interfaz.

El diseño de la pestaña se adapta a distintos dispositivos para mantener legibilidad y facilidad de uso, incorporando detalles visuales que hacen más clara la interacción. Se integra de forma coherente con las demás, reutilizando componentes como la tarjeta de recurso y conservando atributos personales del usuario, como favoritos y carrito. Además, en situaciones excepcionales, ofrece mensajes informativos que orientan al usuario y refuerzan la confiabilidad del sistema en su conjunto.

5.1.3.5.6 Pestañas de Favoritos y Carrito. Las pestañas de Favoritos y Carrito fueron concebidas sobre una misma base de diseño como se puede apreciar en la Figura 66 y 67, lo que permitió que su desarrollo siguiera una línea coherente en cuanto a estructura y experiencia de uso. Ambas comparten elementos centrales como la tarjeta de recurso para presentar la

información, un encabezado con icono distintivo y un contador dinámico que refleja en todo momento la cantidad de elementos almacenados. Cuando no contienen recursos, se muestra un mensaje con icono que orienta al usuario sobre cómo empezar a utilizarlas.

Figura 66.

Vista de la pestaña Carrito.

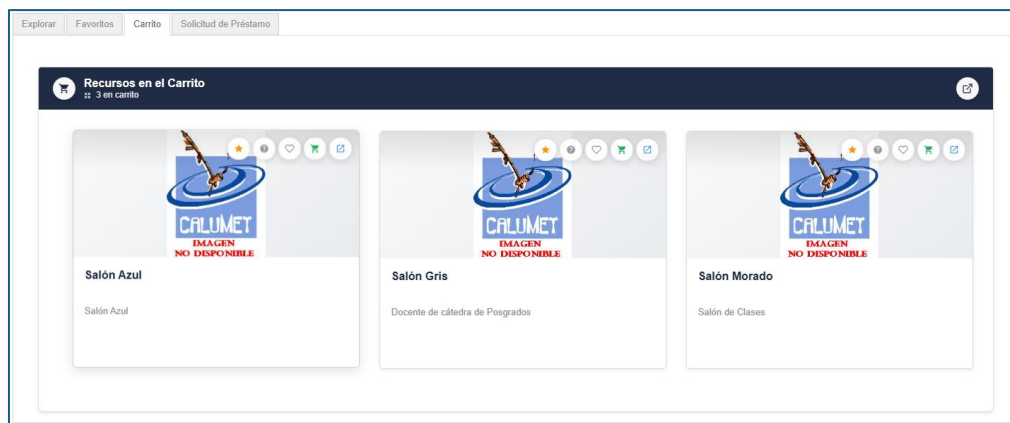
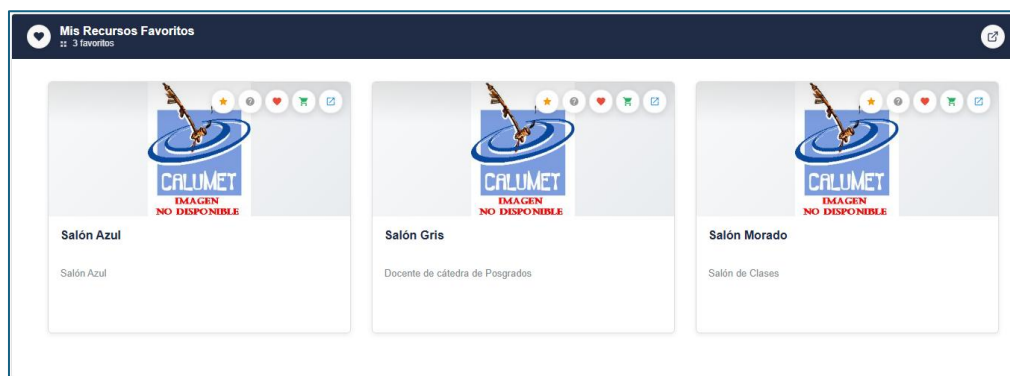


Figura 67.

Vista de la pestaña Favoritos.



Aunque mantienen una apariencia y funcionamiento muy similares, cada pestaña responde a un propósito específico. En Favoritos, los usuarios crean una colección personal de recursos de interés, mientras que en Carrito organizan aquellos que desean solicitar en préstamo de manera conjunta.

El diseño mantiene principios consistentes en ambas: claridad en la jerarquía de la información, retroalimentación visual inmediata y adaptación automática a distintos dispositivos, garantizando así una experiencia sencilla y uniforme en todo momento.

5.1.3.5.7 Pestaña Validación. La pestaña Validación se concibió como una herramienta exclusiva para administradores, destinada a revisar los recursos registrados y gestionar su estado de verificación. Para acceder a ella es necesario contar con perfil de administrador y tener habilitado el servicio correspondiente, lo que asegura que únicamente los usuarios autorizados puedan realizar este tipo de acciones.

Como se puede observar en la Figura 68, cada recurso se muestra en una tarjeta coherente con el resto del sistema, que incorpora un icono o botón para indicar si está validado. Desde allí, el administrador puede poner o quitar la validación según corresponda, de manera visible y directa. Este mecanismo no bloquea el uso de los recursos para otros usuarios, sino que ofrece una referencia clara de su validación actual.

Figura 68.

Vista de un recurso que aún no se ha validado.

Salón Azul
Información del Recurso

ID: RC1

CATEGORÍA: TR6

NOMBRE: Salón Azul

ESTADO: Activo

URL DE LA IMAGEN: /eisi/images/recursos/pordefecto/NoImage.JPG

USUARIO PROPIETARIO: U0

CANTIDAD DISPONIBLE: 1

ESTADO DE VERIFICACIÓN: No

ACCIÓN DE VERIFICACIÓN: Marcar como verificado

Para facilitar la gestión en casos de grandes volúmenes, se implementó nuevamente un sistema de paginación que organiza los elementos en bloques manejables y mantiene la fluidez de la navegación. Además, se añadieron criterios de filtrado que permiten segmentar la vista entre recursos validados, pendientes u otras condiciones operativas, agilizando la revisión.

Cuando no hay más elementos disponibles, la pestaña informa de manera explícita, evitando vacíos en la interacción y transmitiendo retroalimentación inmediata al administrador. Gracias a esta integración de seguridad, organización y coherencia visual, la pestaña de Validación asegura un control confiable sobre los recursos sin interrumpir la continuidad del sistema.

5.1.3.5.8 Pestaña de Parámetros. La pestaña Parámetros se diseñó como una herramienta administrativa clave para la gestión de categorías de recursos dentro del sistema. Su acceso está limitado a usuarios con perfil de administrador y servicio habilitado, garantizando que solo quienes cuentan con permisos adecuados puedan configurar este apartado.

Desde esta pestaña es posible crear nuevas categorías, editarlas, cambiar su estado y mantener un control ordenado de toda la información. Como se puede observar en la Figura 69, cada categoría se representa en una tarjeta que muestra su identificador, nombre, descripción y estado actual, con opciones de edición directa que permiten realizar ajustes sin necesidad de navegar a otras vistas.

Figura 69.

Vista individual de una categoría que el administrador puede editar o modificar.



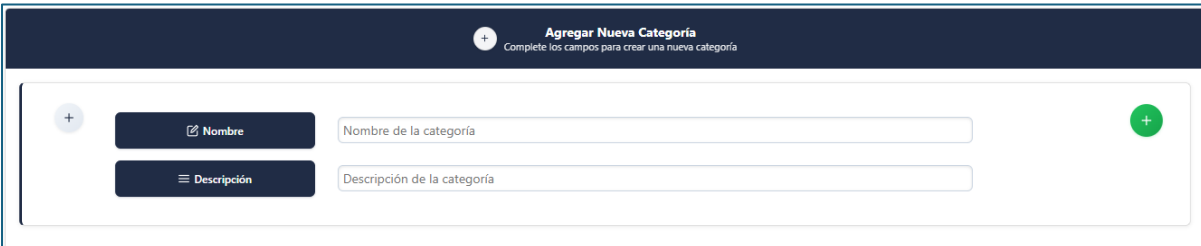
The screenshot shows a form for editing a category. On the left, there are four dark blue buttons with white text: 'ID Tipo', 'Nombre', 'Descripción', and 'Estado'. To the right of these buttons are input fields. The 'ID Tipo' field contains 'TR2'. The 'Nombre' field contains 'Video Been'. The 'Descripción' field contains 'Herramienta para exposiciones'. Below the 'Estado' button is a green pill-shaped button with a white checkmark and the text 'Activo'. In the top right corner, there are two circular icons: a green one with a document icon and an orange one with a trash can icon.

El sistema de estados sigue una lógica binaria (activo o inactivo) representada de manera visual mediante iconos y colores diferenciados, lo que permite identificar rápidamente la situación de cada categoría. Los botones de acción se han dispuesto de manera clara para separar responsabilidades: uno para guardar cambios en los campos editables y otro para alternar el estado, evitando así modificaciones accidentales.

La interfaz incorpora validaciones que aseguran la integridad de los datos ingresados y proporciona retroalimentación inmediata en cada acción. Además, cuando no existen categorías registradas, la pestaña muestra un mensaje contextual que orienta al administrador hacia la creación de la primera categoría, reduciendo la curva de aprendizaje como se puede observar en la Figura 70.

Figura 70.

Vista para agregar una nueva categoría.



The screenshot shows the 'Agregar Nueva Categoría' form. At the top, there is a dark blue header with a white plus icon, the text 'Agregar Nueva Categoría', and a subtitle 'Complete los campos para crear una nueva categoría'. Below the header, there are two dark blue buttons with white text: 'Nombre' and 'Descripción'. To the right of these buttons are input fields. The 'Nombre' field contains 'Nombre de la categoría'. The 'Descripción' field contains 'Descripción de la categoría'. In the top right corner, there is a green circular button with a white plus icon.

En su diseño se cuidó la coherencia visual con el resto del sistema: paleta de colores consistente, iconos reconocibles y transiciones sutiles que aportan claridad en la interacción. Gracias a esta integración de usabilidad, control y uniformidad, la pestaña de Parámetros se convierte en un soporte esencial para mantener organizado y actualizado el catálogo de categorías de recursos.

5.1.3.5.9 Pestaña Auditoría. La pestaña de Auditoría constituye una herramienta fundamental para los administradores, al centralizar la trazabilidad de las operaciones y garantizar transparencia en la gestión de recursos. Su acceso está restringido a usuarios con perfil administrativo y con el servicio habilitado, reforzando la seguridad en el manejo de datos sensibles.

Como se puede apreciar en la Figura 71, la información se presenta de forma jerárquica: cada solicitud funciona como contenedor de los distintos usos de recursos asociados, lo que permite seguir con claridad el ciclo de vida de una transacción desde su creación hasta su cierre. Dentro de cada registro se recopilan datos clave como fechas, usuarios involucrados, comentarios, evaluaciones entre solicitantes y propietarios, además de configuraciones de recurrencia en los casos donde se establecen patrones repetitivos. Este nivel de detalle aporta insumos valiosos para identificar patrones de uso, evaluar la calidad del servicio y detectar oportunidades de mejora en la operación.

Figura 71.*Vista de una solicitud.*

Solicitud ID: RP12 Asignado

Uso ID: UR2639

Información de la Solicitud

Solicitante: Sin información	Evento: Mega expo 2025	Asignador: Sin información
Fecha Solicitud: 2025-09-08 19:17:10	Inicio: 2025-09-08 08:00:00	Final: 2025-09-12 10:00:00
EvaluaSolicParaPropi: Sin información	EvaluaPropiParaSolic: Sin información	EvaluaSolicParaRecur: Sin información
ComenSolicParaPropi: Sin información	ComenPropiParaSolic: Sin información	ComenSolicParaRecur: Sin información

Días:

Lun Mié Vie

Información del Recurso

Recurso: RC1	Cantidad: 1
------------------------	-----------------------

Para facilitar la consulta de grandes volúmenes de datos, la pestaña incorpora nuevamente un sistema de carga progresiva que mantiene el rendimiento estable. Asimismo, cuando no se obtienen resultados, se despliegan mensajes contextuales que orientan al administrador en el ajuste de filtros.

El diseño responsivo asegura que la información sea accesible desde distintos dispositivos sin afectar la legibilidad ni la usabilidad. Gracias a esta combinación de trazabilidad, organización clara y herramientas de apoyo, la pestaña de Auditoría se consolida como un pilar para la confiabilidad y el control del sistema.

5.1.3.6 Evaluación y ajustes. Una vez habilitadas las funciones principales, se comenzó con pruebas en un entorno local sobre Windows. Este espacio permitió validar de forma rápida los flujos principales, pero al migrar el proyecto al servidor de pruebas (un entorno

Linux con un contenedor Docker y su propia base de datos) surgieron algunas diferencias que fue necesario ajustar.

En particular, MySQL en Linux aplica de forma estricta la distinción entre mayúsculas y minúsculas en nombres de tablas, columnas y rutas, lo que llevó a revisar y corregir varios llamados a la base de datos que en Windows pasaban inadvertidos. Este trabajo incluyó la normalización de nombres en consultas y la verificación de consistencia en los esquemas, evitando posibles fallos futuros al pasar a producción.

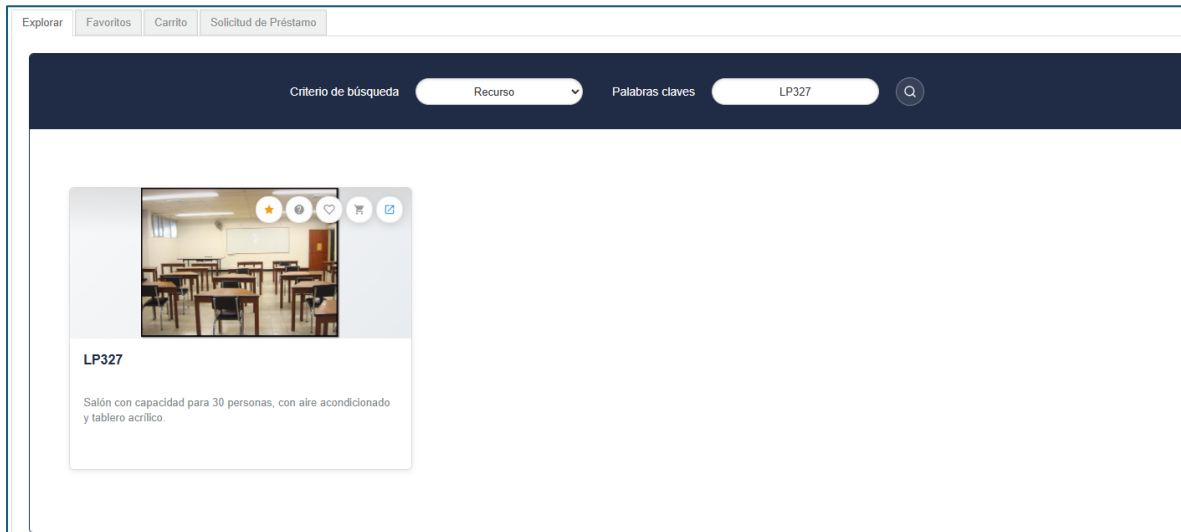
Con el entorno ya estabilizado, se llevaron a cabo pruebas basadas en historias de usuario. Estas pruebas se enfocaron en validar no solo la respuesta técnica del sistema, sino también la claridad y fluidez de cada interacción. Las observaciones detectadas se categorizaron según su impacto y facilidad de resolución, lo que permitió dar prioridad a los ajustes más relevantes.

