

**Software Para el Control de Alquiler de los Recursos Disponibles del Campus
Universitario de la Universidad de Cundinamarca, Extensión Soacha**

Autor:

Deicy Tatiana Lozano Barón

Director y Coautor:

Ing. Dilia Inés Molina Cubillos

Trabajo Para Obtener El Título de Tecnólogo en Desarrollo de Software

Universidad de Cundinamarca - Facultad de Ingeniería

Programa Tecnología En Desarrollo De Software Soacha (Cundinamarca)

Octubre 2025

Nota De Aceptación

Firma presidente del jurado

Firma del jurado

Firma del jurado

Ciudad _____ y Fecha (____, _____, ____)

Dedicatoria

Este proyecto está dedicado principalmente a Dios, quien me ha dado la vida, la fortaleza y la guía necesarias para llevar a cabo este trabajo. También quiero agradecer a mi familia y amigos, quienes han sido pilares fundamentales en mi vida; sin su apoyo, amor y confianza, jamás hubiera tenido la valentía ni la tenacidad para completar este proyecto. Gracias por su cariño incondicional, sus consejos y enseñanzas, que han contribuido a formarme como mejor persona y profesional.

Agradecimientos

Quiero expresar mi más profundo agradecimiento a Dios, quien me ha brindado la sabiduría, la fuerza y el conocimiento necesarios para llevar a cabo este proyecto. Su guía ha sido fundamental en cada paso de este proceso.

Agradezco de manera especial a mi familia, por su amor incondicional, paciencia y apoyo constante, que han sido un ejemplo de perseverancia y motivación. Sus consejos y enseñanzas han dejado en mí un valioso legado que me impulsa a alcanzar mis metas con dedicación y valentía.

También deseo reconocer el acompañamiento de mis docentes y, en particular, de la ingeniera Dilia Inés Molina Cubillos, directora de este proyecto, por su guía, motivación y apoyo incondicional, que han sido esenciales para el desarrollo y finalización de este trabajo.

Finalmente, agradezco al personal administrativo por la información y la colaboración brindada, sin la cual este proyecto no habría sido posible. Su apoyo ha facilitado enormemente la realización de este proceso y contribuye a mi crecimiento profesional.

Tabla de contenido

Dedicatoria.....	2
Agradecimientos	3
Lista de Figuras	10
Glosario	13
Resumen	17
Abstrac.....	18
Introducción	19
Capítulo 1. Planteamiento Del Problema.....	20
Descripción Del Problema	21
Formulación Del Problema	22
Capítulo 2. Justificación	23
Capítulo 3. Objetivos.....	25
Objetivo General	25
Objetivos Específicos	25
Capítulo 4. Alcance.....	26
Capítulo 5. Diseño Metodológico.....	27
Tipo de Investigación:.....	27
Metodologías:.....	27
Metodología Prototipo:	27
Fase 1 Análisis	28
Fase 2 Diseño (Modelado, Diseño Rápido)	28

Fase 3 Desarrollo (Construcción Del Prototipo).....	29
Fase 4 Prueba De Funcionalidad Y Aprobación	30
Capítulo 6. Estado Del Arte.....	31
Nivel Nacional	32
<i>Diseño Y Desarrollo De La Solución De Software De Gestión De Espacios Físicos En La</i> <i>Universidad Distrital.....</i>	32
<i>Universidad De Bogotá Jorge Tadeo Lozano.</i>	34
<i>Sistema De Préstamo De Espacios Y Equipos De La Facultad De Artes Y Humanidades. ...</i>	35
<i>Proceso Préstamo De Espacios – Pontificia Universidad Javeriana</i>	36
<i>Conclusión.</i>	37
Capítulo 7. Marco Referencial.....	38
Marco Histórico	38
<i>Primeros Métodos De Gestión Manual De Recursos.....</i>	38
<i>Migración Hacia Sistemas En Línea.....</i>	38
<i>Evolución Hacia La Nube Y Aplicaciones Móviles.</i>	38
<i>Evolución de la gestión de espacios en la Universidad de Cundinamarca,.....</i>	39
Marco Teórico	40
<i>Gestión de Recursos en Entornos Universitarios.....</i>	40
<i>Sistemas de Información y Controles de Recursos.....</i>	40
<i>Software de Gestión de Alquiler de Recursos.....</i>	40
<i>Tecnologías de Apoyo en el Desarrollo de Software.</i>	41
<i>Beneficios del Software de Control de Alquiler en Campus Universitarios.....</i>	41

<i>Herramientas CASE.</i>	42
<i>Herramienta CASE Diagrama</i>	44
<i>Herramientas para modelado de proceso.</i>	45
<i>Herramientas de documentación.</i>	45
<i>Herramientas de análisis.</i>	45
<i>Herramientas de diseño.</i>	46
<i>Herramientas de programación.</i>	46
<i>Herramientas de desarrollo Web.</i>	47
<i>Herramientas de Aseguramiento de la calidad.</i>	47
<i>Herramientas de mantenimiento.</i>	48
<i>Metodologías de desarrollo de software.</i>	48
<i>Metodología Tradicional.</i>	49
<i>Metodología Ágil.</i>	50
<i>UML</i>	52
<i>Diagrama de caso de uso.</i>	53
<i>Diagrama de clases.</i>	54
<i>Diagrama de objetos.</i>	54
<i>Diagrama de componentes.</i>	55
<i>Diagrama de despliegue o distribución.</i>	56
<i>Diagrama de secuencias.</i>	56
<i>Diagrama de Actividades.</i>	57

<i>Diagrama de Interacción</i>	58
Metodología de Prototipado.....	59
<i>Etapas Del Modelo De Prototipos</i>	60
<i>Recogida Y Perfeccionamiento De Requisitos</i>	60
<i>Diseño Y Modelado Ágil</i>	61
<i>Desarrollo Del Prototipo</i>	61
<i>Revisión Del Prototipo Por El Cliente</i>	61
<i>Mejora Del Prototipo</i>	61
<i>Prototipo Final</i>	61
Marco Legal.....	63
<i>Ley 30 De 1992</i>	63
<i>Ley 1581 de 2012 – Protección de datos personales</i>	64
<i>Ley 603 de 2000 – Licenciamiento de software</i>	65
Marco Tecnológico.....	66
<i>Bootstrap</i>	66
<i>Java</i>	66
<i>JSTL</i>	67
<i>MVC</i>	67
<i>DAO</i>	67
<i>Oracle</i>	68
<i>NetBeans</i>	68

<i>Apache Tomcat</i>	69
<i>Software Libre</i>	70
Marco Geográfico	72
Capítulo 8. Estructura Temática	73
Fase 1 Análisis (Recolección y Refinamiento De Requisitos)	73
<i>Levantamiento De Información</i>	73
<i>Técnicas De Entrevista</i>	74
<i>Mesas De Trabajo</i>	74
<i>Análisis De Documentación</i>	74
<i>Casos de uso</i>	75
<i>Requerimientos Funcionales</i>	78
<i>Requerimientos No Funcionales</i>	81
<i>Requerimientos No Funcionales De Software Y Hardware</i>	84
Fase 2 Diseño (Modelado, Diseño Rápido)	85
<i>Mockups</i>	85
<i>Diagrama de actividades</i>	95
<i>Diagrama de estados</i>	96
<i>Diagrama de flujo</i>	99
<i>Modelo entidad relación</i>	101
<i>Diseño conceptual base de datos</i>	102
Fase 3 Desarrollo (Construcción del prototipo)	103

Capítulo 9. Resultados.....	113
Capítulo 10. Discusión	116
Capítulo 11. Conclusiones	117
Capítulo 12. Recomendaciones	118
Capítulo 13. Bibliografía.....	119
Capitulo 14. Anexos.....	122

Lista de Figuras

Figura 1	44
Figura 2	52
Figura 3	53
Figura 4	54
Figura 5	55
Figura 6	55
Figura 7	56
Figura 8	57
Figura 9	58
Figura 10	59
Figura 11	62
Figura 12	72
Figura 13	72
Figura 14	87
Figura 15	88
Figura 16	89
Figura 17	90
Figura 18	91
Figura 19	92
Figura 20	92
Figura 21	93
Figura 22	93
Figura 23	94
Figura 24	94
Figura 25	97

Figura 26	98
Figura 27	98
Figura 28	99
Figura 29	100
Figura 30	101
Figura 31	102
Figura 32	104
Figura 33	105
Figura 34	105
Figura 35	106
Figura 36	107
Figura 37	107
Figura 38	108
Figura 39	108
Figura 40	109
Figura 41	110
Figura 42	110
Figura 43	111
Figura 44	111
Figura 45	112
Figura 46	113
Figura 47	114
Figura 48	114
Figura 49	115

Lista de Tablas

Tabla 1	50
Tabla 2	77
Tabla 3	78
Tabla 4	79
Tabla 5	79
Tabla 6	79
<i>Tabla 7</i>	80
Tabla 8	80
Tabla 9	80
Tabla 10	81
Tabla 11	81
Tabla 12	82
Tabla 13	82
Tabla 14	82
Tabla 15	83
Tabla 16	83
Tabla 17	83
Tabla 18	84
Tabla 19	84

Glosario

Sostenibilidad Financiera: Se refiere a la capacidad de una organización o institución para mantener sus operaciones a lo largo del tiempo mediante una correcta administración de los recursos económicos. Implica no solo recaudar fondos suficientes, sino también utilizarlos de manera eficiente para cubrir necesidades actuales, realizar mantenimientos, aprovechar oportunidades de crecimiento y garantizar un rendimiento estable y confiable que asegure la continuidad del proyecto o la institución.