El resultado fue un entorno de pruebas más cercano a las condiciones reales de despliegue, con mayor control sobre la base de datos y menor riesgo de errores relacionados con la infraestructura. Además, este proceso ayudó a pulir detalles visuales y de comportamiento, fortaleciendo la experiencia de uso en escenarios cotidianos.

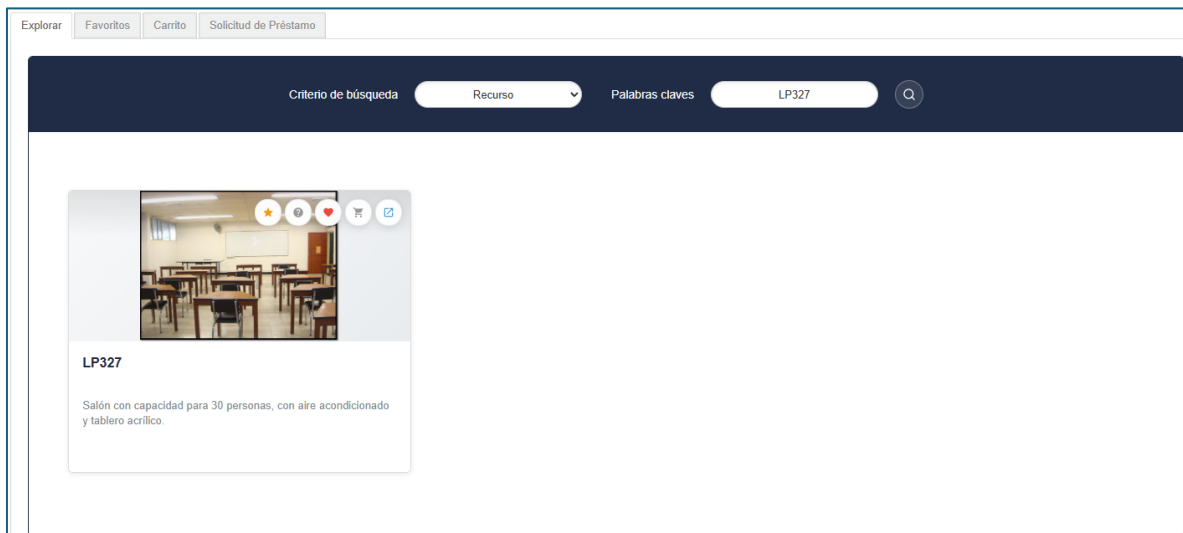
5.1.3.6.1 HU-01 Marcar recursos como favoritos. En el entorno de pruebas se verificó la funcionalidad de marcado y desmarcado de recursos como favoritos, con persistencia por usuario entre sesiones como se puede apreciar en la Figura 72 y 73. La gestión se apoya en la tabla TR_RecFavoritos y en los endpoints que permiten establecer, consultar y retirar elementos.

Figura 72.

Vista de la pestaña Explorar y se desmarca como favorito.

**Figura 73.**

Vista de la pestaña Explorar y se marca como favorito.

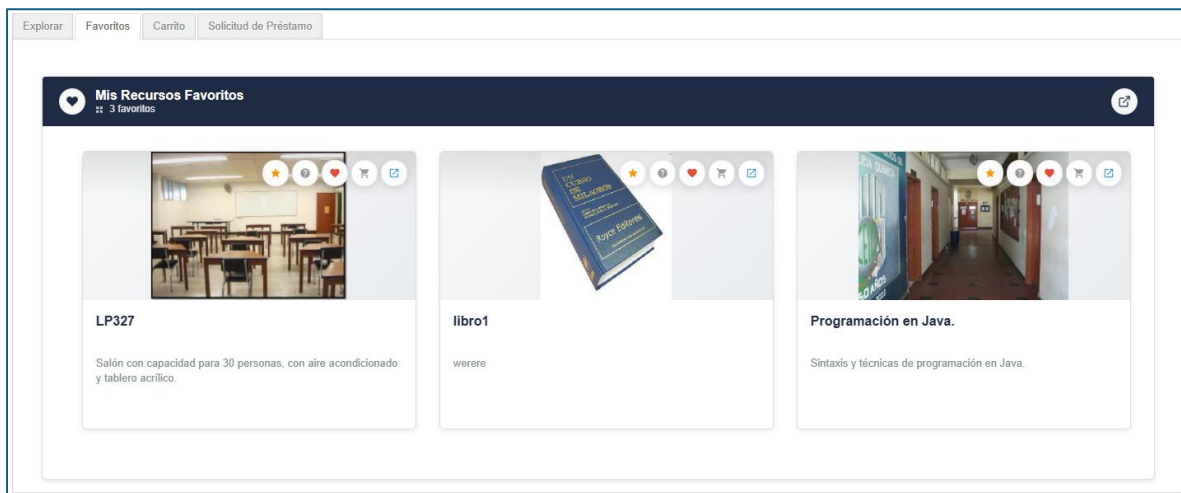


Se comprobó que el sistema solo permite seleccionar recursos activos y disponibles para el usuario, bloqueando los que no cumplen estas condiciones.

Como se puede apreciar en la Figura 74, las pruebas confirmaron la actualización consistente en la base de datos, la conservación del estado entre sesiones y la correcta visualización en la pestaña Mis favoritos, cumpliendo con los criterios definidos en la historia de usuario.

Figura 74.

Vista de la pestaña Favoritos después de agregar un nuevo recurso.



5.1.3.6.2 HU-02 Agregar recursos al carrito. En el entorno de pruebas se validó la gestión de un carrito persistente por usuario, sustentada en las tablas TB_RecCarrito y TR_RecCarItems. Como se puede observar en la Figura 75 y 76, los endpoints permiten registrar, consultar y retirar recursos del carrito según la operación requerida.

Figura 75.

Vista de la pestaña Explorar y se desmarca del carrito.

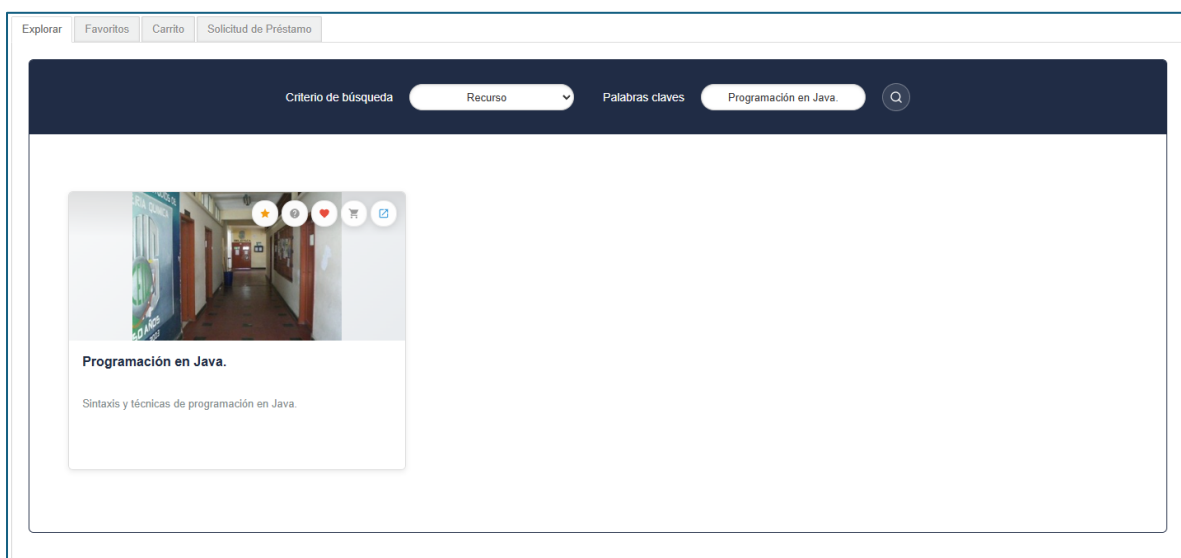
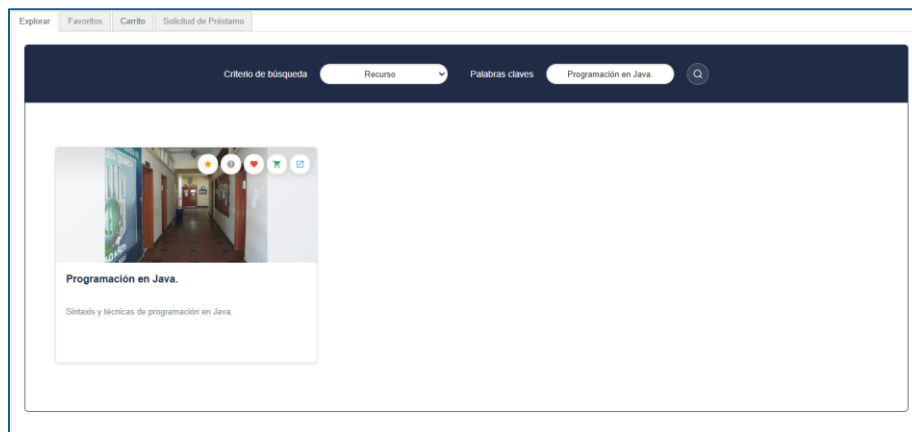


Figura 76.

Vista de la pestaña Explorar y se marca del carrito.

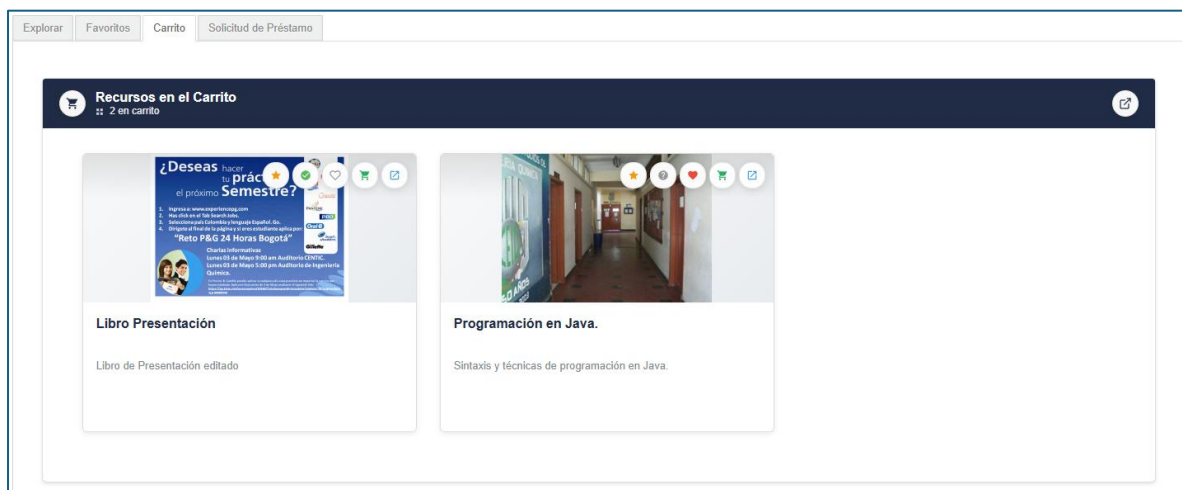


Se verificó que, antes de agregar un recurso, el sistema comprueba que esté activo en TP_Recurso y cuente con categorías habilitadas en TR_CategoriaRecurso, rechazando aquellos que no cumplen las condiciones.

Como se puede observar en la Figura 77, las pruebas confirmaron la persistencia del carrito entre sesiones, la actualización consistente en la base de datos y la correcta visualización en la pestaña Mi carrito, de acuerdo con los criterios establecidos en la historia de usuario.

Figura 77.

Vista de la pestaña Carrito después de agregar un nuevo recurso.



5.1.3.6.3 HU-03 Registro de solicitud de préstamo de recurso. En el entorno de pruebas se validó el registro de solicitudes de préstamo a partir de diferentes puntos de acceso (Mi carrito, Mis favoritos y Explorar) como se puede apreciar en la Figura 78 y 79. El proceso se soporta en las tablas TB_RecPedido y TR_UsoRecurso, donde cada solicitud queda registrada con estado inicial En Estudio.

Figura 78.

Vista desde diferentes pestañas para confirmar el uso del o los recursos.

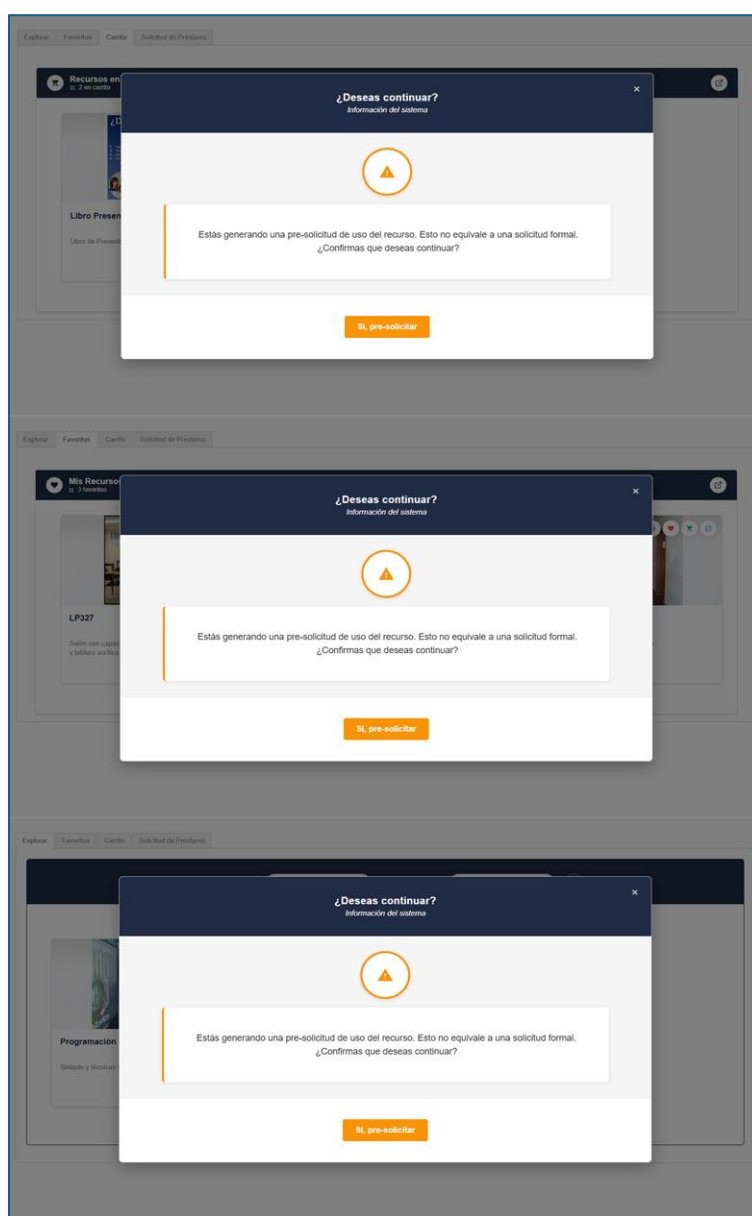
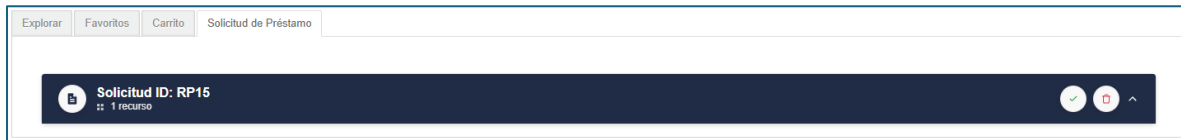


Figura 79.

Vista de la solicitud realizada desde la pestaña Explorar.



Como se puede apreciar en la Figura 80, se comprobó que el formulario de solicitud contempla los campos requeridos (fechas de inicio y fin, nombre del evento y repetición cuando aplica), y que el sistema ejecuta validaciones de disponibilidad sobre cantidad de recursos y cruces de horarios antes de confirmar. En caso de conflicto, se mostró la notificación correspondiente sin permitir avanzar como se puede apreciar en la Figura 81.

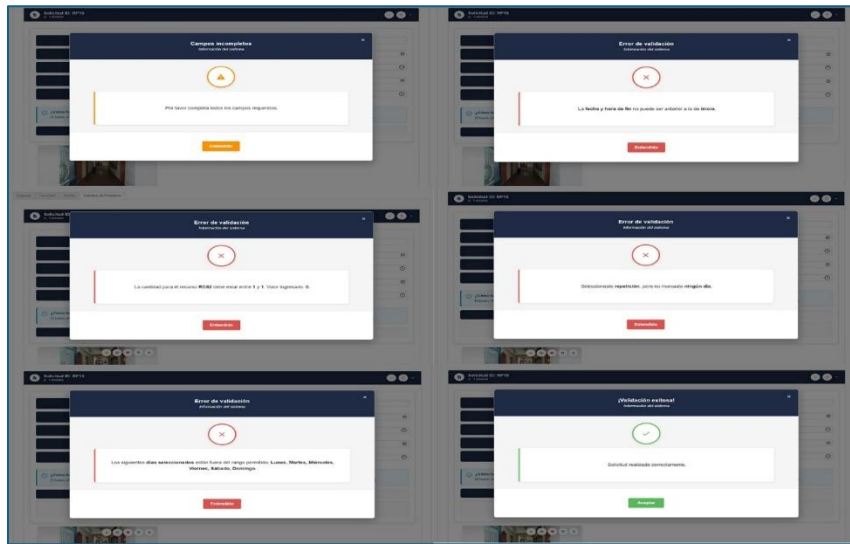
Figura 80.

Vista del formulario para realizar una solicitud.

This image shows the main form for creating a loan request. At the top, it has the same header as Figure 79. The form contains several input fields, each with a dark blue label on the left and a white input area on the right. The fields are: 'Nombre del evento' (with a star icon), 'Fecha de inicio' (with a calendar icon), 'Hora de inicio' (with a clock icon), 'Fecha de finalización' (with a calendar icon), and 'Hora de finalización' (with a clock icon). Below these fields is a light blue informational box with a question mark icon and the text '¿Cómo funciona la repetición? El botón «Repetir» permite reservar el recurso solo en los días que elijas dentro del rango de fechas. Si no lo activas, se asumirá que necesitas el recurso todos los días y durante todo el día.' At the bottom of the form, there is a 'Repetir' button and a toggle switch labeled 'Activar evento recurrente'.

Figura 81.

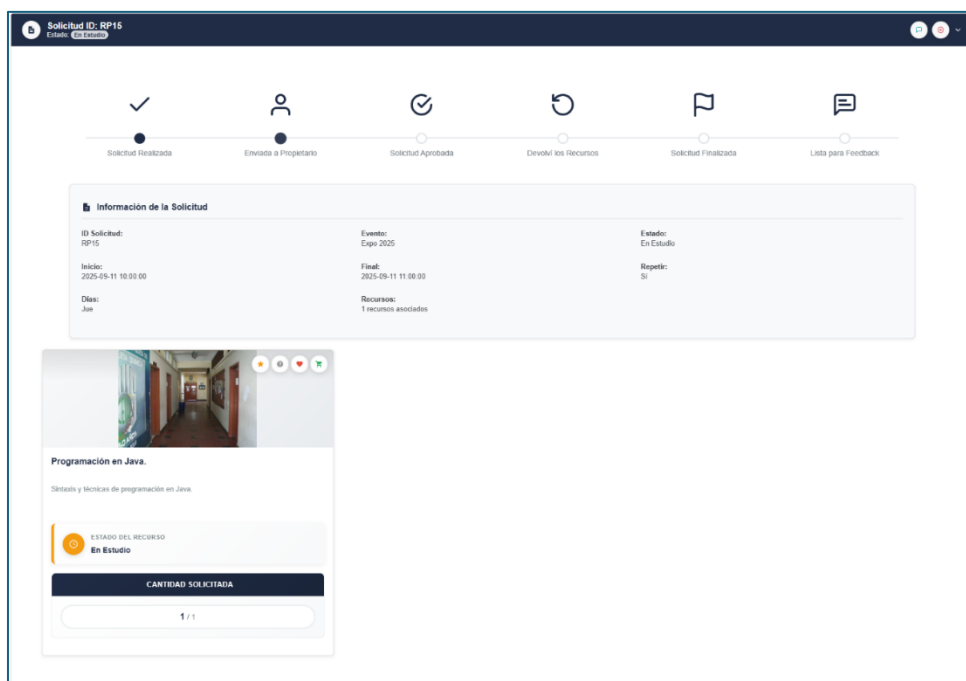
Notificaciones del sistema.



Como se puede apreciar en la Figura 82, las pruebas confirmaron la correcta creación del registro en base de datos, la asociación de los recursos seleccionados en TR_Usorecurso y la generación de un resumen con los datos de la solicitud, cumpliendo con los criterios de verificación establecidos en la historia de usuario.

Figura 82.

Solicitud creada previamente desde la pestaña Pre-solicitud.



5.1.3.6.4 HU-04 Consulta del estado de una solicitud de préstamo. Como se puede apreciar en la Figura 85, en el entorno de pruebas se validó la consulta del estado de las solicitudes desde la pestaña Solicitudes. El listado mostró correctamente identificador, fecha de registro, recursos asociados, rango de fechas y estado actual como se puede apreciar en la Figura 84.

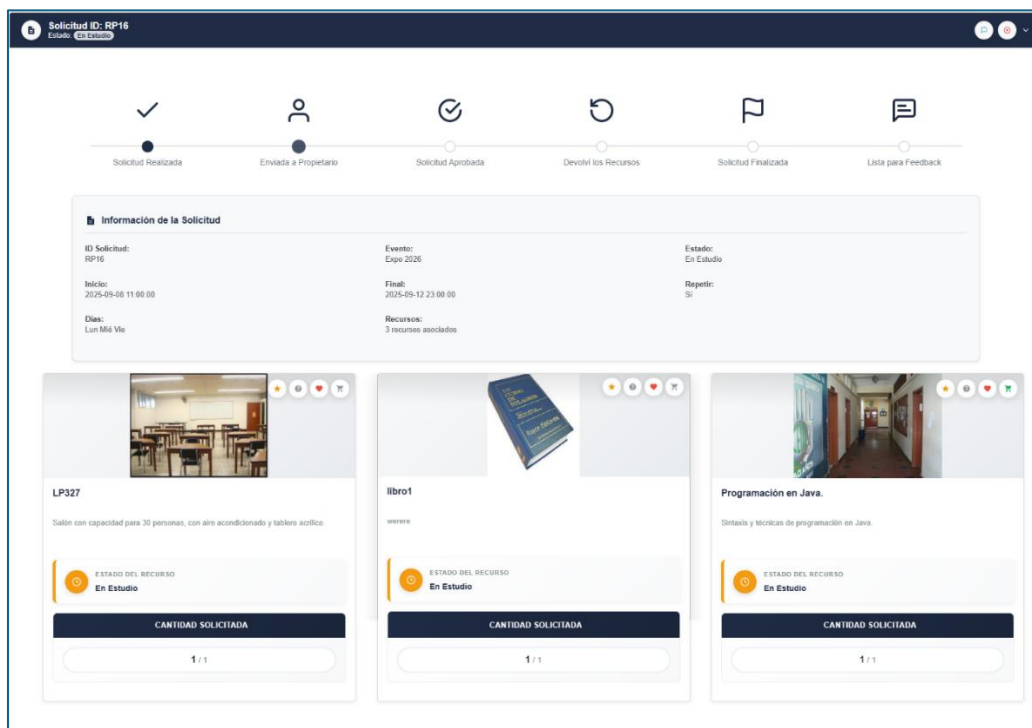
Figura 83.

Vista de la pestaña Solicitudes.



Figura 84.

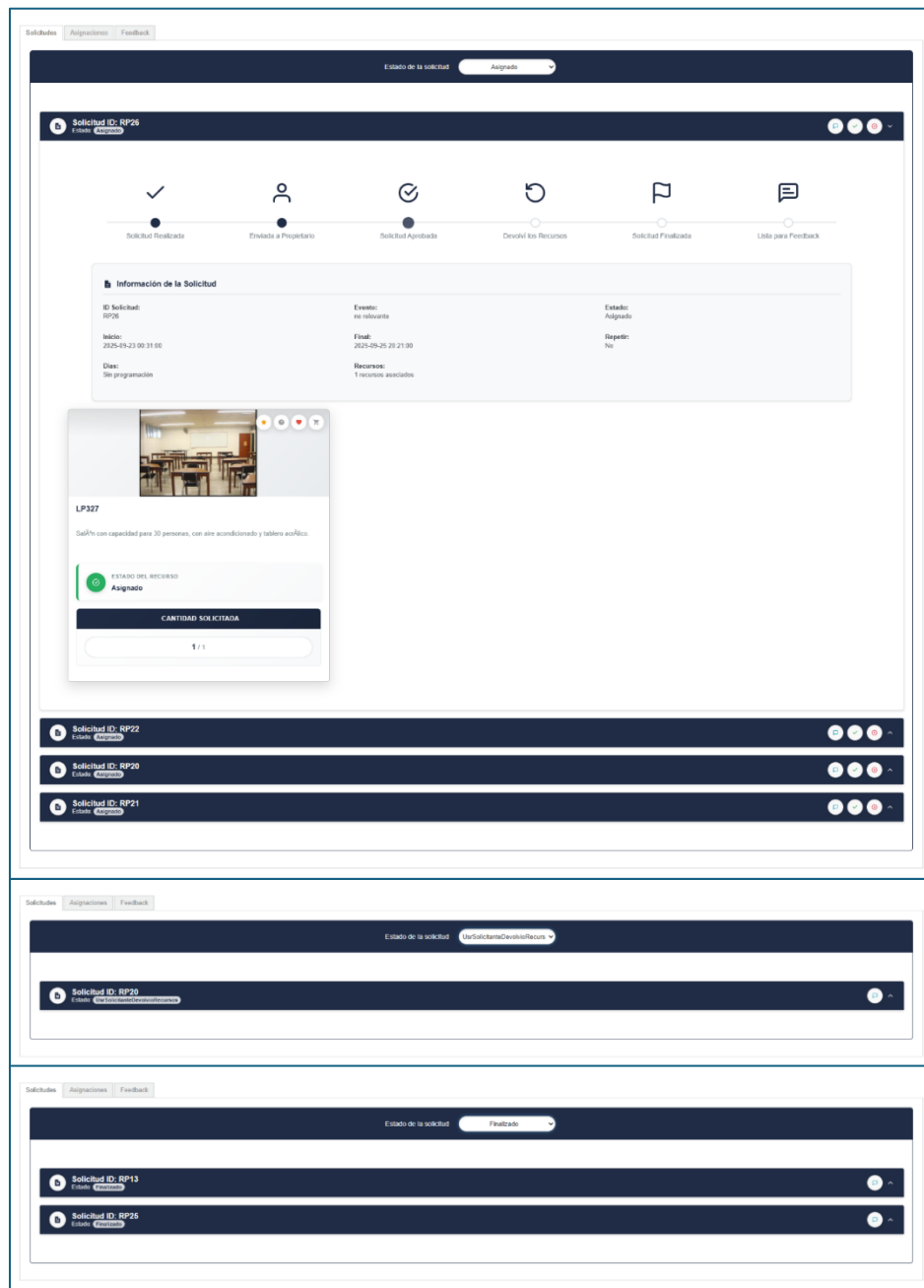
Vista de la información de una solicitud.



Como se puede apreciar en la Figura 85, se comprobó que los filtros por estado funcionan de acuerdo con los valores almacenados en TR_UsoRecurso.Estado (En Estudio, Asignado, Devolví Recursos y Finalizado), y que el acceso al detalle recupera la información registrada en TB_RecPedido junto con los recursos vinculados en TR_UsoRecurso.