Recaudos: Son los procesos, acciones o mecanismos mediante los cuales una organización logra reunir recursos económicos. Esto puede incluir cobros directos, pagos de servicios, donaciones, o intermediación para otros. La correcta gestión de los recaudos permite que la institución tenga los fondos necesarios para cumplir con sus objetivos y mantener la operación de manera eficiente.

Visibilidad de espacios: Este concepto implica la presentación clara y comprensible de un espacio físico para que los usuarios puedan conocer su disponibilidad y características. Se logra a través de medios visuales como fotografías, videos o representaciones gráficas que faciliten la planificación de uso, la reserva de los espacios y una mejor organización de recursos, evitando conflictos o falta de información.

Gestión: Se entiende como el proceso mediante el cual una persona o grupo de personas planifica, organiza, coordina y supervisa recursos, actividades y procesos para alcanzar objetivos específicos. La gestión puede ser aplicada tanto a nivel personal como empresarial y busca optimizar el uso de los recursos disponibles, asegurar eficiencia en las tareas y garantizar que las estructuras y procesos funcionen correctamente dentro de la organización.

Oracle: Es una empresa reconocida a nivel mundial por sus soluciones de software, especialmente en bases de datos y sistemas de gestión. Entre sus herramientas se incluyen interfaces gráficas de usuario que permiten interactuar con bases de datos de forma intuitiva, reduciendo la complejidad de operaciones y haciendo más eficientes las tareas de administración y análisis de información, tanto para usuarios como para administradores.

NetBeans: Es un entorno de desarrollo integrado (IDE) gratuito y de código abierto que facilita la creación de aplicaciones de software. Permite a los desarrolladores escribir, depurar y ejecutar programas en distintos sistemas operativos como Windows, Mac, Linux y Solaris. NetBeans incluye herramientas que optimizan el flujo de trabajo, permiten detectar errores rápidamente y simplifican la construcción de aplicaciones eficientes y funcionales.

Bootstrap: Es una biblioteca de código abierto ampliamente utilizada en el desarrollo web. Proporciona componentes prediseñados, plantillas y estilos que permiten construir páginas web y aplicaciones de manera más rápida y uniforme. Al usar Bootstrap, los desarrolladores pueden enfocarse en la funcionalidad y experiencia del usuario sin tener que diseñar todos los elementos desde cero, lo que reduce tiempo y esfuerzo.

Metodología Ágil: Se trata de un conjunto de prácticas y principios que guían el desarrollo de software de forma flexible y colaborativa. Permite que los equipos respondan rápidamente a cambios, mejoren continuamente el producto y entreguen versiones funcionales en ciclos cortos. El enfoque ágil prioriza la satisfacción del usuario final y la comunicación constante entre los miembros del equipo, logrando productos más ajustados a las necesidades reales de los usuarios.

Tecnologías Emergentes: Son innovaciones y herramientas tecnológicas recientes que están transformando la manera en que las organizaciones trabajan. Estas tecnologías permiten automatizar procesos, mejorar la eficiencia, crear nuevos productos y adaptarse a los

cambios del mercado. Su adopción facilita la evolución constante de las empresas y asegura que se mantengan competitivas en un entorno que avanza rápidamente.

Software Público: Se refiere a programas de computadora cuyo uso está restringido según las condiciones establecidas por sus creadores. Estas restricciones pueden limitar la modificación del código, la distribución de copias o la ejecución de ciertas funciones. Su objetivo es proteger la propiedad intelectual y asegurar que el software se utilice de manera controlada y responsable.

Diagramas: Son representaciones gráficas que muestran la estructura y funcionamiento de un sistema, permitiendo identificar claramente los elementos que interactúan entre sí. Los diagramas ayudan a los desarrolladores a planificar, organizar y comprender los procesos del sistema, facilitando la toma de decisiones, la detección de errores y la comunicación entre los miembros del equipo.

Herramientas CASE: Son programas diseñados para apoyar y mejorar el desarrollo de software. Facilitan la creación de diagramas, documentación, pruebas y mantenimiento del código, reduciendo el tiempo de desarrollo y los costos asociados. Su uso permite que los equipos de desarrollo trabajen de manera más organizada y eficiente, mejorando la calidad de los productos finales.

Metodología Prototipo: Es una técnica que consiste en crear una versión inicial de un producto para probar su funcionamiento, detectar fallos y mejorar antes de desarrollarlo por completo. Permite validar ideas de forma rápida y eficiente.

Página Web: Una página web es un documento digital compuesto por contenido textual y elementos multimedia, creado mediante lenguajes de marcado y estilos, y accesible a través de un navegador web. Es la forma principal en la que se presenta información en Internet y se

ha convertido en una herramienta fundamental para la comunicación.

Lenguajes de bajo nivel: Los lenguajes de programación de bajo nivel son los encargados de proporcionar la abstracción de la arquitectura de un conjunto de instrucciones que una computadora debe realizar, esto se realiza con el fin de que lleve el control directo del hardware.

Lenguajes de alto nivel: Los lenguajes de programación de alto nivel son los encargados de facilitar el trabajo que realizan los programadores, ya que mediante una interfaz gráfica te permiten crear, modificar e implementar los algoritmos necesarios para que así el programa pueda funcionar de manera correcta.

Resumen

El presente proyecto, titulado “**Software Para El Control De Alquiler De Los Recursos Disponibles Del Campus Universitario De La Universidad De Cundinamarca, Extensión Soacha**”, tiene como objetivo principal optimizar la gestión y control de los recursos físicos del campus mediante el desarrollo de un sistema especializado. Actualmente, la administración de los espacios enfrenta dificultades debido a la falta de un software eficiente; los procesos presenciales dificultan la planificación y generan problemas de organización, mientras que el personal encargado carece de una base de datos confiable que respalde la disponibilidad y asignación de los espacios. Con este proyecto se busca brindar una solución tecnológica que permita a los usuarios internos y externos consultar la disponibilidad de los espacios y realizar solicitudes de préstamo de manera ágil.

Para alcanzar este objetivo, se diseñó, desarrolló e implementó el software, que centraliza la información de los recursos disponibles del campus y facilita la visualización y control de las reservas. Esta herramienta proporciona a los usuarios un acceso sencillo y claro para solicitar espacios, mientras que el personal administrativo puede gestionar y supervisar las asignaciones de manera eficiente. La implementación del software busca mejorar la visibilidad, el uso efectivo y la planificación de los recursos universitarios, contribuyendo a un proceso más organizado y optimizando la experiencia de estudiantes, docentes y personal administrativo, fortaleciendo la gestión integral del campus.

Abstrac

The present project, entitled “**Software for the Management of the Rental of Available Resources at the University Campus of the University of Cundinamarca, Soacha Extension**”, has as its main objective the optimization of the management and control of the campus's physical resources through the development of a specialized system. Currently, the administration of the spaces faces difficulties due to the lack of efficient software; in-person processes hinder planning and generate organizational problems, while the staff in charge lacks a reliable database to support the availability and allocation of spaces. This project aims to provide a technological solution that allows both internal and external users to check space availability and submit rental requests efficiently.

To achieve this objective, the software was designed, developed, and implemented, centralizing information on the campus's available resources and facilitating the visualization and control of reservations. This tool provides users with simple and clear access to request spaces, while administrative staff can manage and supervise allocations efficiently. The implementation of the software seeks to improve visibility, effective use, and planning of university resources, contributing to a more organized process and enhancing the experience of students, faculty, and administrative staff, thereby strengthening the comprehensive management of the campus.

Introducción

El presente proyecto tiene como objetivo el desarrollo de un prototipo de software para la gestión de los espacios físicos de la Universidad de Cundinamarca, extensión Soacha. Este sistema se concibe como una herramienta que facilitará la planificación, control y uso eficiente de las áreas destinadas a actividades académicas y administrativas, permitiendo además que usuarios externos puedan consultar y solicitar préstamos de estos espacios de manera organizada.

El problema actual radica en la falta de un sistema digital que permita conocer la disponibilidad de los espacios de forma clara y confiable. Actualmente, los procesos presenciales dificultan la coordinación y generan retrasos en la asignación de recursos. Además, el personal encargado carece de una base de datos centralizada y actualizada, lo que complica la evaluación de las solicitudes y limita la eficiencia de la gestión.

El desarrollo de este prototipo surge del interés de mejorar la administración de los recursos del campus y optimizar la experiencia tanto de estudiantes, docentes y personal administrativo como de usuarios externos. Para su creación, se consideraron aspectos como la visualización intuitiva de los espacios, la facilidad para realizar solicitudes y la capacidad de mantener un registro actualizado de reservas.

En conclusión, la implementación de este sistema busca ofrecer una solución tecnológica que permita gestionar los espacios de la extensión Soacha de manera más eficiente, aumentando la visibilidad, control y aprovechamiento de los recursos disponibles. De esta forma, la universidad podrá garantizar un uso organizado y rentable de sus espacios físicos, contribuyendo al fortalecimiento de la gestión integral del campus.

Capítulo 1. Planteamiento Del Problema

La extensión Soacha de la Universidad de Cundinamarca enfrenta actualmente un reto importante en la administración de sus espacios físicos. La universidad cuenta con aulas, salas de cómputo, auditorios, canchas deportivas y otras instalaciones que deben ser utilizadas por estudiantes, docentes y personal administrativo. Sin embargo, no existe un sistema que permita visualizar de manera clara y actualizada la disponibilidad de estos espacios, lo que dificulta su uso eficiente y limita las oportunidades de aprovecharlos económicamente mediante préstamos a personas externas.

La ausencia de un software especializado para la gestión de espacios genera dificultades en la planificación y organización de su uso. El personal encargado de procesar las solicitudes debe recurrir a métodos presenciales, lo que implica demoras y errores al no contar con información confiable sobre la disponibilidad real de cada espacio. Esta situación afecta directamente la optimización de los recursos y limita la posibilidad de generar ingresos adicionales mediante su uso controlado.

Descripción Del Problema

Actualmente, la universidad enfrenta un problema relacionado con la gestión ineficiente de sus espacios físicos. Aunque existen diversas instalaciones destinadas a actividades académicas y administrativas, muchas de ellas no se utilizan de forma óptima. Esto ha llevado a considerar la opción de prestar estos espacios a personas externas para obtener beneficios económicos. Sin embargo, la falta de un sistema digital que centralice y muestre la disponibilidad de los espacios impide que esta práctica se realice de manera organizada y eficiente.

El procedimiento actual requiere que los interesados realicen la solicitud de manera presencial, lo que resulta poco práctico y propenso a errores. Además, el personal encargado carece de una base de datos confiable y actualizada que le permita verificar rápidamente la disponibilidad de los espacios. Esto genera retrasos en la asignación, reduce la eficiencia de los procesos y limita la utilización completa de los recursos universitarios.

En resumen, la universidad necesita un sistema que permita gestionar y visualizar los espacios disponibles, facilitando su uso tanto para la comunidad interna como para usuarios externos. Este sistema debe centralizar la información, permitir solicitudes ágiles y asegurar un control efectivo, contribuyendo a un aprovechamiento estratégico y económico de los recursos.

Formulación Del Problema

¿Cómo mejorar el control de alquiler de los recursos disponibles del campus universitario de la Universidad de Cundinamarca, extensión Soacha?

Capítulo 2. Justificación

La Universidad de Cundinamarca, extensión Soacha, enfrenta actualmente desafíos significativos en la gestión de sus espacios físicos debido a la ausencia de un sistema digital que permita su visualización y control eficiente. El procedimiento actual, que depende de solicitudes presenciales y procesos manuales, resulta lento y poco efectivo, generando retrasos que afectan tanto a estudiantes como a docentes y personal administrativo. Además, la falta de información centralizada y actualizada limita la planificación estratégica, la asignación correcta de espacios y la posibilidad de generar ingresos mediante préstamos a usuarios externos.

El desarrollo de un software especializado permitirá centralizar la información de los espacios disponibles, facilitando su consulta y reserva de manera rápida y confiable. Esto contribuirá a optimizar el uso de los recursos físicos del campus, mejorar la organización interna y reducir los tiempos de respuesta en los procesos de solicitud. Al digitalizar estas operaciones, se garantiza mayor transparencia en la gestión y se minimizan errores que se producen actualmente por la dependencia de métodos manuales.

Desde el enfoque técnico y tecnológico, la implementación de este sistema se alinea con los avances en informática aplicada a la gestión institucional. La integración de bases de datos confiables, interfaces intuitivas y herramientas de control digital permitirá no solo agilizar los procesos, sino también generar reportes y estadísticas sobre la ocupación y disponibilidad de los espacios, lo que facilita la toma de decisiones informadas y el seguimiento del uso de los recursos. Además, la plataforma podrá ser escalable y adaptable a futuras necesidades de la universidad, incorporando nuevas funcionalidades según los requerimientos de los usuarios.

En el ámbito social, la herramienta beneficia a toda la comunidad universitaria y a los usuarios externos, garantizando un acceso organizado y equitativo a los espacios del campus. Esto fomenta la inclusión, permite la participación en actividades académicas, culturales y

deportivas, y asegura que todos los interesados puedan aprovechar los recursos disponibles de manera eficiente y sin conflictos. Asimismo, al proporcionar información clara y accesible, se mejora la comunicación entre los diferentes actores involucrados en la gestión de espacios.

En términos económicos, el software contribuirá a la sostenibilidad financiera de la universidad al optimizar la asignación de los espacios y permitir el préstamo controlado a terceros. Esto generará ingresos adicionales que podrán ser reinvertidos en el mantenimiento y mejora de los recursos, fortaleciendo la infraestructura del campus y garantizando un uso rentable y responsable de los activos universitarios.

Finalmente, desde el punto de vista de la gestión, la implementación de esta plataforma refuerza la administración integral de la universidad. Facilita la supervisión y control de los espacios, reduce la carga de trabajo del personal administrativo y asegura que la universidad cuente con herramientas modernas para la planificación y organización de sus recursos. En conjunto, este proyecto representa una solución tecnológica integral que mejora la eficiencia, visibilidad y sostenibilidad de los espacios físicos en la extensión Soacha, generando beneficios a corto y largo plazo para toda la comunidad universitaria.

Capítulo 3. Objetivos

Objetivo General

- Desarrollar un software para la gestión de los espacios físicos de la Universidad de Cundinamarca, extensión Soacha, que permita su reserva por usuarios internos y externos, optimice su uso, facilite el control administrativo y mejore la eficiencia operativa de la institución.

Objetivos Específicos

- Recopilar información relevante mediante el levantamiento de información, con el objetivo de detectar la viabilidad y los procesos relacionados con el software, analizar los datos detalladamente de la información recopilada para determinar el alcance y las necesidades del desarrollo del software, asegurando una comprensión completa de los requisitos.
- Definir claramente los requerimientos funcionales y no funcionales que el software debe cumplir, estableciendo los criterios necesarios para su correcto funcionamiento.
- Generar un prototipo para la plataforma principal de la Universidad de Cundinamarca que permita visualizar de manera transparente los espacios disponibles para el entorno social, facilitando su acceso y utilización.
- Realizar las pruebas de funcionalidad del prototipo con las partes interesadas.

Capítulo 4. Alcance

El proyecto tiene como propósito desarrollar un software que permita gestionar de manera eficiente los espacios físicos de la Universidad de Cundinamarca, extensión Soacha, facilitando su control, visibilidad y reserva tanto para la comunidad interna como para usuarios externos. Este sistema busca centralizar la información sobre la disponibilidad de los espacios y optimizar los procesos de solicitud, reserva y autorización, garantizando un uso organizado y transparente de los recursos del campus.

El software permitirá a los usuarios consultar en tiempo real la disponibilidad de los espacios y seleccionar aquellos que deseen ocupar. Además, gestionará la reserva o préstamo de espacios para personas externas que cumplan con los requisitos establecidos por la universidad, registrando de manera digital las solicitudes para asegurar que los datos sean completos y precisos. Los solicitantes recibirán notificaciones sobre el estado de sus solicitudes mediante correo electrónico u otros medios de comunicación integrados, mientras que el personal administrativo podrá validar las solicitudes y supervisar el control de los espacios previamente seleccionados. Asimismo, los usuarios tendrán la posibilidad de evaluar y calificar su experiencia durante el proceso de reserva o préstamo, permitiendo mejorar continuamente el sistema.

De esta manera, el software no solo servirá como una herramienta de gestión, sino que también fortalecerá la organización interna del campus, mejorará la experiencia de los usuarios y contribuirá a la optimización de los recursos físicos de la universidad.

Capítulo 5. Diseño Metodológico

Tipo de Investigación: Para el desarrollo de este proyecto se utilizó la investigación aplicada, ya que su propósito es ofrecer una solución concreta a un problema real. En este caso, se busca mejorar la gestión de los espacios disponibles en la Universidad de Cundinamarca, extensión Soacha, mediante la creación de un software que permita controlar y administrar los préstamos de estos recursos. La investigación aplicada permite generar un resultado práctico que impacte directamente en la eficiencia del campus y en la experiencia de los usuarios.

Metodologías: Las metodologías en el desarrollo de software son conjuntos de técnicas y procesos que guían al equipo para organizar, planificar y controlar las etapas de creación de un proyecto. Usarlas ayuda a reducir errores, anticipar posibles dificultades y garantizar que el producto final cumpla con los objetivos planteados. Existen diferentes tipos de metodologías, pero las más comunes se dividen en dos grupos: las ágiles, que priorizan la flexibilidad y la entrega continua, y las tradicionales, que siguen un flujo más estructurado y secuencial.

Metodología Prototipo: En este proyecto se eligió la metodología de prototipo por su carácter iterativo y enfocado en la interacción con el usuario, lo que facilita la validación temprana de los requisitos y la adaptación a cambios durante el desarrollo del software. Esta metodología se basa en la creación de representaciones preliminares del sistema (prototipos), que permiten visualizar y evaluar las funcionalidades antes de construir el producto final. Cada prototipo ofrece una versión aproximada del software que puede ser revisada por el equipo y los usuarios, lo que posibilita realizar ajustes, corregir errores y redefinir requisitos de manera efectiva.