Figura 85.

Vista de la pestaña Solicitudes con diferentes filtros de estado.

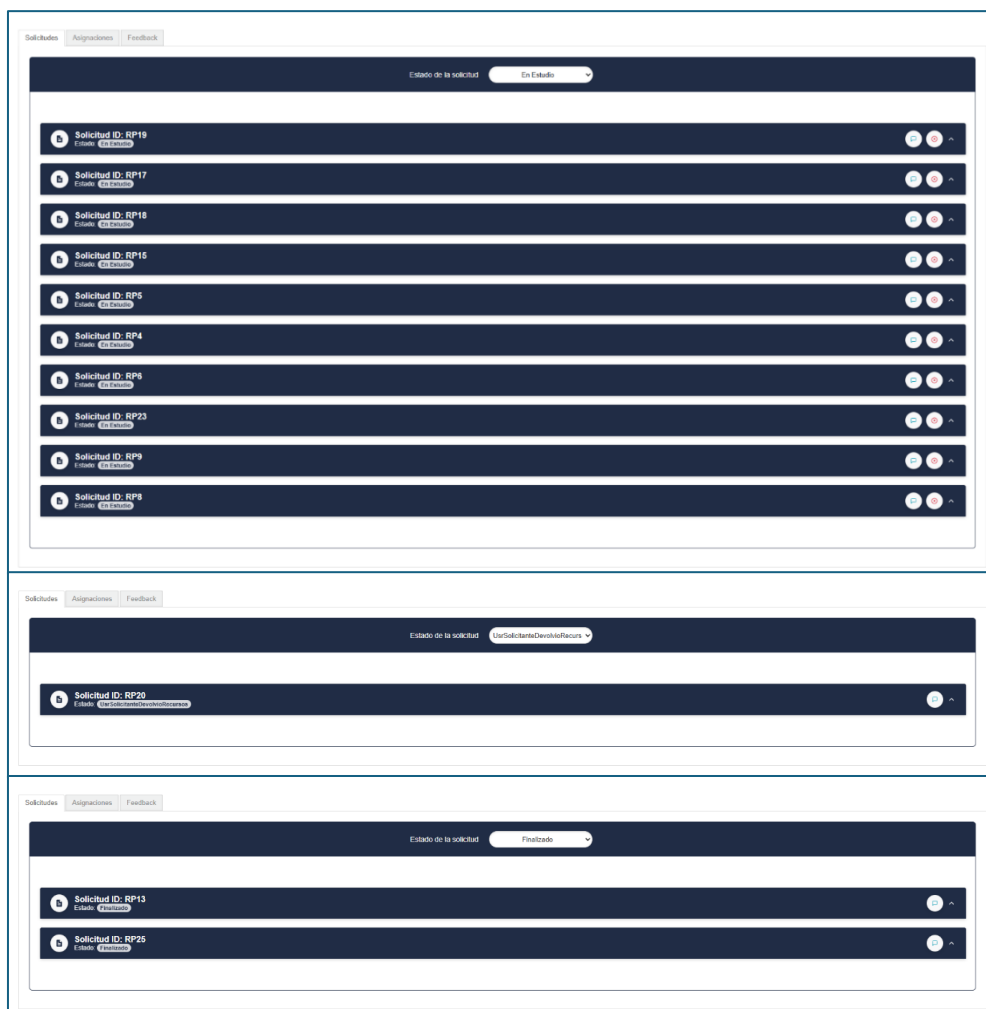


Las pruebas confirmaron que los estados reflejan fielmente la trazabilidad en base de datos y que el usuario puede dar seguimiento a sus solicitudes de manera clara y consistente, cumpliendo con los criterios establecidos en la historia de usuario.

5.1.3.6.5 HU-05 Cancelación de una solicitud antes de su aprobación. En el entorno de pruebas se verificó la cancelación de solicitudes cuyo estado en TR_UsorRecurso.Estado era En Estudio, confirmando que la acción no estaba disponible en estados distintos como se puede observar en la Figura 86.

Figura 86.

Vista de la pestaña solicitudes para cancelar.



Se comprobó que el sistema solicita confirmación explícita antes de ejecutar la cancelación como se puede observar en la Figura 87, y que al realizarla se actualiza el estado de la solicitud a Cancelada en la base de datos como se puede observar en la Figura 88. Las pruebas confirmaron además que estas solicitudes dejan de listarse como activas en la sección Mis solicitudes, pero permanecen registradas en TR_UsoRecurso para conservar la trazabilidad.

Figura 87.

Notificación del sistema para cancelar una solicitud.

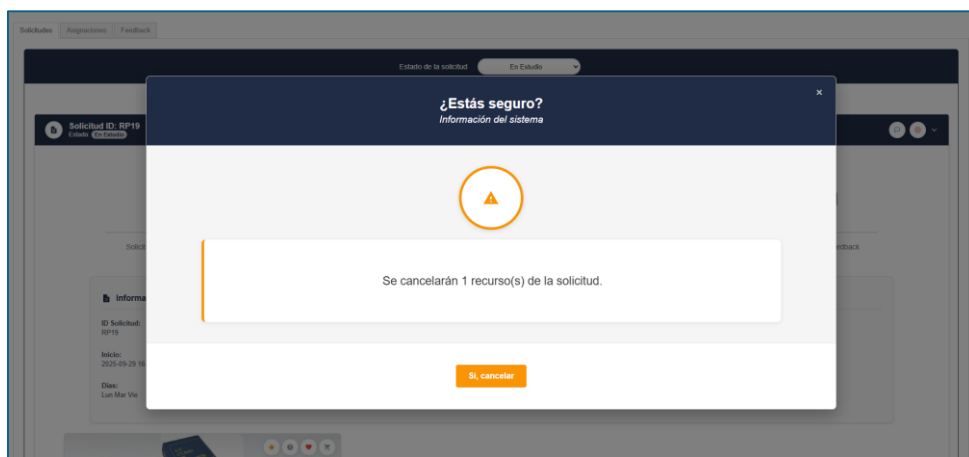
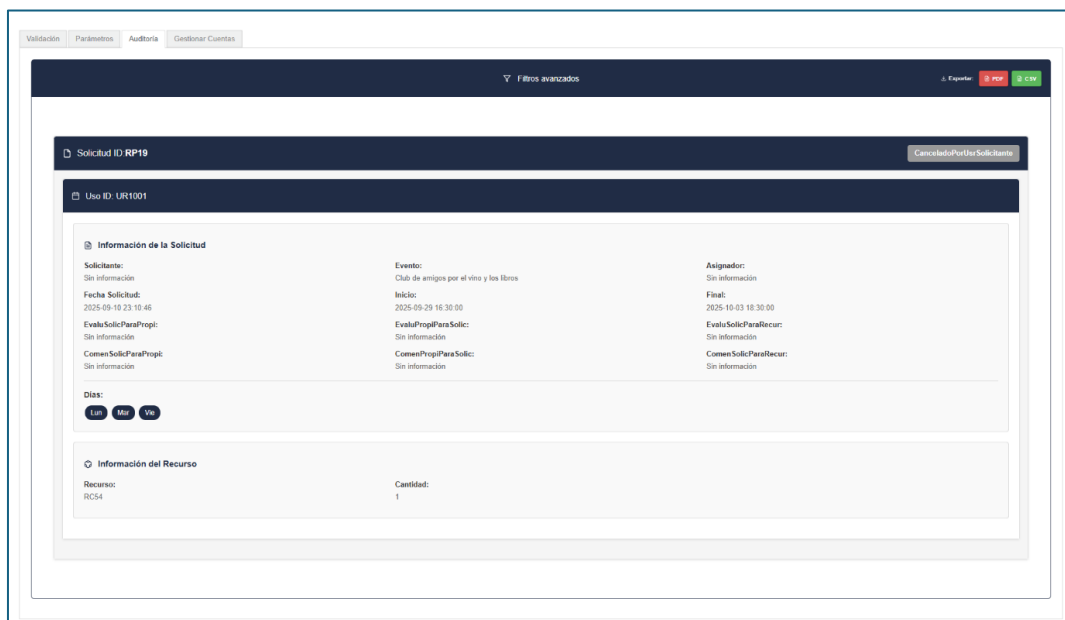


Figura 88.

Vista de la pestaña Auditoría.



De esta manera, se validó que la cancelación cumple con las reglas de negocio, garantiza un historial íntegro en base de datos y ofrece una vista limpia al usuario final, acorde con los criterios definidos en la historia de usuario.

5.1.3.6.6 HU-06 Filtrado de recursos disponibles. En el entorno de pruebas se verificó el filtrado de recursos por nombre, categoría, bodega y propietario. Se confirmó que el sistema permite mantener activo un único filtro a la vez, deshabilitando los demás antes de aplicar el nuevo criterio.

Las pruebas de búsqueda por nombre aceptaron coincidencias parciales sin distinción de mayúsculas o minúsculas como se puede apreciar en la Figura 89, mientras que los filtros por categoría, bodega y propietario recuperaron datos de acuerdo con las relaciones definidas en TR_CategoriaRecurso, TR_RecBodItems y TP_Recurso como se puede apreciar en la Figura 90. En todos los casos, los resultados incluyeron únicamente recursos activos con categorías también activas.

Figura 89.

Vista de la pestaña Explorar para búsqueda de recurso.

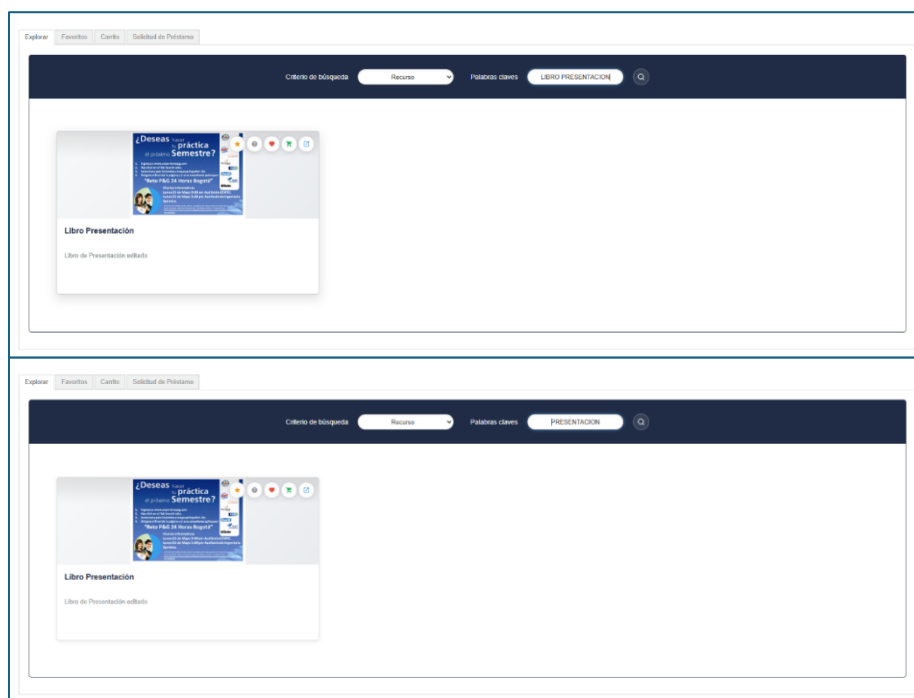
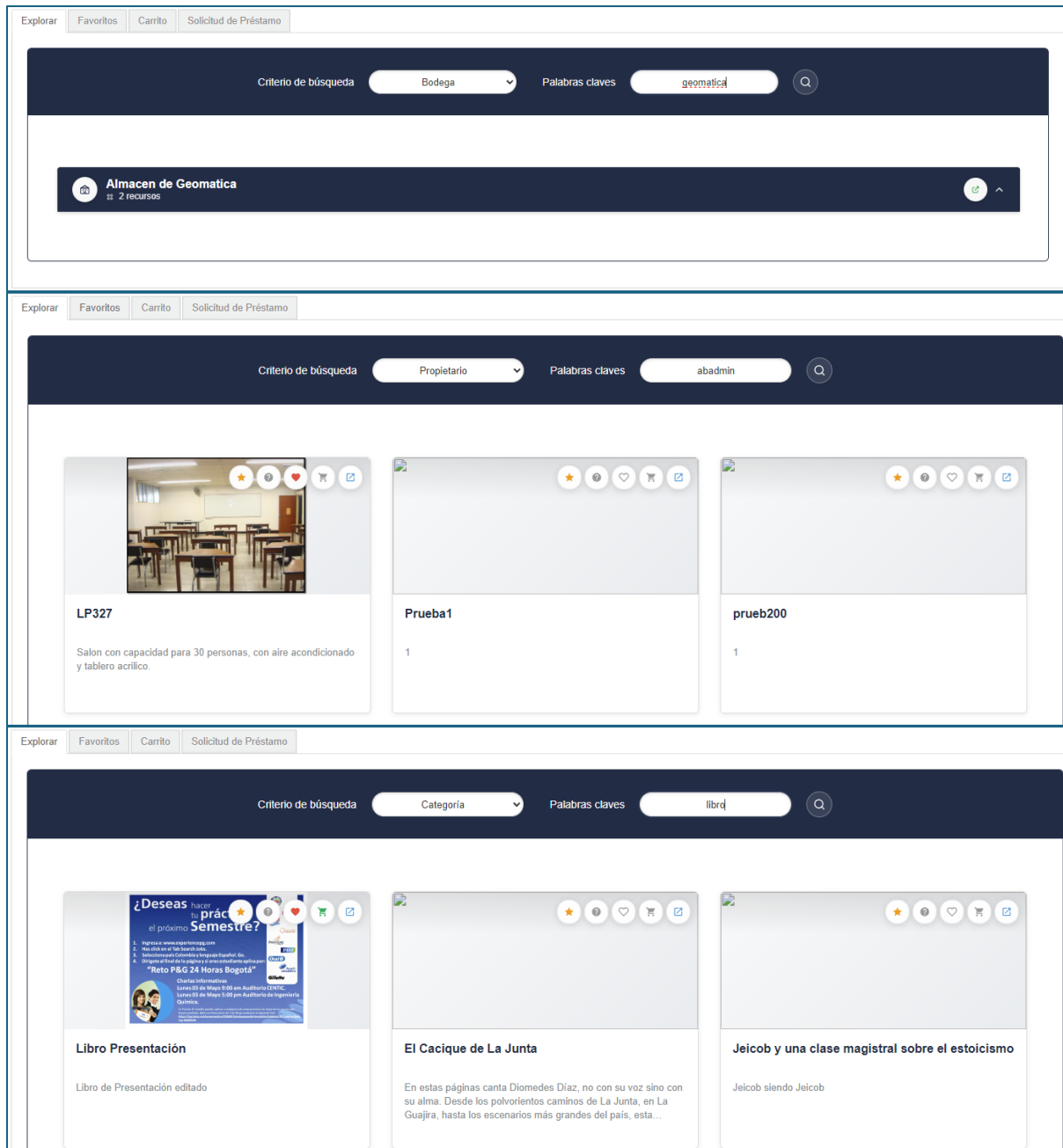


Figura 90.