La metodología de prototipo fomenta la comunicación constante entre desarrolladores y usuarios, asegurando que las expectativas sean claras y se reduzcan riesgos de malinterpretación. A través de este enfoque, se organizaron y ejecutaron las distintas fases del

desarrollo de manera flexible, garantizando que el resultado final responda de forma precisa a las necesidades de la universidad y permita administrar los espacios del campus de manera más organizada y efectiva.

Fase 1 Análisis

En esta primera fase del proyecto, se llevó a cabo la recopilación y análisis de requisitos, con el propósito de transformar las necesidades iniciales en lineamientos claros y procesables para el desarrollo del sistema. Para ello, se aplicaron diferentes técnicas como entrevistas en profundidad con las partes interesadas, mesas de trabajo colaborativas, lluvias de ideas y revisión de documentos relevantes.

Una vez recopilada la información, se procedió a aclarar y organizar los requisitos funcionales y no funcionales, asegurando una comprensión precisa de las expectativas del sistema. Con base en este análisis, se elaboraron especificaciones detalladas y se establecieron puntos de referencia de evaluación que servirán como guía durante las siguientes fases.

Fase 2 Diseño (Modelado, Diseño Rápido)

En la fase de diseño, se definió la organización y el funcionamiento del software antes de su desarrollo, estableciendo un marco preliminar que sirviera como base para los prototipos iniciales. En esta etapa se describió la arquitectura fundamental del sistema y se identificaron sus principales componentes junto con sus relaciones.

Se elaboraron casos de uso, que permitieron detallar la interacción de los usuarios con el sistema y delimitar el alcance funcional. Asimismo, se construyeron modelos entidad-relación y tablas de datos para estructurar la información de manera lógica y coherente. Para complementar la vista del sistema, se desarrollaron diagramas de clases, de objetos y de

actividades, los cuales facilitaron la comprensión de las dinámicas internas del software.

En paralelo, se diseñaron mockups que mostraron de forma visual la disposición de las interfaces y el flujo de navegación, permitiendo anticipar la experiencia de uso y evaluar la facilidad de interacción. Gracias a estos elementos, se logró definir la estructura general del sistema y sentar las bases técnicas y funcionales que guiarán el desarrollo de los prototipos posteriores, asegurando que el software responda tanto a los requisitos establecidos como a la usabilidad esperada.

Fase 3 Desarrollo (Construcción Del Prototipo)

En la fase de desarrollo, se procedió a programar y materializar el software siguiendo las ideas y prototipos diseñados previamente. Se desarrollaron las pantallas del sistema y se conectaron con la base de datos, asegurando que la información fluyera de manera correcta y coherente entre los distintos módulos. Durante este proceso, se consideraron tanto las funcionalidades principales como los aspectos técnicos necesarios para garantizar la estabilidad y el buen desempeño del sistema.

Para la implementación se emplearon diversas tecnologías. En la parte visual (front-end), se utilizó Bootstrap con el fin de lograr una interfaz clara, responsiva y fácil de usar. En la parte lógica (back-end), se trabajó con Java y JSTL, aplicando patrones de diseño como MVC (Model-View-Controller), DAO (Data Access Object) y VO (Value Object), lo que permitió mantener una estructura organizada, modular y de fácil mantenimiento. La base de datos fue gestionada en Oracle, garantizando seguridad, integridad y eficiencia en el manejo de la información. Además, se utilizaron herramientas de apoyo como NetBeans para el entorno de desarrollo y Apache Tomcat como servidor de aplicaciones.

Gracias al uso de estas tecnologías y patrones de diseño, se logró construir un sistema

estable, confiable y alineado con lo planificado, que contribuye a la administración eficiente de los espacios de la universidad.

Fase 4 Prueba De Funcionalidad Y Aprobación

En esta fase de evaluación de la funcionalidad del prototipo, se trabajó directamente con las partes interesadas, permitiendo que interactuaran con el sistema en un entorno controlado. Esta interacción facilitó la recolección de comentarios detallados sobre el desempeño técnico, la usabilidad y la adecuación del software a los requisitos planteados.

Los aportes recibidos fueron fundamentales para identificar oportunidades de mejora y realizar los ajustes necesarios en el prototipo, asegurando que la solución evolucionara de acuerdo con las expectativas de los usuarios. De esta manera, se garantizó el cumplimiento de los objetivos del proyecto y se generó un valor tangible para las partes interesadas, fortaleciendo la alineación entre el producto desarrollado y las necesidades reales de la universidad.

Capítulo 6. Estado Del Arte

Hoy en día, muchas instituciones públicas o privadas están usando software para organizar y prestar sus espacios. Esto demuestra que una herramienta digital facilita mucho el trabajo, evita errores y ayuda a que todo funcione mejor. Por eso, creemos que es necesario crear un software para la Universidad de Cundinamarca, extensión Soacha, que haga más fácil el control y la reserva de sus espacios.

Hemos visto cómo otras universidades han desarrollado sistemas similares y hemos aprendido de lo que les funcionó y de lo que no. Esa experiencia nos sirve de guía para diseñar una herramienta que realmente cumpla con las necesidades de nuestra universidad.

La idea es que el software permita a los usuarios internos y externos ver qué espacios están disponibles, hacer sus reservas de manera rápida y organizada, y que el personal de la universidad pueda gestionar todo sin complicaciones. Además, buscamos que sea una solución práctica que ayude a planificar mejor los espacios y aprovecharlos al máximo.

En resumen, nuestro proyecto no solo quiere mejorar la organización y el uso de los espacios, sino también hacer que todo el proceso sea más fácil, transparente y accesible para todos los que usan o necesitan estos recursos.

Nivel Nacional

Diseño Y Desarrollo De La Solución De Software De Gestión De Espacios Físicos En La Universidad Distrital.

La Universidad Distrital Francisco José de Caldas ha realizado esfuerzos significativos para mejorar la gestión de sus recursos físicos mediante el uso de tecnología. Un ejemplo destacado es el proyecto denominado **“Solución de software para apoyar el proceso de gestión de espacios físicos de la Universidad Distrital”**, desarrollado por Reyes (2017). Este sistema se diseñó para facilitar la administración de los espacios universitarios, permitiendo una asignación más eficiente de aulas, laboratorios y otros ambientes, de acuerdo con las necesidades académicas y administrativas de la institución.

La relevancia de este proyecto radica en que contribuye a reducir problemas comunes como la sobreocupación de espacios o la subutilización de recursos. Al digitalizar la gestión de espacios, la universidad logra optimizar tanto el tiempo del personal encargado como la disponibilidad de instalaciones para estudiantes y docentes (Reyes, 2017). Esta estrategia también está alineada con una tendencia global en las instituciones de educación superior, donde la incorporación de herramientas tecnológicas busca aumentar la eficiencia administrativa, mejorar la planificación.

Desde una perspectiva práctica, el software permite tener un control más claro sobre la ocupación y uso de los espacios, evitando conflictos de agenda y facilitando la programación de actividades académicas y administrativas. Esto no solo beneficia a quienes gestionan los espacios, sino también a la comunidad universitaria en general, mejorando la experiencia de los usuarios y promoviendo un ambiente más organizado y funcional.

Además, iniciativas como la de la Universidad Distrital demuestran cómo la tecnología

puede integrarse de manera estratégica en la administración de las instituciones educativas, convirtiéndose en una herramienta de apoyo clave para la toma de decisiones y la planificación de recursos. La implementación de este tipo de soluciones refleja un compromiso institucional con la modernización, la eficiencia y la calidad en la gestión de los espacios físicos, aspectos fundamentales para el desarrollo académico y administrativo de la universidad.

En síntesis, el proyecto desarrollado por la Universidad Distrital no solo mejora la asignación de espacios y la eficiencia administrativa, sino que también evidencia cómo las herramientas digitales pueden contribuir a la organización y optimización de los recursos dentro de instituciones de educación superior (Reyes, 2017; González, 2019).

Universidad De Bogotá Jorge Tadeo Lozano.

La Universidad de Bogotá Jorge Tadeo Lozano ha desarrollado e implementado una herramienta tecnológica destinada a la gestión y administración de sus espacios físicos, con un enfoque específico en la optimización de actividades académicas (Universidad de Bogotá Jorge Tadeo Lozano, 2015). Esta herramienta permite a los estudiantes, docentes y personal administrativo reservar aulas, laboratorios y auditorios, asegurando un uso eficiente de los recursos disponibles y reduciendo conflictos en la programación de espacios. Según el informe institucional, la plataforma busca mejorar la organización interna y facilitar la planificación de eventos académicos y administrativos, lo que representa un avance significativo en la gestión educativa moderna.

Desde una perspectiva más general, la implementación de sistemas de gestión de espacios físicos en instituciones de educación superior se ha relacionado con una mayor eficiencia operativa y una mejora en la experiencia de los usuarios. Este tipo de herramientas no solo agiliza los procesos internos, sino que también permite recopilar datos importantes sobre la ocupación y uso de los espacios, información que puede ser utilizada para la toma de decisiones estratégicas y la optimización de recursos.

Además, experiencias en otras universidades muestran que la digitalización de la gestión de instalaciones contribuye a reducir tiempos de espera, minimizar la doble reserva de espacios y aumentar la satisfacción de la comunidad académica, lo cual refuerza la importancia de la adopción de estas tecnologías en el contexto educativo actual.

En conclusión, la implementación de la herramienta por parte de la Universidad Jorge Tadeo Lozano constituye un ejemplo de cómo la tecnología aplicada a la administración de espacios físicos puede mejorar la organización interna y apoyar la eficiencia académica.

Sistema De Préstamo De Espacios Y Equipos De La Facultad De Artes Y Humanidades.

La Facultad de Artes y Humanidades de la Universidad de los Andes ha implementado un sistema integral para la gestión y préstamo de espacios y equipos, dirigido a su comunidad académica. (Universidad de los Andes, s.f.) . Este sistema busca optimizar el uso de los recursos disponibles y garantizar su acceso equitativo para actividades académicas, de investigación y creación artística.

El sistema de préstamo de espacios y equipos está disponible para estudiantes, profesores y personal administrativo de la comunidad Facartes. (Universidad de los Andes, s.f.) . Todos los miembros pueden solicitar el acceso y uso de los recursos disponibles, independientemente del programa o dependencia al que pertenezcan.

Para la reserva de espacios, la Facultad utiliza la herramienta Bookeau, que permite a los usuarios consultar la disponibilidad de los espacios y realizar las reservas correspondientes.

En cuanto al préstamo de equipos, este está sujeto a normas específicas que incluyen la obligación de presentar el carné actualizado, la responsabilidad por el buen uso y la devolución en condiciones óptimas, y la posibilidad de sanciones en caso de incumplimiento.

Estas iniciativas reflejan un compromiso institucional con la eficiencia en la gestión de recursos y el apoyo a las actividades académicas y creativas de la comunidad universitaria.

Proceso Préstamo De Espacios – Pontificia Universidad Javeriana .

La Pontificia Universidad Javeriana (PUJ) ha implementado diversos sistemas y estrategias para la gestión eficiente de sus espacios, tanto físicos como virtuales, con el objetivo de optimizar el uso de recursos y facilitar el acceso a la comunidad universitaria. 1. Sistema de Reservas de Espacios La PUJ cuenta con plataformas digitales que permiten a estudiantes, docentes y personal administrativo solicitar y gestionar la reserva de espacios para actividades académicas, culturales y administrativas. Estas plataformas incluyen: Sistema de Reservas. Gestionado por el Centro Ático, permite a los usuarios reservar espacios y equipos, previa capacitación técnica. Portal Universitario +1 Biblioreservas: Plataforma destinada a la reserva de espacios en las bibliotecas de la universidad, facilitando el acceso a salas de estudio y recursos bibliográficos. .(Pontificia Universidad Javeriana, s.f.-a, 2025).

Portal Universitario Infraestructura y Adaptación de Espacios La PUJ ha llevado a cabo proyectos de adaptación de espacios para mejorar su funcionalidad y adecuarlos a las necesidades académicas y tecnológicas actuales. Un ejemplo destacado es: Adaptación del Laboratorio de Electrónica: Este proyecto abordó los retos y procesos de adaptación necesarios para implementar un nuevo plan de estudios con un alto componente práctico. Pontificia Universidad Javeriana. La gestión de Espacios en el Contexto Urbano La universidad también ha explorado la apropiación del espacio urbano a través de iniciativas artísticas, como: Talleres artísticos en el barrio Brisas del Volador: Estos talleres buscan fomentar la participación y la subjetividad política de niños y niñas en el contexto urbano, promoviendo la apropiación del espacio público mediante la acción colectiva.

Conclusión.

Analizados los proyectos de otras universidades podemos concluir que, aunque todos tienen como punto en común el uso de herramientas digitales para organizar y gestionar sus espacios, cada uno lo hace desde realidades y necesidades muy específicas. Algunos priorizan la eficiencia administrativa, otros incluyen además el préstamo de equipos, y otros se enfocan en reservas para actividades académicas o culturales. Esto demuestra que no existe una única solución válida, sino que cada institución adapta la tecnología a su propio contexto.

En nuestro caso, la Universidad de Cundinamarca, extensión Soacha, tiene sus propios retos y particularidades. Por eso, más que copiar lo que ya existe, lo que buscamos es aprender de esas experiencias y diseñar un software que realmente responda a lo que nuestra comunidad necesita. La idea es que sea una herramienta cercana, fácil de usar y que, al mismo tiempo, permita aprovechar al máximo los espacios disponibles, evitando conflictos y mejorando la organización interna.

Lo que nos enseñan los proyectos revisados es que la tecnología es una aliada clave para optimizar recursos y brindar un mejor servicio a estudiantes, docentes y personal administrativo. Nuestro proyecto toma ese aprendizaje como base, pero lo adapta a la realidad de nuestra universidad, con el propósito de construir una solución que no solo modernice la gestión de espacios, sino que también haga más sencilla y transparente la experiencia de quienes los utilizan.

Capítulo 7. Marco Referencial

Marco Histórico

Primeros Métodos De Gestión Manual De Recursos.

En sus inicios, la administración de recursos en las universidades era completamente manual. Para llevar un control, se utilizaban libros de actas, formularios impresos y registros físicos que cumplían la función básica de organizar las reservas. Sin embargo, estos métodos eran poco eficientes y presentaban limitaciones importantes, como la duplicidad de solicitudes, la pérdida de información y la falta de transparencia en los procesos. Tal como señala Montaña (2018, p. 45), este tipo de gestión manual respondía a las necesidades del momento, pero resultaba insostenible con el crecimiento de la demanda académica y administrativa.

Migración Hacia Sistemas En Línea.

Durante los años 1990 y principios del 2000, el internet abrió nuevas posibilidades para las instituciones de educación superior. Como explican Turban et al. (2018, p. 76), los sistemas de gestión en línea introdujeron mejoras sustanciales como el acceso remoto, la verificación de disponibilidad en tiempo real y la generación de reportes estadísticos. Este cambio transformó la manera en que las universidades administraban sus recursos, ya que permitía una mejor planificación y una mayor eficiencia en el uso de los espacios y equipos.

Evolución Hacia La Nube Y Aplicaciones Móviles.

En la actualidad, la gestión de recursos universitarios se apoya en tecnologías en la nube y aplicaciones móviles. Según Pressman y Maxim (2020, p. 320), este tipo de software no solo automatiza las reservas en tiempo real, sino que también incluye funcionalidades avanzadas como análisis de datos, notificaciones automáticas y autenticación de usuarios. En la misma línea, Laudon y Laudon (2016, p. 59) concluyen que los sistemas modernos

favorecen la transparencia, mejoran la experiencia de los usuarios y responden a las demandas de digitalización que caracterizan a la educación superior en el siglo XXI.

Evolución de la gestión de espacios en la Universidad de Cundinamarca, extensión Soacha.

La Universidad de Cundinamarca, extensión Soacha, refleja en la actualidad un punto clave dentro de la evolución histórica de la gestión de recursos en entornos universitarios. Durante años, el control de los espacios físicos se ha llevado a cabo mediante procesos manuales y solicitudes presenciales, un sistema que, aunque funcional en etapas iniciales, hoy resulta limitado y genera retrasos que afectan a estudiantes, docentes y personal administrativo. Esta situación evidencia una problemática similar a la vivida por muchas universidades en décadas pasadas, antes de la incorporación de sistemas digitales que optimizaron la administración de recursos. En este sentido, la falta de centralización y actualización de la información limita la planificación estratégica y restringe oportunidades de aprovechamiento económico, tal como la posibilidad de alquilar espacios a usuarios externos.

La revisión del recorrido histórico de la gestión de recursos en instituciones de educación superior muestra un proceso de transformación constante, impulsado por las necesidades de eficiencia, transparencia y adaptación tecnológica. En este contexto, la Universidad de Cundinamarca, extensión Soacha, se encuentra en un punto de inflexión: sus métodos manuales aún cumplen una función básica, pero resultan insuficientes frente a los retos actuales de digitalización y optimización del uso de espacios. De esta manera, la adopción de plataformas en la nube y aplicaciones móviles no solo representaría una mejora operativa, sino también una oportunidad estratégica para fortalecer la planificación institucional, responder con mayor agilidad a la comunidad académica y explorar nuevas fuentes de sostenibilidad mediante la gestión inteligente de sus recursos.

Marco Teórico

Gestión de Recursos en Entornos Universitarios.

Las instituciones de educación superior disponen de una gran variedad de recursos físicos y tecnológicos que son esenciales para el desarrollo de actividades académicas, investigativas, deportivas y culturales. Estos recursos comprenden aulas, laboratorios, auditorios, bibliotecas, equipos audiovisuales y espacios deportivos. La adecuada administración de dichos espacios es fundamental, pues de ello depende la optimización en su uso y la reducción de conflictos en la programación de actividades (Montaño, 2018). La gestión de recursos puede entenderse como un conjunto de procesos y estrategias orientadas a planificar, organizar y supervisar el uso de los bienes institucionales, con el objetivo de asegurar su disponibilidad, transparencia y sostenibilidad (Chiavenato, 2017).

Sistemas de Información y Controles de Recursos.

Los sistemas de información han transformado la manera en que las organizaciones gestionan sus procesos internos. De acuerdo con (Laudon & Laudon, 2016), un sistema de información es un conjunto de componentes interrelacionados que recolecta, procesa, almacena y distribuye información para apoyar la toma de decisiones, la coordinación y el control. En el ámbito universitario, un sistema de información enfocado en el control de recursos permite centralizar las solicitudes de uso, validar la disponibilidad en tiempo real y generar reportes estadísticos que orienten la planeación institucional (Turban et al., 2018).