Vista de la pestaña Explorar para búsqueda por bodega, categoría y propietario.



Se validó además que la lista de recursos se actualiza inmediatamente al aplicar o retirar un filtro, mostrando resultados consistentes con los criterios de verificación definidos en la historia de usuario.

5.1.3.6.7 HU-07 Visualización de detalles de un recurso. En el entorno de pruebas se verificó la visualización de recursos con información básica obtenida desde TP_Recurso, mostrando al menos el nombre, descripción y la imagen en la tarjeta como se puede observar en la Figura 91.

Figura 91.

Vista de la tarjeta del recurso en la pestaña Explorar.



Al acceder al detalle de un recurso, se confirmó la integración de un calendario en modo solo lectura, alimentado en tiempo real con los rangos de fechas aprobados o confirmados en TR_UsoRecurso. El calendario reflejó correctamente los períodos ocupados, diferenciando entre disponibilidad y reservas activas como se puede observar en la Figura 92 y 93.

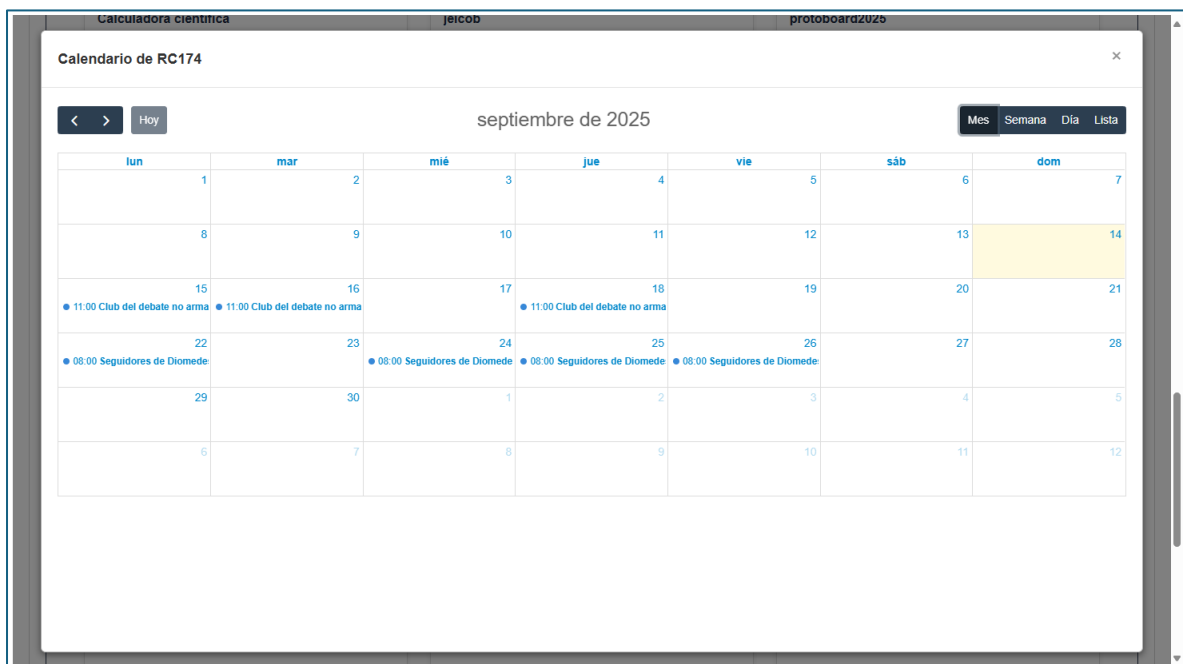
Figura 92.

Vista del calendario en modo lista.



Figura 93.

Vista del calendario en modo mes.



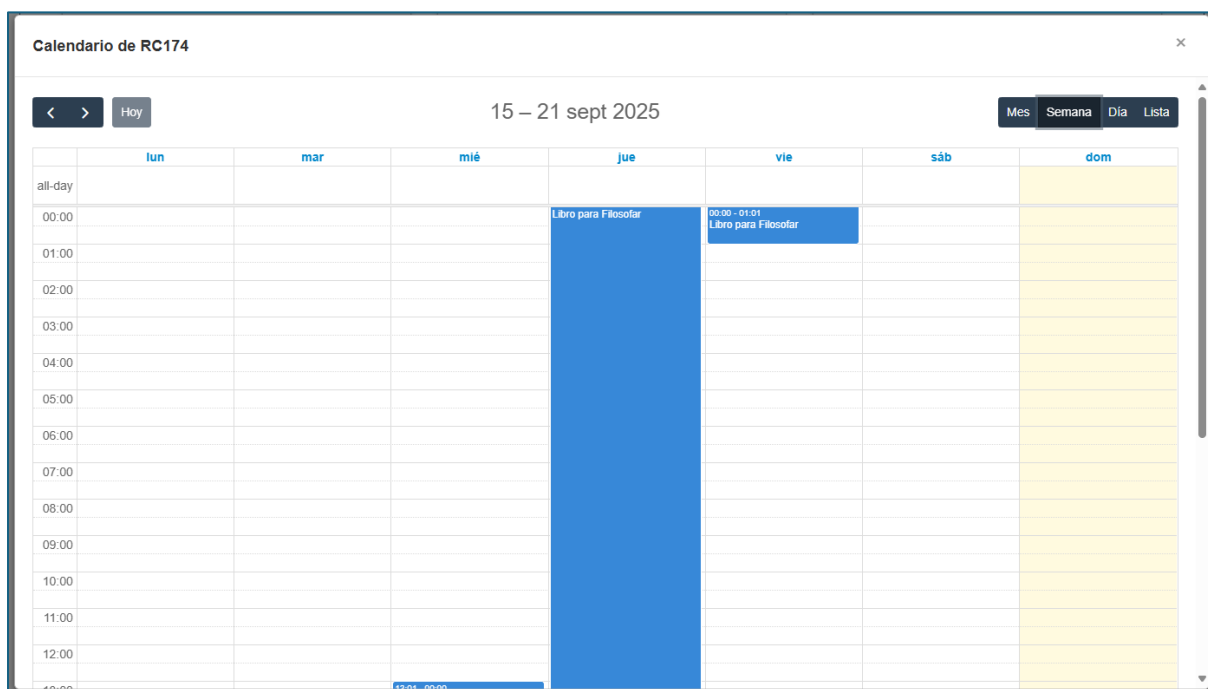
Las pruebas confirmaron que la información se recupera de la base de datos sin inconsistencias y que el calendario ofrece al solicitante una visión clara de la ocupación del recurso, cumpliendo con los criterios definidos en la historia de usuario.

5.1.3.6.8 HU-08 Seguimiento de disponibilidad de un recurso. En el entorno de pruebas se validó la consulta de disponibilidad a través de un calendario accesible desde la tarjeta del recurso.

Las pruebas confirmaron que el calendario refleja en tiempo real los periodos ocupados y los distingue claramente de los periodos libres. Se verificó además que la vista se comporta como un componente de solo lectura, sin permitir iniciar solicitudes ni modificar reservas desde el calendario como se puede observar en la Figura 94.

Figura 94.

Vista del calendario en modo semana.



De esta manera, se comprobó que la funcionalidad ofrece al solicitante una herramienta confiable para planificar con anticipación, cumpliendo con los criterios de verificación definidos en la historia de usuario.

5.1.3.6.9 HU-09 Notificaciones por cambios en una solicitud. En el entorno de pruebas se verificó el sistema de notificaciones por correo ante cambios de estado en las solicitudes.

Durante las pruebas se confirmó que cada modificación registrada en TR_UsoRecurso.Estado activa el envío automático de un mensaje al propietario, con la información mínima requerida: identificador de la solicitud, nombre del evento (cuando aplica) y el nuevo estado asignado. El sistema consultó de manera correcta la dirección de correo en TP_Usuario y generó los mensajes a partir de una plantilla predefinida, asegurando uniformidad en el contenido como se puede apreciar en la Figura 95.

Figura 95.

Vista de la tabla TP_Correo.

	IdCorreo	Email	Asunto	Mensaje
1	446.240	caluwebapp@gmail.com	EISiWeb Solicitud de Calificador por Hoover Tibursio Rueda Tocancipa	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
2	446.241	caluwebapp@gmail.com	EISiWeb Solicitud de Calificador por Hoover Tibursio Rueda Tocancipa	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
3	446.242	caluwebapp@gmail.com	EISiWeb Solicitud de Calificador por Hoover Tibursio Rueda Tocancipa	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
4	446.243	caluwebapp@gmail.com	EISiWeb Solicitud de Calificador por Hoover Tibursio Rueda Tocancipa	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
5	446.244	caluwebapp@gmail.com	EISiWeb Solicitud de Calificador por Hoover Tibursio Rueda Tocancipa	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
6	446.245	caluwebapp@gmail.com	EISiWeb Solicitud de Calificador por Hoover Tibursio Rueda Tocancipa	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
7	446.246	caluwebapp@gmail.com	EISiWeb Solicitud de Calificador por Hoover Tibursio Rueda Tocancipa	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
8	446.247	caluwebapp@gmail.com	EISiWeb Solicitud de Calificador por Hoover Tibursio Rueda Tocancipa	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
9	446.248	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
10	446.249	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
11	446.250	caluwebapp@gmail.com	Módulo de préstamo de recursos - Nueva solicitud de préstamo (ID: RP55)	Estimado(a) propietario(a) del recurso, Ha recibido una nueva solicitud c
12	446.251	caluwebapp@gmail.com	Módulo de préstamo de recursos - Devolución de recurso (ID: RP55)	Estimado(a) propietario(a) del recurso, Le informamos que se ha realizad
13	446.252	caluwebapp@gmail.com	Módulo de préstamo de recursos - Nueva solicitud de préstamo (ID: RP58)	Estimado(a) propietario(a) del recurso, Ha recibido una nueva solicitud c
14	446.253	caluwebapp@gmail.com	Módulo de préstamo de recursos - Devolución de recurso (ID: RP58)	Estimado(a) propietario(a) del recurso, Le informamos que se ha realizad
15	446.254	caluwebapp@gmail.com	Módulo de préstamo de recursos - Devolución de recurso (ID: RP59)	Estimado(a) propietario(a) del recurso, Le informamos que se ha realizad
16	446.255	caluwebapp@gmail.com	Módulo de préstamo de recursos - Nueva solicitud de préstamo (ID: RP60)	Estimado(a) propietario(a) del recurso, Ha recibido una nueva solicitud c
17	446.256	caluwebapp@gmail.com	Módulo de préstamo de recursos - Nueva solicitud de préstamo (ID: RP61)	Estimado(a) propietario(a) del recurso, Ha recibido una nueva solicitud c
18	446.257	caluwebapp@gmail.com	Módulo de préstamo de recursos - Nueva solicitud de préstamo (ID: RP62)	Estimado(a) propietario(a) del recurso, Ha recibido una nueva solicitud c
19	446.258	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
20	446.259	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
21	446.260	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
22	446.261	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
23	446.262	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
24	446.263	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
25	446.264	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
26	446.265	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
27	446.266	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
28	446.267	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
29	446.268	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="
30	446.269	caluwebapp@gmail.com	Mensaje Proyectos Local	<table width="100%" style="width:100%;background:#F7F8F9;" ><tr><td style="

En esta figura se observa el procesamiento del job o proceso en segundo plano encargado de la distribución automática de mensajes.

Se comprobó además que las notificaciones se enviaron de forma inmediata tras el cambio de estado y que fueron recibidas exitosamente en la cuenta del propietario, cumpliendo con los criterios de verificación de la historia de usuario.

5.1.3.6.10 HU-10 Evaluación de un recurso después del uso. En el entorno de pruebas se validó la funcionalidad de evaluación de recursos posterior al préstamo.

Se comprobó que la opción de evaluación únicamente se habilita para solicitudes registradas en TR_UsosRecurso con estado Finalizado y sin evaluaciones previas como se puede apreciar en la Figura 96, cumpliendo la restricción de una sola evaluación por préstamo. El formulario exigió correctamente una calificación obligatoria en escala de 1 a 10, permitiendo de forma opcional un comentario adicional como se puede apreciar en la Figura 97.

Figura 96.

Vista de la pestaña Feedback

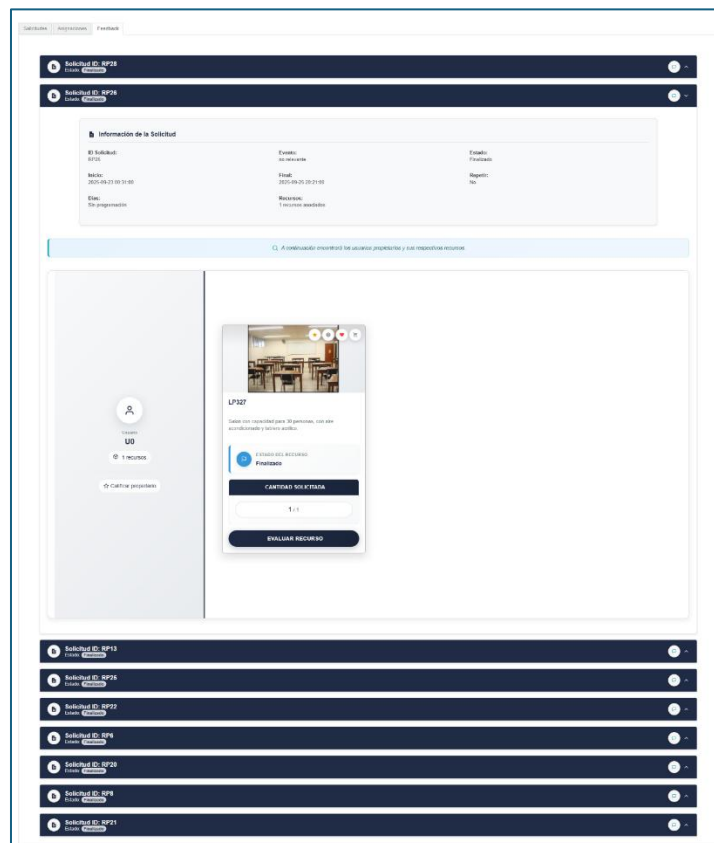
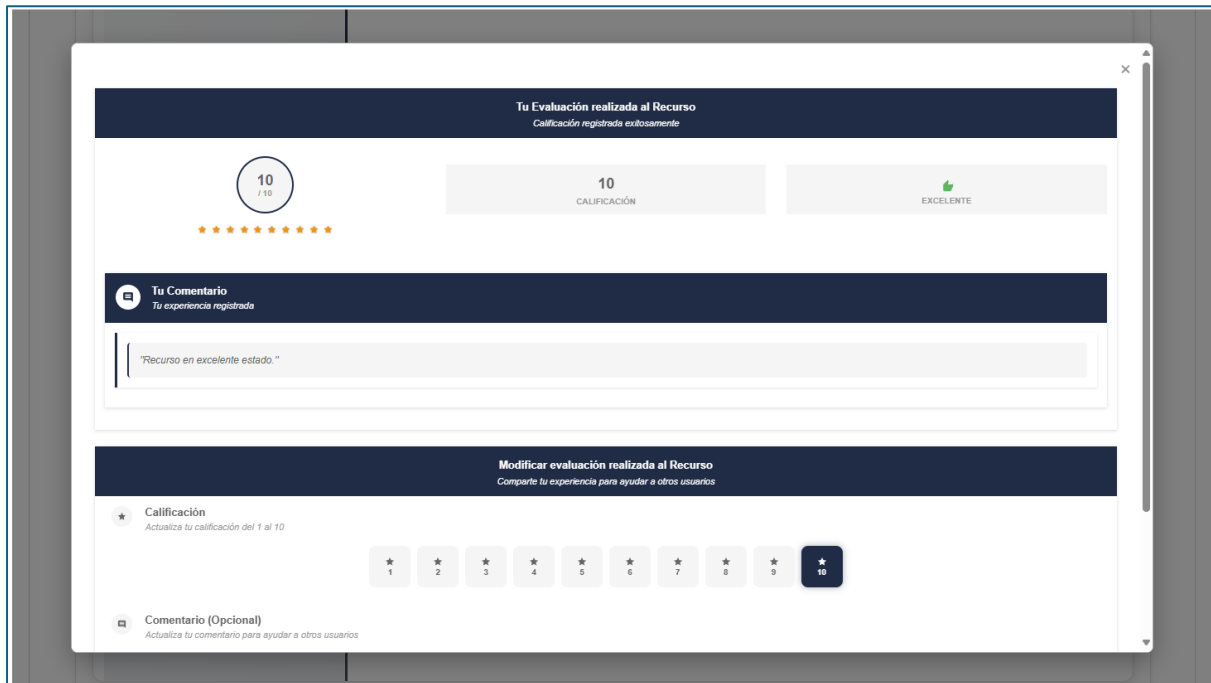


Figura 97.

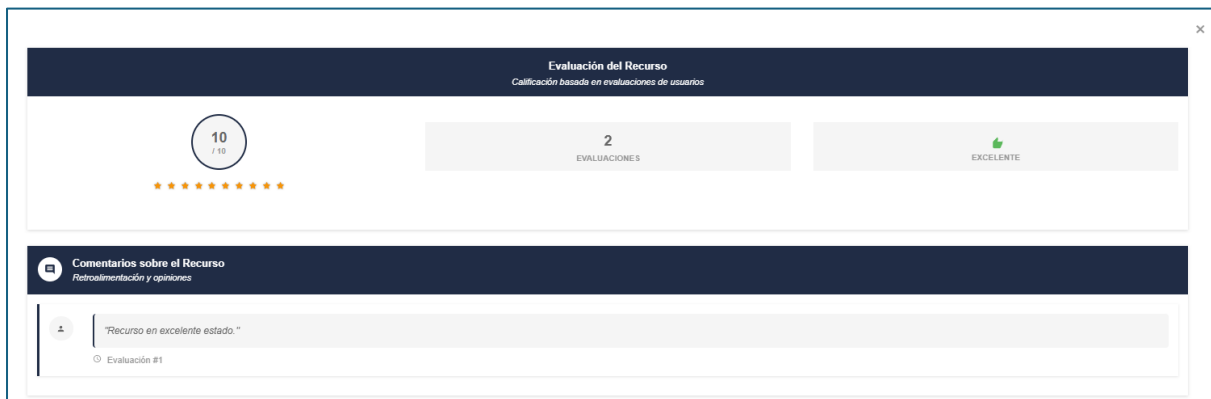
Vista de la calificación realizada al recurso.



Las evaluaciones registradas quedaron almacenadas en la base de datos con la relación adecuada entre TR_UsorRecurso, TP_Recurso y TP_Usuario, y fueron reflejadas en el historial del recurso, garantizando trazabilidad y consulta posterior como se puede apreciar en la Figura 98.

Figura 98.

Vista del historial de calificaciones del recurso.

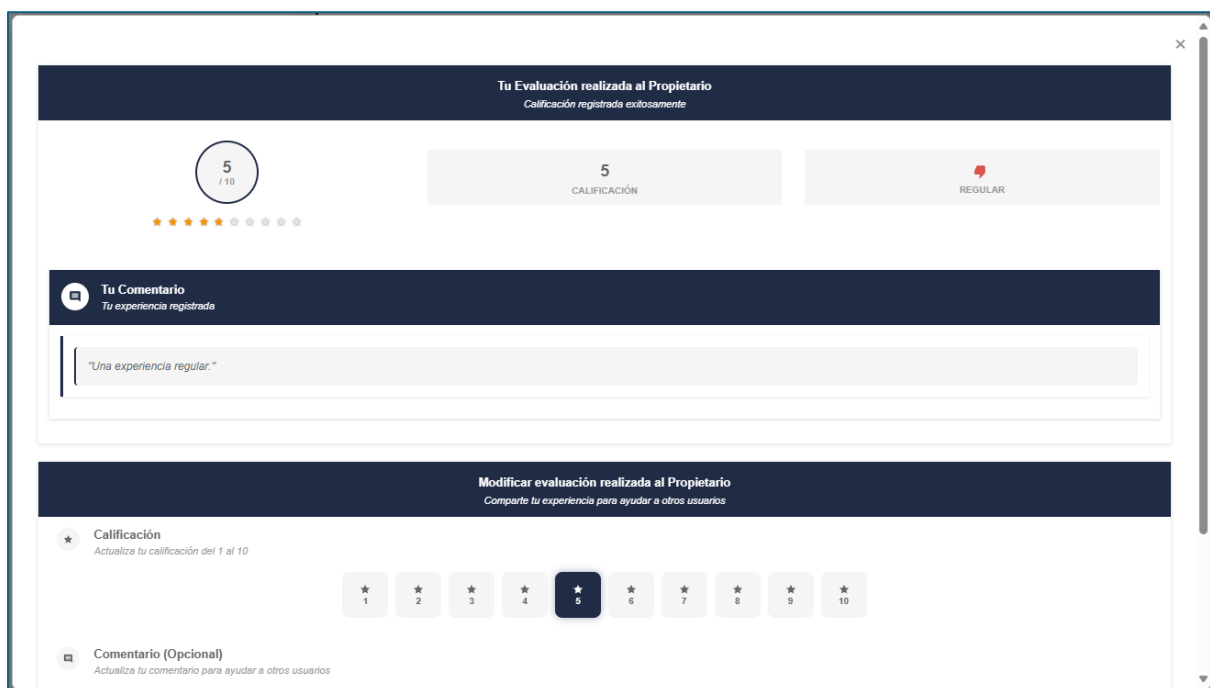


5.1.3.6.11 HU-11 Evaluación del propietario del recurso. En el entorno de pruebas se validó la funcionalidad de evaluación del propietario tras la finalización de un préstamo.

Se verificó que la opción de evaluación solo se habilite para solicitudes en TR_UsoRecurso con estado Finalizado como se puede observar en la Figura 97, garantizando además que cada solicitante solo pueda registrar una evaluación por préstamo y propietario. El formulario exigió correctamente una calificación obligatoria y aceptó comentarios opcionales, aplicando las validaciones definidas como se puede apreciar en la Figura 99.

Figura 99.

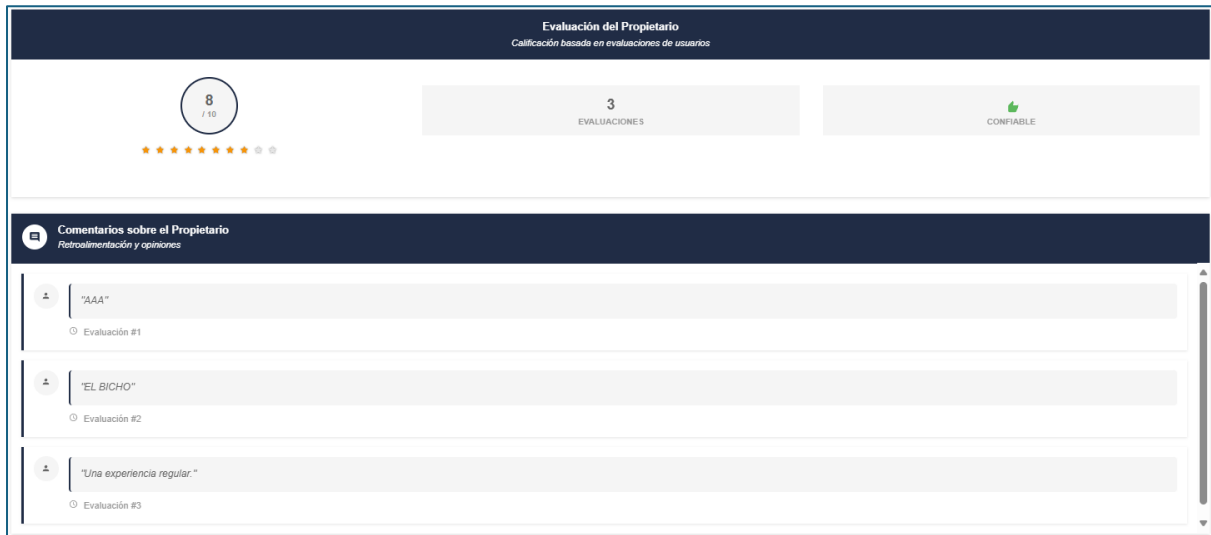
Vista de la calificación realizada al propietario del recurso.



Las evaluaciones quedaron almacenadas en la base de datos con la relación correspondiente entre TR_UsoRecurso, TP_Usuario y el propietario registrado en TP_Recurso. Estas evaluaciones quedaron disponibles para consulta según los permisos configurados, asegurando trazabilidad y coherencia en la retroalimentación hacia los propietarios como se puede apreciar en la Figura 100.

Figura 100.

Vista del historial de calificaciones del propietario.



5.1.3.6.12 HU-12 Historial personal de solicitudes y asignaciones. En el entorno de pruebas se validó la correcta implementación del historial personal de solicitudes y asignaciones.

Se comprobó que el acceso esté disponible desde el menú principal y que la consulta recupere todos los registros vinculados al solicitante en TR_UsoRecurso. El listado mostró datos completos, incluyendo identificador de la solicitud, fechas de registro e inicio/fin, recursos asociados, propietario y estado final como se puede apreciar en la Figura 84 y 101.

Figura 101.

Vista de la información de una asignación.

Solicitudes Asignaciones Feedback

Estado de la asignación Todos los estados

Solicitud ID: RP17 Estado: AsigPorDerechaUsrConfirmadoPorUsrAsignado

Solicitud ID: RP18 Estado: AsigPorDerechaUsrConfirmadoPorUsrAsignado

Solicitud ID: RP33 Estado: AsigPorDerechaUsrAsignadoDevolvióRecursos

Asignación Realizada Asignación Confirmada Recursos Devueltos Asignación Finalizada Lista para Feedback

Información de la Asignación

ID Solicitud: RP33	Evento: Sin información	Estado: AsigPorDerechaUsrAsignadoDevolvióRecursos
Inicio: 2025-09-28 12:25:00	Final: 2025-10-04 12:25:00	Repetir: No
Días: Sin programación	Recursos: 1 recurso asociados	Usuario asignador: UI

LP327

Salen con capacidad para 30 personas, con aire acondicionado y tablero acrílico.