Software de Gestión de Alquiler de Recursos.

El software especializado en la gestión de alquiler de recursos está diseñado para automatizar los procesos de reserva, asignación y control de espacios o equipos. Estos sistemas suelen contar con módulos de autenticación, calendarios dinámicos, reportes de uso y, en algunos casos, herramientas de facturación cuando el alquiler implica un costo (Pressman

& Maxim, 2020). Según (Sommerville, 2011), el software de gestión debe ser accesible, escalable y seguro, de manera que pueda adaptarse a las necesidades de la institución y garantizar un funcionamiento confiable en el tiempo.

Tecnologías de Apoyo en el Desarrollo de Software.

El desarrollo de software de control de recursos requiere herramientas tecnológicas que aseguren eficiencia y escalabilidad. Las bases de datos relacionales como MySQL, PostgreSQL o SQL Server permiten almacenar y gestionar información de manera confiable (Elmasri & Navathe, 2017). Asimismo, los lenguajes de programación como Java, Python, PHP y JavaScript, en conjunto con frameworks como Django, Laravel, Spring y React, proporcionan la estructura necesaria para el desarrollo de aplicaciones modernas (Sharma & Kumar, 2020). Por otra parte, las tecnologías en la nube se han convertido en un recurso indispensable, ya que garantizan disponibilidad continua, accesibilidad desde cualquier dispositivo conectado a internet y respaldo seguro de los datos (Mell & Grance, 2011).

Beneficios del Software de Control de Alquiler en Campus Universitarios.

La implementación de software de control de recursos en universidades representa una mejora significativa en la eficiencia administrativa. Estos sistemas permiten reducir tiempos en los procesos de reserva, brindan transparencia y evitan conflictos en la asignación de espacios

(Rodríguez, 2019). Además, el acceso remoto a la información facilita la gestión de solicitudes por parte de estudiantes, docentes y personal administrativo, fomentando un uso más eficiente de la infraestructura. Finalmente, la recopilación y análisis de datos apoyan la toma de decisiones estratégicas, pues permiten identificar patrones de uso y proyectar futuras necesidades en infraestructura y equipamiento (Laudon & Laudon, 2016).

Herramientas CASE.

En el campo de la ingeniería de software, las herramientas CASE (Computer-Aided Software Engineering) se han consolidado como recursos indispensables para apoyar el análisis, diseño y mantenimiento de sistemas. Estas aplicaciones no solo automatizan tareas repetitivas, sino que también permiten mantener la coherencia de los modelos, mejorar la documentación y aumentar la productividad de los equipos de desarrollo. En otras palabras, constituyen un puente entre la teoría de la ingeniería de software y la práctica cotidiana de los proyectos tecnológicos. Según Pressman (2010, p. 143), la incorporación de estas herramientas contribuye directamente a la reducción de errores y a la estandarización de procesos, aspectos clave en proyectos de gran alcance como los que se implementan en instituciones educativas.

De acuerdo con Sommerville (2005, p. 256), el valor agregado de las herramientas CASE radica en su capacidad para integrarse a lo largo del ciclo de vida del software. No se limitan únicamente a las fases iniciales de modelado de requerimientos, sino que también tienen aplicaciones en la generación de código y en la ejecución de pruebas. En la práctica, esto significa que un equipo de trabajo puede mantener una línea de desarrollo mucho más organizada, reduciendo la duplicidad de esfuerzos y facilitando la comunicación entre los distintos actores del proyecto.

En palabras de Piattini y Calero (2002, p. 89), estas herramientas suelen clasificarse en tres tipos: **Upper CASE**, enfocadas en etapas tempranas como el análisis y el diseño; **Lower CASE**, orientadas al desarrollo de código, pruebas y mantenimiento; e **Integrated CASE**, que abarcan todas las fases del proceso de desarrollo. Esta clasificación resulta muy útil porque permite seleccionar la herramienta adecuada según las características del proyecto. En el caso de un software de control de recursos universitarios, por ejemplo, una herramienta *Upper CASE*

ayudaría a modelar diagramas de casos de uso y procesos de reserva de espacios, mientras que una *Lower CASE* apoyaría la implementación del código que gestionará las solicitudes en tiempo real.

Como señala Pressman (2010, p. 145), más allá de la clasificación técnica, el impacto real de las herramientas CASE se percibe en la calidad del software final y en los beneficios organizacionales que generan. Al adoptarlas, no solo se reducen los tiempos y costos de desarrollo, sino que también se asegura que la solución tecnológica responda a las necesidades reales de la institución. En el contexto de la Universidad de Cundinamarca, este tipo de herramientas se convierte en un aliado estratégico para garantizar que el software de gestión de alquiler de recursos cumpla con los estándares de confiabilidad, escalabilidad y seguridad que la comunidad universitaria requiere.

Figura 1

Componentes herramientas CASE.



Nota: La figura muestra los principales componentes que integran las herramientas CASE. Fuente: *Case Tools Overview* [Página web], por Tutorialspoint, s. f., en (https://www.tutorialspoint.com/es/software_engineering/case_tools_overview.htm)

Herramienta CASE Diagrama

Las herramientas CASE basadas en **diagramas** permiten representar de forma visual los procesos y estructuras de un sistema, facilitando la comprensión entre desarrolladores y usuarios. Según Piattini y Calero, en su libro *Análisis y diseño orientado a objetos con UML* (2002, p. 101), los diagramas son esenciales en el análisis y diseño, pues ayudan a modelar casos de uso, clases y procesos de interacción. En la misma línea, Sommerville, en *Ingeniería del software* (2005, p. 265), explica que funcionan como “mapas” del sistema, reduciendo ambigüedades y mejorando la comunicación en los equipos de trabajo. Además, Pressman, en su obra *Ingeniería del software: Un enfoque práctico* (2010, p. 152), señala que su utilización incrementa la calidad del software al detectar errores en etapas tempranas. En el caso de un

sistema de gestión de recursos universitarios, diagramas como los de casos de uso o de secuencia permiten visualizar de manera clara procesos como la reserva y control de espacios.

Herramientas para modelado de proceso.

El modelado de procesos es fundamental para representar y analizar los flujos de trabajo de un sistema. Según Pressman, en *Ingeniería del software: Un enfoque práctico* (2010, p. 198), estas herramientas permiten visualizar y optimizar procesos complejos. Sommerville, en *Ingeniería del software* (2005, p. 274), señala que su valor radica en reducir ambigüedades y mejorar la comunicación entre los actores del proyecto. Por su parte, Piattini y Calero, en *Análisis y diseño orientado a objetos con UML* (2002, p. 115), destacan que diagramas como los de flujo de datos o modelos BPMN facilitan la documentación y mejora continua. En el contexto universitario, estas herramientas resultan clave para representar actividades como reservas o aprobaciones de espacios de forma clara y ordenada.

Herramientas de documentación.

Las herramientas de documentación permiten registrar de manera clara las etapas del desarrollo de software, facilitando la comunicación y la trazabilidad del proyecto. Según Pressman, en *Ingeniería del software: Un enfoque práctico* (2010, p. 210), la documentación asegura la calidad del sistema al respaldar cada fase del ciclo de vida. Sommerville, en *Ingeniería del software* (2005, p. 298), añade que también simplifica el mantenimiento y la evolución futura del software. Por su parte, Piattini y Calero, en *Análisis y diseño orientado a objetos con UML* (2002, p. 130), resaltan que los diagramas y descripciones normalizadas evitan ambigüedades y mejoran la comprensión del proyecto

Herramientas de análisis.

Las herramientas de análisis permiten identificar, especificar y validar los requerimientos de un sistema antes de su diseño y construcción. Según Pressman, en

Ingeniería del software: Un enfoque práctico (2010, p. 156), estas herramientas apoyan la comprensión de las necesidades del usuario y reducen errores al inicio del ciclo de vida del software. Por su parte, Sommerville, en *Ingeniería del software* (2005, p. 214), explica que su principal ventaja es ofrecer modelos estructurados que mejoran la comunicación entre usuarios y desarrolladores. Finalmente, Piattini y Calero, en *Análisis y diseño orientado a objetos con UML* (2002, p. 95), destacan que los diagramas de casos de uso y de clases son esenciales para representar el comportamiento y la estructura del sistema en la fase de análisis.

Herramientas de diseño.

Las herramientas de diseño facilitan la creación de modelos que representan cómo funcionará el sistema a nivel estructural y operativo. Según Pressman, en *Ingeniería del software: Un enfoque práctico* (2010, p. 175), estas herramientas permiten transformar los requerimientos en arquitecturas claras y detalladas. Sommerville, en *Ingeniería del software* (2005, p. 230), explica que su importancia radica en asegurar que el software sea comprensible, modular y fácil de mantener. Por su parte, Piattini y Calero, en *Análisis y diseño orientado a objetos con UML* (2002, p. 112), destacan el uso de diagramas de clases, secuencia y colaboración como elementos clave para representar las interacciones y estructuras del sistema.

Herramientas de programación.