CANTIDAD SOLICITADA

0 / 1

Solicitud ID: RP31 Estado: AsigPorDerechaUsrConfirmadoPorUsrAsignado

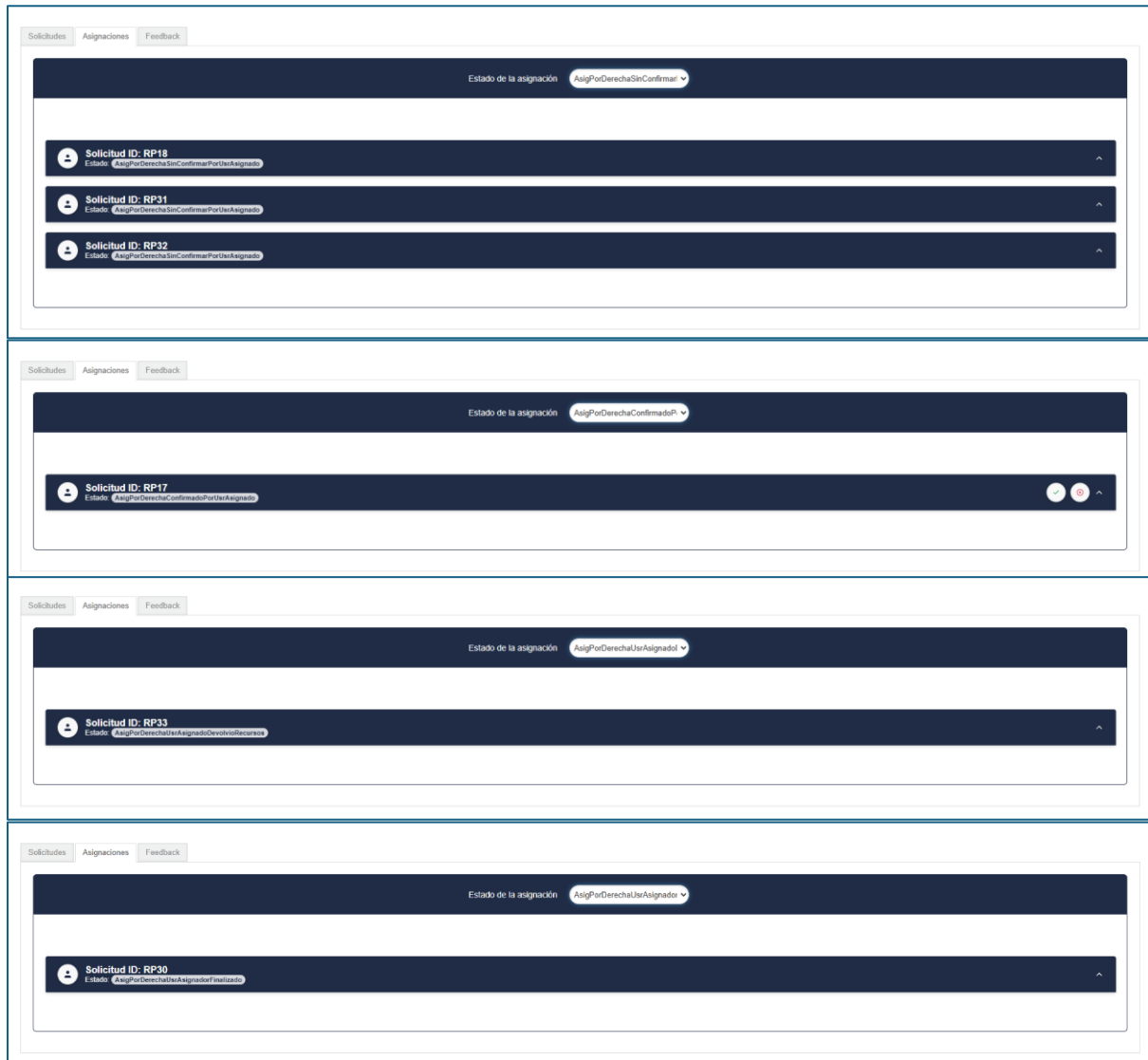
Solicitud ID: RP32 Estado: AsigPorDerechaUsrConfirmadoPorUsrAsignado

Solicitud ID: RP30 Estado: AsigPorDerechaUsrAsignadoFinalizado

Asimismo, se verificó que los filtros por estado funcionaran conforme a lo definido (En Estudio, Aprobada, Rechazada, Cancelada, Finalizada), permitiendo refinar la búsqueda sin pérdida de información como se puede apreciar en la Figura 85 y 102. Cada registro permitió acceder al detalle con trazabilidad completa, asegurando consistencia entre solicitudes, recursos y estados finales.

Figura 102.

Vista de la pestaña Asignaciones con diferentes filtros de estado.



5.1.3.6.13 HU-13 Validación de recursos propuestos por propietarios. En el entorno de pruebas se verificó la funcionalidad de validación de recursos propuestos por propietarios desde el área de administración.

Se confirmó que el listado de recursos en TP_Recurso incluyera el campo de validación como se puede observar en la Figura 103 y que los cambios realizados por el administrador quedarán reflejados en la base de datos como se puede apreciar en la Figura 104. Las actualizaciones persistieron correctamente y fueron visibles en consultas posteriores,

cumpliendo con el objetivo de identificar qué recursos habían sido revisados sin afectar su disponibilidad.

Figura 103.

Vista de la pestaña Validación.

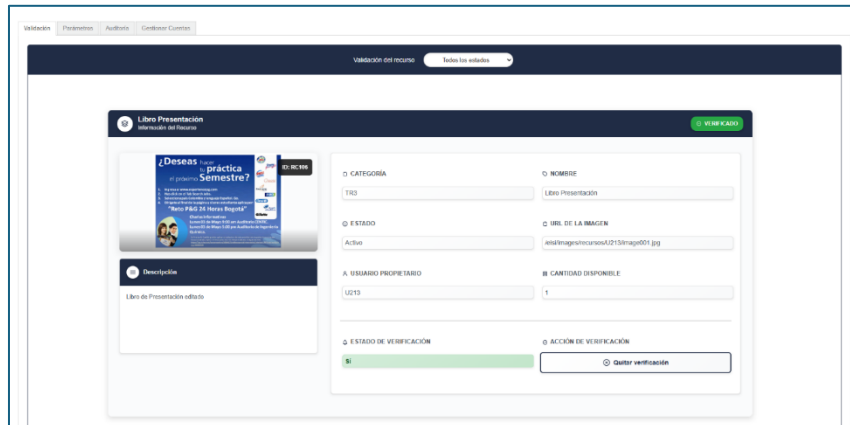
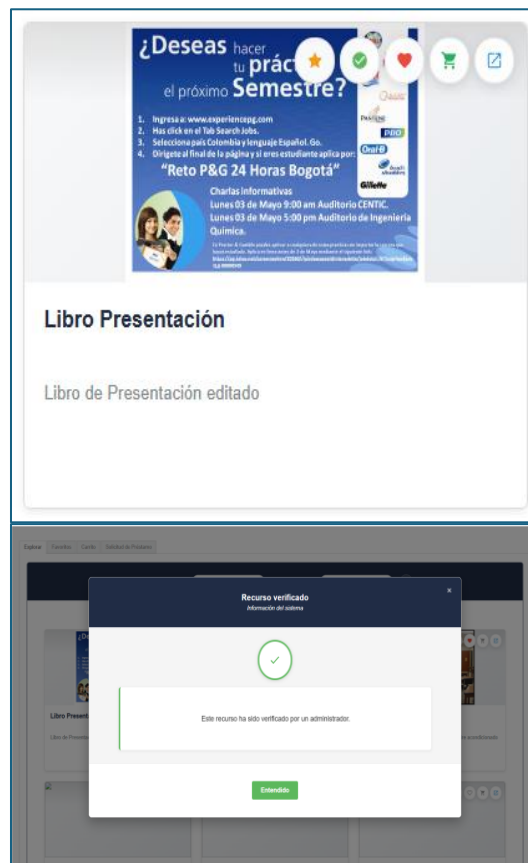


Figura 104.

Vista de la tarjeta del recurso en cuestión y su respectivo icono de validación.



5.1.3.6.14 HU-14 Gestión de categorías de recursos. En el entorno de pruebas se comprobó la correcta gestión de categorías de recursos por parte del administrador.

Se validó que fuera posible crear, editar y desactivar categorías en TP_CategoriaRecurso como se puede observar en la Figura 105 y 106, garantizando la unicidad de los nombres para evitar duplicados. El listado mostró nombre, descripción y estado de cada categoría, confirmándose que las categorías inactivas permanecen registradas en la base de datos y en los historiales, pero dejan de estar disponibles en formularios de registro de nuevos recursos. Así mismo se mostraron las respectivas notificaciones luego de realizar alguna acción

Figura 105.

Vista de la pestaña Parámetros.

The screenshot displays the 'Parámetros' (Parameters) tab in the COMA platform. It is divided into two main sections: 'Agregar Nueva Categoría' (Add New Category) and 'Categorías Registradas' (Registered Categories).

Agregar Nueva Categoría: This section has a dark header with a plus icon and the text 'Agregar Nueva Categoría' and 'Complete los campos para crear una nueva categoría'. Below the header, there are two input fields: 'Nombre' (Name) with the value 'Categoría pianos' and 'Descripción' (Description) with the value 'Categoría pensada para agrupar todos los salones de la escuela.' A green plus icon is visible on the right side of the form.

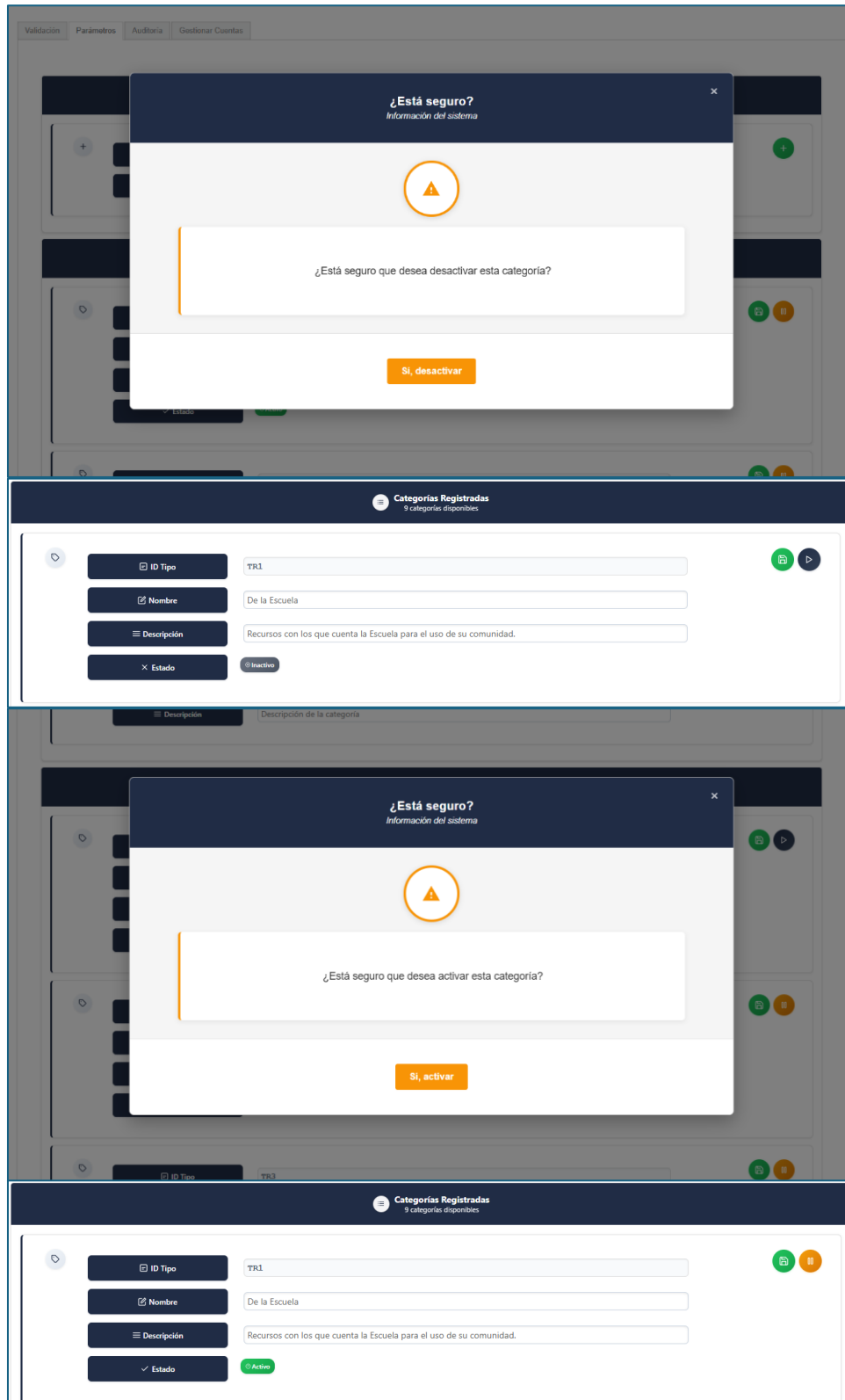
Categorías Registradas: This section has a dark header with a minus icon and the text 'Categorías Registradas' and '10 categorías disponibles'. It contains a list of three categories, each with its own form:

- Category 1:** ID Tipo: TR1, Nombre: De la Escuela, Descripción: Recursos con los que cuenta la Escuela para el uso de su comunidad. Estado: Activo (green checkmark).
- Category 2:** ID Tipo: TR11, Nombre: Categoría salones, Descripción: Categoría pensada para agrupar todos los salones de la escuela. Estado: Activo (green checkmark).
- Category 3:** ID Tipo: TR11, Nombre: Categoría pianos, Descripción: Categoría pensada para agrupar todos los pianos de la escuela. Estado: Activo (green checkmark).

Each category entry includes a trash icon and an edit icon on the right side.

Figura 106.

Vista para activar y desactivar una categoría.



Con estas pruebas se aseguró la persistencia de datos y la consistencia en el uso de categorías, manteniendo un orden claro que facilita tanto la administración como las búsquedas posteriores.

5.1.3.6.15 HU-15 Reportes de auditoría del módulo. En el entorno de pruebas se verificó la disponibilidad de un reporte consolidado por solicitud en el área de administración como se puede apreciar en la Figura 107.

Figura 107.

Vista de la pestaña Auditoría.

The screenshot displays the 'Auditoría' (Audit) tab in the COMA platform. It shows a list of resource loan requests for users UFR31 through UFR35. Each entry includes details such as 'Solicitud' (Request), 'Evento' (Event), 'Asignador' (Assigner), and 'Información del Recurso' (Resource Information).

UFR31

Información de la Solicitud		
Solicitud: Sin información	Evento: Primeros puntos del estado de recursos	Asignador: Sin información
Fecha Solicitud: 2025-05-01 01:00:00	Inicio: 2025-05-01 00:00:00	Final: 2025-05-12 10:00:00
EvalúaSolicitud: Sin información	FechaProcesoSolicitud: Sin información	FechaSolicitudFinalizada: Sin información
ComentaSolicitud: Sin información	ComentaProcesoSolicitud: Sin información	ComentaSolicitudFinalizada: Sin información

Información del Recurso

Recurso	Cantidad
RC100	1

UFR32

Información de la Solicitud		
Solicitud: Sin información	Evento: Primeros puntos del estado de recursos	Asignador: Sin información
Fecha Solicitud: 2025-05-01 01:00:00	Inicio: 2025-05-01 00:00:00	Final: 2025-05-12 10:00:00
EvalúaSolicitud: Sin información	FechaProcesoSolicitud: Sin información	FechaSolicitudFinalizada: Sin información
ComentaSolicitud: Sin información	ComentaProcesoSolicitud: Sin información	ComentaSolicitudFinalizada: Sin información

Información del Recurso

Recurso	Cantidad
RC100	1

UFR33

Información de la Solicitud		
Solicitud: Sin información	Evento: Primeros puntos del estado de recursos	Asignador: Sin información
Fecha Solicitud: 2025-05-01 01:00:00	Inicio: 2025-05-01 00:00:00	Final: 2025-05-12 10:00:00
EvalúaSolicitud: Sin información	FechaProcesoSolicitud: Sin información	FechaSolicitudFinalizada: Sin información
ComentaSolicitud: Sin información	ComentaProcesoSolicitud: Sin información	ComentaSolicitudFinalizada: Sin información

Información del Recurso

Recurso	Cantidad
RC100	1

UFR34

Información de la Solicitud		
Solicitud: Sin información	Evento: Primeros puntos del estado de recursos	Asignador: Sin información
Fecha Solicitud: 2025-05-01 01:00:00	Inicio: 2025-05-01 00:00:00	Final: 2025-05-12 10:00:00
EvalúaSolicitud: Sin información	FechaProcesoSolicitud: Sin información	FechaSolicitudFinalizada: Sin información
ComentaSolicitud: Sin información	ComentaProcesoSolicitud: Sin información	ComentaSolicitudFinalizada: Sin información

Información del Recurso

Recurso	Cantidad
RC100	1

UFR35

Información de la Solicitud		
Solicitud: Sin información	Evento: Primeros puntos del estado de recursos	Asignador: Sin información
Fecha Solicitud: 2025-05-01 01:00:00	Inicio: 2025-05-01 00:00:00	Final: 2025-05-12 10:00:00
EvalúaSolicitud: Sin información	FechaProcesoSolicitud: Sin información	FechaSolicitudFinalizada: Sin información
ComentaSolicitud: Sin información	ComentaProcesoSolicitud: Sin información	ComentaSolicitudFinalizada: Sin información

Información del Recurso

Recurso	Cantidad
RC100	1

Se comprobó que la vista reuniera información desde TB_RecPedido, TR_UsoRecurso, TP_Recurso y TP_Usuario, mostrando de manera coherente fecha y hora de registro, usuario solicitante, estado actual y recursos vinculados, junto con datos asociados como comentarios o calificaciones cuando existían.