Las herramientas de programación son fundamentales para implementar los diseños y convertirlos en código ejecutable. Según Pressman, en *Ingeniería del software: Un enfoque práctico* (2010, p. 190), los entornos de desarrollo integrados (IDE) y compiladores facilitan la escritura, depuración y mantenimiento del software. Sommerville, en *Ingeniería del software* (2005, p. 245), señala que estas herramientas incrementan la productividad de los desarrolladores al automatizar tareas repetitivas y mejorar la calidad del código. Por su parte,

Piattini y Calero, en *Análisis y diseño orientado a objetos con UML* (2002, p. 140), resaltan que los lenguajes orientados a objetos, junto con sus entornos de desarrollo, permiten implementar de manera más clara los modelos realizados en la fase de diseño.

Herramientas de desarrollo Web.

Las herramientas de desarrollo web permiten crear aplicaciones accesibles desde navegadores, integrando lenguajes, frameworks y entornos especializados. Según Pressman, en *Ingeniería del software: Un enfoque práctico* (2010, p. 205), estas herramientas agilizan la construcción de sistemas interactivos y escalables mediante entornos integrados y bibliotecas de componentes. Sommerville, en *Ingeniería del software* (2005, p. 260), señala que su importancia radica en la capacidad de soportar arquitecturas cliente-servidor y servicios distribuidos, lo que favorece la accesibilidad y la interoperabilidad. Por su parte, Piattini y Calero, en *Análisis y diseño orientado a objetos con UML* (2002, p. 150), destacan que los modelos orientados a objetos sirven de base para implementar aplicaciones web organizadas y coherentes con el diseño planteado.

Herramientas de Aseguramiento de la calidad.

El aseguramiento de la calidad en software busca garantizar que los productos cumplan con los estándares establecidos y satisfagan las necesidades del usuario. Según Pressman, en *Ingeniería del software: Un enfoque práctico* (2010, p. 230), las herramientas de aseguramiento de la calidad permiten planificar, medir y controlar procesos, reduciendo defectos y asegurando la confiabilidad del sistema. Sommerville, en *Ingeniería del software* (2005, p. 310), explica que estas herramientas apoyan actividades como revisiones técnicas, pruebas automatizadas y métricas de calidad que fortalecen la gestión del proyecto. Por su parte, Piattini y Calero, en *Análisis y diseño orientado a objetos con UML* (2002, p. 165), destacan que la calidad también depende de la correcta documentación y del cumplimiento de

estándares en cada fase del desarrollo, lo que puede verificarse con metodologías y herramientas de control.

En el caso de un software para la gestión de recursos universitarios, estas herramientas aseguran que el sistema sea confiable, accesible y libre de errores que afecten la experiencia de los estudiantes, docentes y personal administrativo.

Herramientas de mantenimiento.

El mantenimiento de software es una de las fases más críticas del ciclo de vida, pues asegura que el sistema continúe siendo útil y eficiente después de su implementación. Según Pressman, en *Ingeniería del software: Un enfoque práctico* (2010, p. 250), las herramientas de mantenimiento facilitan la corrección de errores, la adaptación a nuevos entornos y la mejora continua del sistema. Por su parte, Sommerville, en *Ingeniería del software* (2005, p. 330), afirma que este tipo de herramientas apoyan la gestión de versiones, la documentación de cambios y la reutilización de componentes, lo que incrementa la confiabilidad del software a largo plazo.

En un software de gestión de recursos universitarios, estas herramientas resultan esenciales para actualizar funcionalidades, atender nuevas necesidades académicas y garantizar un rendimiento adecuado del sistema en el tiempo.

Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son esenciales para guiar de manera organizada y eficiente los proyectos tecnológicos, pues permiten planificar, controlar y asegurar la calidad del producto final. Según Abuchar Porras, en su libro *Metodologías ágiles para el desarrollo de software* (2023, p. 45), los enfoques modernos como Scrum y Kanban se centran en la adaptabilidad y la mejora continua, ofreciendo un marco flexible que se ajusta a los

cambios frecuentes de los requerimientos.

Por otro lado, Ruiz-Vanoye et al., en *Metodologías de desarrollo de software* (2017, p. 62), explican que los modelos tradicionales como cascada y espiral proporcionan una visión más estructurada del ciclo de vida, siendo útiles en proyectos donde los requisitos están claramente definidos desde el inicio. Estos autores también destacan la relevancia de metodologías mixtas, que combinan prácticas tradicionales con técnicas ágiles para optimizar resultados.

En el contexto universitario, la aplicación de estas metodologías contribuye a desarrollar software más confiable y alineado con las necesidades institucionales, permitiendo que herramientas como los sistemas de gestión de recursos se construyan de forma organizada y sostenible.

Metodología Tradicional.

Las metodologías tradicionales corresponden a una forma despectiva como metodologías pesadas, ya que estas centran su atención en llevar una planificación exhaustiva de todo el proyecto, de acuerdo con esto se pueden establecer las especificaciones precisas de los requerimientos y el modelado para poder cumplir con el plan de trabajo establecido.

Estas metodologías imponen una disciplina rigurosa sobre cada proceso esto con el fin de conseguir un sistema más eficiente, ya que esto depende de la planificación del trabajo que se va a realizar, donde estas se centran principalmente en realizar un análisis detallado de acuerdo con el ciclo del software, esta metodología muchas veces no se adaptan a los cambios, ya que los métodos que se manejan no son adecuados cuando se requiere trabajar en un entorno específico, ya que los requisitos pueden variar en algunas ocasiones.

Metodología Ágil.

Las metodologías ágiles nacen como un modelo de respuesta a los problemas que se pueden ocasionar por las metodologías tradicionales basándose en dos aspectos que son, retrasar las decisiones y la planificación adaptativa, estableciendo un fundamento de adaptabilidad en el desarrollo de los procesos. Un modelo de desarrollo ágil consiste en realizar entregas frecuentes con ciclos rápidos, esto con el fin de que los desarrolladores trabajen constantemente en conjunto con el cliente permitiendo así una comunicación efectiva y constante. Las metodologías ágiles cuentan con una serie de pautas y principios relacionada con una serie de técnicas pragmáticas que hacen que los proyectos sean menos complicados a la hora de fijar el desarrollo y que sea satisfactorio para los clientes, estas metodologías se usan con el fin de generar relevancia al proyecto de tal manera que la capacidad de respuesta sea más rápida y específica.

En conclusión, con lo anterior podemos deducir que la diferencia entre las metodologías tradicionales y las metodologías ágiles es que en el uso de las metodologías tradicionales los principales problemas que se presentan es que muchas veces no se logra planificar de manera concreta los procedimientos a seguir en esta metodologías, pero de acuerdo con esto si de algún manera se logra definir las métricas que logren apoyar las actividades que se desarrollan dentro de los parámetros establecidos. Para esto es importante tener en cuenta que el uso de las metodologías ágiles no siempre valen para la realización de cualquier proyecto, debido a esto una de las principales ventajas que ofrecen las metodologías ágiles es que se pueden acoplar los diferentes procesos de acuerdo al nivel de las practicas que se van a desarrollar durante las diferentes fases permitiendo que tanto las personas que se encuentran a cargo de llevar el desarrollo del software como los clientes estén atentos a cualquier cambio que se considere necesario realizar.

Tabla 1

Comparación de metodologías

METODOLOGIAS TRADICIONALES	ETAPA	METODOLOGIAS AGILES
Planificación predictiva y aislada	Análisis y requerimientos	Planificación adaptativa, entregas frecuentes y colaboración con el cliente.
Diseño flexible, modelos, documentación exhaustiva	Diseño	Diseño simple, documentación mínima, focalizada en la comunicación.
Desarrollo individual con roles y responsabilidades estrictas	Codificación	Transferencia de conocimiento, programación en pares y conocimiento colectivo
Actividades de control orientado a los hitos y gestión de mini proyectos.	Pruebas	Liderazgo, colaboración, empoderamiento y autoorganización.

Nota. Cuadro comparativo de las etapas metodológicas Tradicionales y Ágiles. Fuente: Elaboración propia (2025).

Hoy en día, crear software es un trabajo que requiere mucha dedicación y organización, ya que la competencia en este campo es cada vez mayor. Por eso, existen diferentes formas de trabajar que ayudan a que los proyectos se realicen de manera ordenada. Las metodologías tradicionales, por ejemplo, marcan paso a paso lo que se debe hacer, los documentos que se deben preparar, las herramientas que se deben usar y se enfocan mucho en controlar que todo el proceso se cumpla correctamente.

Por otro lado, las metodologías ágiles se enfocan más en el proyecto en sí y en las personas que lo llevan a cabo. Esto permite trabajar de manera más cercana con el cliente y hacer mejoras continuas en el proyecto mediante pequeñas etapas o pasos cortos. Gracias a esto, resultan muy efectivas cuando los requerimientos cambian con frecuencia y se necesita terminar más rápido, sin dejar de cuidar la calidad del software.