Adicionalmente, se validó que los filtros dinámicos funcionaran correctamente, permitiendo al administrador ajustar la consulta según criterios definidos y facilitando el seguimiento administrativo de los casos como se puede apreciar en la Figura 108 y 109.

Figura 108.

Filtros de búsqueda de la pestaña Auditoría.

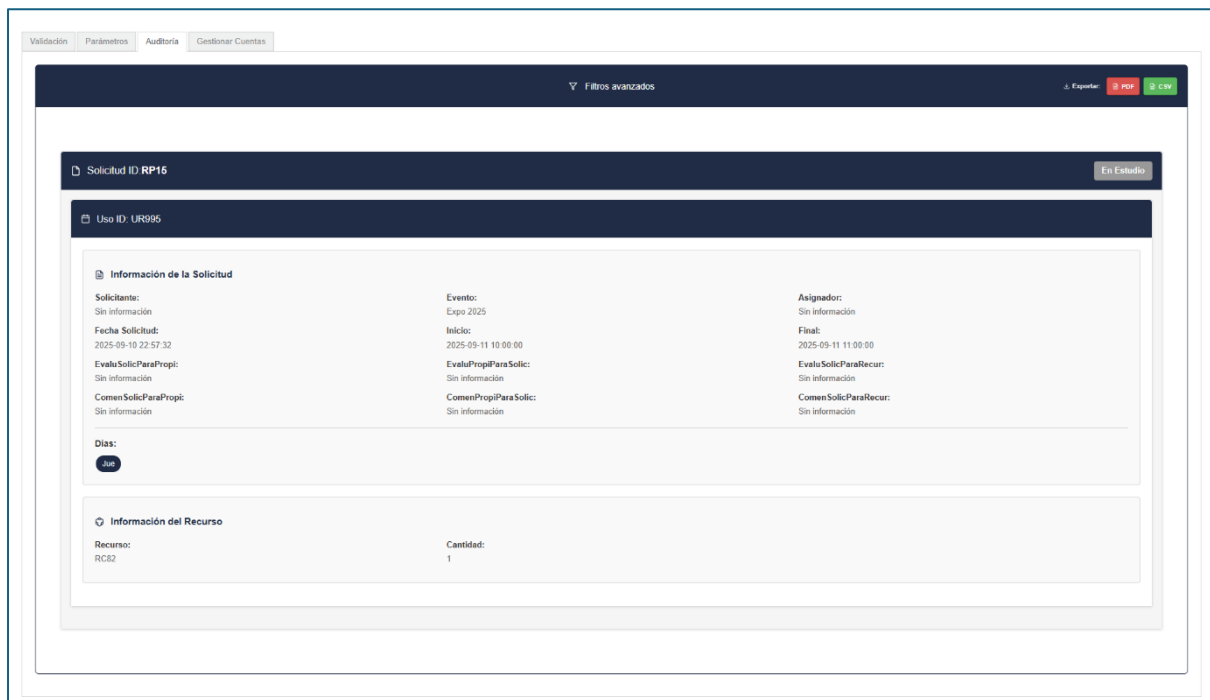
The screenshot shows a web application interface with a modal window titled "Filtros de búsqueda" (Search Filters). The modal is open over a background page that includes a navigation menu with "Validación", "Parámetros", "Auditoría", and "Gestionar Cuentas". The modal contains the following fields:

- Solicitud:** Ingrese ID Solicitud
- Recurso:** Ingrese ID Recurso
- Solicitante:** Ingrese ID Solicitante
- Uso Recurso:** Ingrese ID Uso Recurso
- Evento:** Ingrese Evento
- Estado:** Ingrese Estado
- Asignador:** Ingrese ID Asignador

At the bottom of the modal is a button labeled "Aplicar filtros". The background page shows a table with columns for "Recurso:" and "Cantidad:", with a value of "1" under "Cantidad".

Figura 109.

Vista de las coincidencias encontradas luego de hacer una búsqueda.



5.1.3.7 Implantación y seguimiento. Para la puesta en marcha del módulo se utilizó una herramienta interna desarrollada por el grupo Calumet, denominada “Super Condor”. Esta herramienta permite automatizar el despliegue en el servidor de producción mediante un script basado en Node.js y flujos de Git Actions, lo que facilita la publicación de las distintas ramas del repositorio y el seguimiento del proceso de compilación para detectar posibles fallos.

El paso a producción se realizó de manera gradual, procurando un inicio estable del sistema. Durante los primeros días se mantuvo un periodo de observación que permitió identificar las primeras impresiones de los usuarios y hacer ajustes menores cuando fue necesario, favoreciendo una integración natural en las tareas habituales sin generar interrupciones.

En esta etapa se efectuó un monitoreo cercano del funcionamiento general, revisando aspectos como la navegación, los tiempos de respuesta y la claridad de los mensajes mostrados

al usuario. A partir de esas observaciones se realizaron mejoras puntuales que no afectaron la estabilidad del sistema, pero que aportaron mayor comodidad y fluidez en su uso. Por ejemplo, se ajustaron consultas que habían cambiado debido a la actualización de algunos nombres de clases, se definieron valores por defecto en la pestaña de pre-solicitud para simplificar el registro de recursos y se añadió un botón de confirmación al final del proceso para hacerlo más intuitivo como se puede observar en la Figura 110. También se creó una tabla específica para organizar los códigos de inventario de los recursos institucionales, facilitando su registro y consulta.

Figura 110.

Vista de los nuevos cambios agregados al formulario de pre-solicitud.

The screenshot displays a web interface for a resource request form. At the top, there are navigation tabs: 'Explorar', 'Favoritos', 'Carrito', and 'Solicitud de Préstamo'. The main header shows 'Solicitud ID: RP27' and '1 recurso'. The form is organized into several sections:

- Event Information:** A list of fields on the left with corresponding input boxes on the right:
 - Nombre del evento: Sin nombre de evento
 - Fecha de inicio: mm / dd / aaaa
 - Hora de inicio: -- : --
 - Fecha de finalización: mm / dd / aaaa
 - Hora de finalización: -- : --
- Repetition:** A section titled '¿Cómo funciona la repetición?' with a subtext: 'El botón «Repetir» permite reservar el recurso solo en los días que elijas dentro del rango de fechas. Si no lo activas, se asumirá que necesitas el recurso todos los días y durante todo el día.' Below this is a 'Repetir' button and a checkbox for 'Activar evento recurrente'.
- Resource Selection:** A card titled '¿Deseas hacer la práctica Semestre?' with a list of items. Below the card is a 'Libro Presentación' section with a 'CANTIDAD' input field set to '1'.

At the bottom of the form, there is a prominent green button labeled 'Realizar solicitud'.

Como apoyo al proceso, se elaboró un manual de uso destinado a orientar a los usuarios en las funciones principales, resolver dudas frecuentes y servir como referencia en la operación

cotidiana. Este documento ayuda a mantener uniformidad en el manejo de la herramienta y a facilitar su adopción por parte de la comunidad.

En conjunto, estas acciones permitieron dejar el sistema en condiciones adecuadas de funcionamiento, con una base estable que admite ajustes y mejoras futuras según las necesidades que puedan presentarse.

6. Conclusiones

El desarrollo del módulo de préstamos en la plataforma COMA permitió dar cumplimiento al objetivo general de diseñar, implementar y poner en funcionamiento una herramienta orientada al actor solicitante para mejorar la gestión de recursos de diversa naturaleza. A través de este trabajo se comprobó que la adopción de una metodología ágil, apoyada en historias de usuario y prototipado evolutivo, favorece la retroalimentación temprana, la validación progresiva de requisitos y la incorporación oportuna de mejoras, lo que resultó clave para alcanzar los resultados propuestos.

En términos funcionales, el módulo ofrece al solicitante una experiencia de uso más completa al integrar búsqueda y filtrado de recursos, marcación de favoritos, agrupación de recursos en un carrito, validación automática de disponibilidad, seguimiento de estados, cancelación de solicitudes, notificaciones automáticas y evaluaciones posteriores. Asimismo, se completó el desarrollo de las herramientas para el administrador, incluyendo la validación de recursos y gestión de categorías. Frente al objetivo de configuración, se optó por entregar un módulo de reportes de información que agrupa los datos de recursos, usuarios y retroalimentación. Esta decisión permitió priorizar la accesibilidad y transparencia de los datos generados en la plataforma, estableciendo un repositorio de información consultable que sirve de base para la gestión operativa, dejando la automatización de avisos para una fase posterior.

Desde el punto de vista técnico, la implementación bajo el patrón Modelo–Vista–Controlador con Servlets, DAOs y JSP, junto con MySQL como motor de datos, contribuyó a una estructura organizada que facilita el mantenimiento, la escalabilidad y la integración con la plataforma institucional existente. Esta arquitectura permitió separar responsabilidades, mejorar la seguridad mediante validaciones en servidor y dejar bases sólidas para futuras ampliaciones.

No obstante, el proyecto también evidenció desafíos asociados con la necesidad de una integración armónica con los componentes ya existentes en COMA, lo cual implicó ajustes para asegurar la compatibilidad. De igual forma, se reconoce la importancia de realizar pruebas de adopción en entornos productivos para medir el impacto real en los procesos de préstamo y recoger retroalimentación de usuarios finales que permita continuar perfeccionando la solución.

En conjunto, los resultados alcanzados demuestran que la propuesta no solo mejora los flujos de solicitud y administración de recursos, sino que también sienta las bases para escalar a otros contextos y adaptarse a diferentes tipos de recursos. A partir de este trabajo, se abren oportunidades para incorporar nuevas funcionalidades, fortalecer los mecanismos de seguridad y explorar integraciones con otros módulos de la plataforma, lo que proyecta un horizonte de mejora continua para la gestión de recursos tanto institucionales como no institucionales.

7. Recomendaciones

El módulo de préstamos desarrollado para la plataforma COMA abre la posibilidad de fortalecer la gestión de recursos en la institución y de proyectarse a nuevos escenarios. Con el

fin de aprovechar su potencial, se recomienda promover su adopción gradual a través de campañas de socialización que difundan sus beneficios y motiven a la comunidad a integrarlo en sus actividades diarias. Esta estrategia permitirá consolidar una base de usuarios activa que sirva de punto de partida para futuras mejoras.

Asimismo, resulta pertinente establecer mecanismos permanentes de evaluación que permitan identificar tendencias de uso y anticipar necesidades, de modo que las decisiones de actualización se apoyen en información objetiva. La ampliación de funcionalidades, como la incorporación de reportes analíticos, la integración con otros módulos institucionales o la adaptación a distintos tipos de recursos, puede considerarse como una línea natural de crecimiento.

Finalmente, se sugiere mantener una política de innovación continua que favorezca el desarrollo de nuevas características y potencie la experiencia de los usuarios. Con estas acciones, el módulo no solo consolidará su posición dentro de la plataforma COMA, sino que también podrá escalar a otros contextos y responder a los retos futuros de la gestión de recursos de manera flexible y sostenible.

Referencias bibliográficas

- Beck, K. (2004). *Extreme programming explained: Embrace change* (2nd ed.). Addison-Wesley.
- Beck, K., et al. (2001). *Manifesto for Agile Software Development*.
<http://agilemanifesto.org/>
- Brown, J. S., & Duguid, P. (2000). *The social life of information*. Harvard Business School Press.
- Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.
- Cooper, A., Reimann, R., Cronin, D., & Noessel, C. (2014). *About face: The essentials of interaction design* (4th ed.). Wiley.
- Davenport, T. H., & Kirby, J. (2016). *Only humans need apply: Winners and losers in the age of smart machines*. HarperBusiness.
- Fowler, M. (2004, June 29). *Original Strangler Fig Application*.
<https://martinfowler.com/bliki/StranglerFigApplication.html>
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.
- Hammer, M., & Champy, J. (1993). *Reengineering the corporation: A manifesto for business revolution*. HarperBusiness.
- Highsmith, J. (2009). *Agile project management: Creating innovative products* (2nd ed.). Addison-Wesley.
- ISO (International Organization for Standardization). (2013). *ISO/IEC 27001:2013 — Information technology — Security techniques — Information security management systems — Requirements*.
<https://www.iso.org/standard/54534.html>
- Moore, G. A. (1991). *Crossing the chasm: Marketing and selling high-tech products to mainstream customers*. HarperBusiness.
- Nielsen, J. (1993). *Usability engineering*. Academic Press.

Norman, D. A. (2013). *The design of everyday things* (Revised and expanded ed.). Basic Books.

OWASP Foundation. (2021). *OWASP Top Ten 2021*.
<https://owasp.org/Top10/>

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Rogers, E. M. (2003). *Diffusion of innovations* (5th ed.). Free Press.

Shneiderman, B., Plaisant, C., Cohen, M. S., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human–computer interaction* (6th ed.). Pearson.

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

Wroblewski, L. (2011). *Mobile first*. A Book Apart.