UML

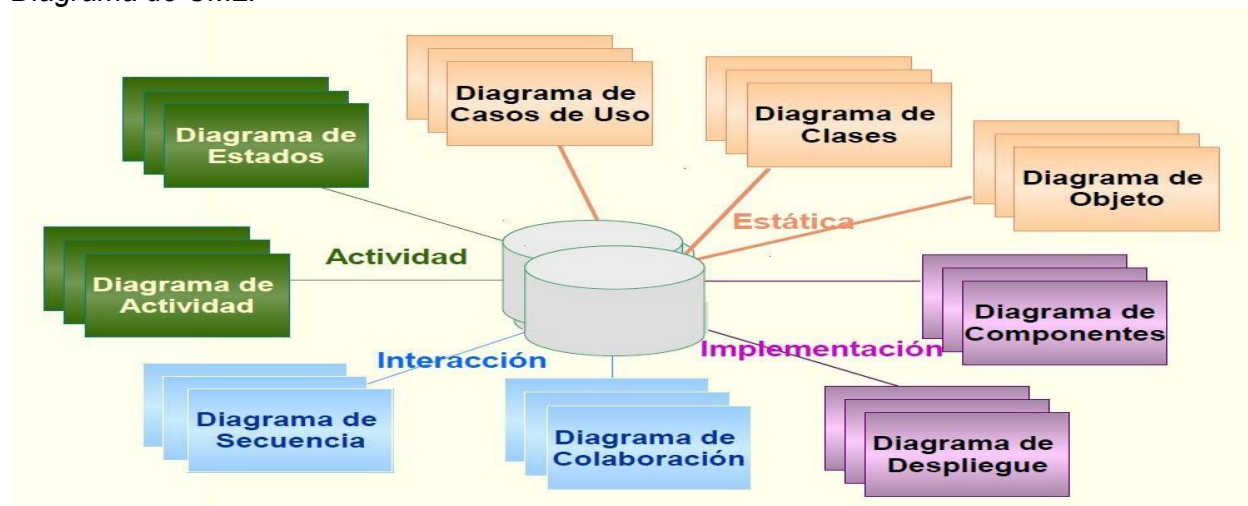
El Lenguaje Unificado de Modelado (UML) es un lenguaje visual que permite representar cómo funciona un sistema de software antes de escribir el código. Los diagramas de UML muestran la estructura del sistema y cómo se conectan sus componentes.

UML incluye diferentes tipos de diagramas, como diagramas de clases (que muestran los objetos y sus relaciones), diagramas de casos de uso (que muestran cómo los usuarios interactúan con el sistema) y diagramas de secuencia (que ilustran cómo fluyen los procesos o eventos en el tiempo). Estos diagramas ayudan a visualizar, especificar y documentar el sistema, facilitando la comunicación entre los miembros del equipo y simplificando el mantenimiento futuro (Sommerville, 2005, p. 298).

UML es una herramienta que combina claridad visual con precisión técnica, haciendo que tanto desarrolladores como personas no técnicas puedan entender cómo se organiza y funciona un sistema de software.

Figura 2

Diagrama de UML.



Nota. Diagrama que ilustra los principales casos de uso de un sistema. Fuente: David Aaron (2011). (<https://goldeneye-criollo.blogspot.com/2011/05/lenguaje-de-modelado-unificado-uml.html#>)

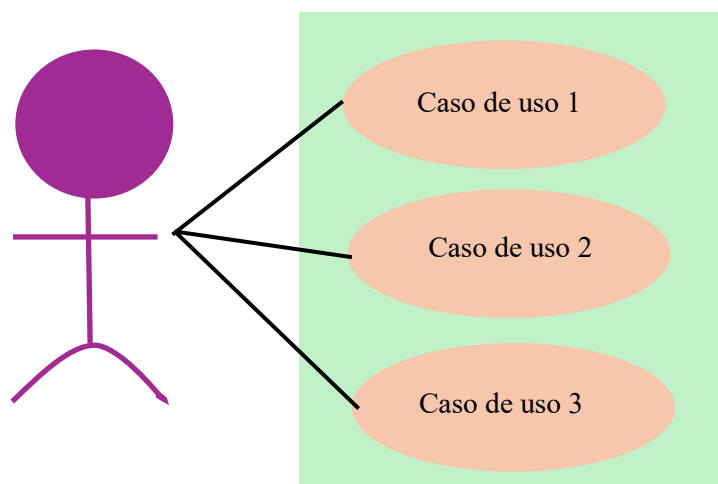
UML tiene diferentes tipos de diagramas que muestran el sistema desde varias perspectivas. Cada diagrama sirve para **ver una parte específica** del sistema y entenderlo mejor. Según el **Object Management Group (s.f.)**, estos diagramas se pueden agrupar en varias categorías:

Diagrama de caso de uso.

El diagrama de casos de uso sirve para mostrar lo que hace un sistema y cómo interactúa con su entorno desde el punto de vista del usuario. Incluye tres elementos principales: los casos de uso, que son las tareas o funciones que realiza el sistema; los actores, que son las personas, otros sistemas o elementos que interactúan con él; y las relaciones, que son las personas, otros sistemas o elementos que interactúan con él; y las relaciones, que indican cómo se conectan los actores con los casos de uso y cómo se relacionan entre sí, incluyendo jerarquías o dependencias.

Figura 3

Diagrama de clases.



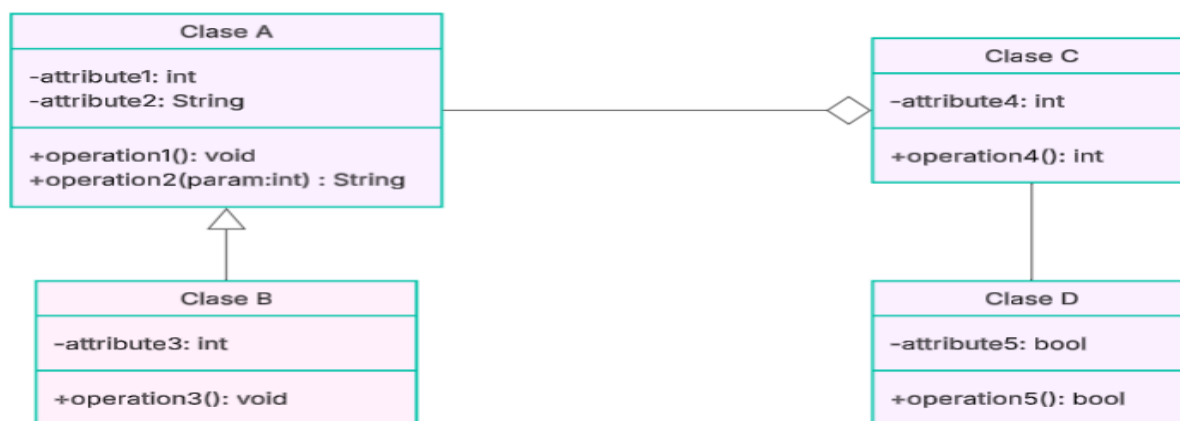
Nota. Diagrama de casos de uso elaborado por el autor. Fuente: Elaboración propia (2025).

Diagrama de clases.

El diagrama de clases muestra la estructura estática de un sistema, es decir, cómo se organizan y relacionan sus elementos principales. Está formado por clases y asociaciones. Las clases representan los objetos o conceptos del sistema y contienen atributos, que son sus características, y operaciones, que son las acciones que pueden realizar. Las asociaciones muestran cómo se conectan las clases entre sí y pueden incluir multiplicidad, composición, agregación y generalización, que describen distintos tipos de relaciones y dependencias entre las clases.

Figura 4

Diagrama de clases.



Nota. Diagrama de clases elaborado por el autor. Fuente: Elaboración propia (2025).

Diagrama de objetos.

El diagrama de objetos muestra cómo están organizados los elementos del sistema en un momento concreto, no de manera general. Es como una foto de la situación real del sistema, basada en lo que ya se definió en el diagrama de clases. Esto permite comprobar que la estructura y las relaciones definidas en las clases funcionan correctamente cuando se aplican a objetos reales del sistema.

Figura 5

Diagrama de componente



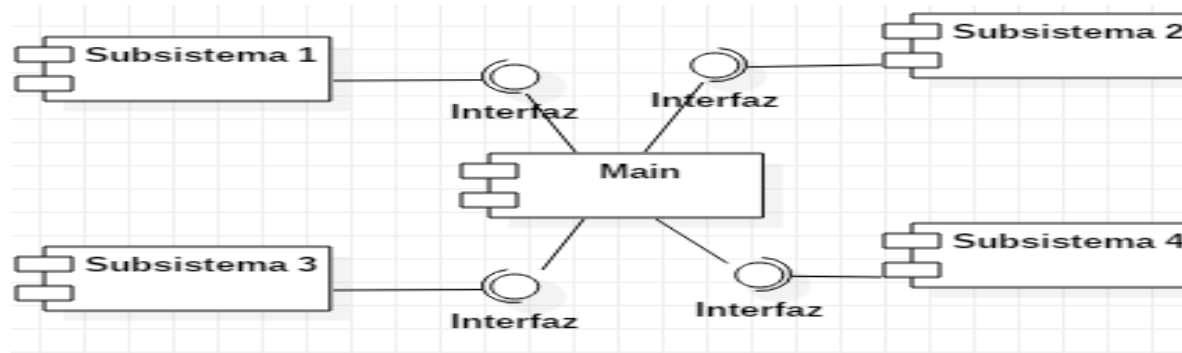
Nota. Diagrama de objetos elaborado por el autor. Fuente: Elaboración propia (2025).

Diagrama de componentes.

Este diagrama muestra cómo se organizan las partes físicas del sistema y cómo se conectan entre sí. Incluye componentes, que son los bloques o módulos del sistema; interfaces, que definen las acciones o servicios que cada componente puede ofrecer o usar; y dependencias, que indican las relaciones y conexiones entre componentes e interfaces.

Figura 6

Diagrama de componentes.



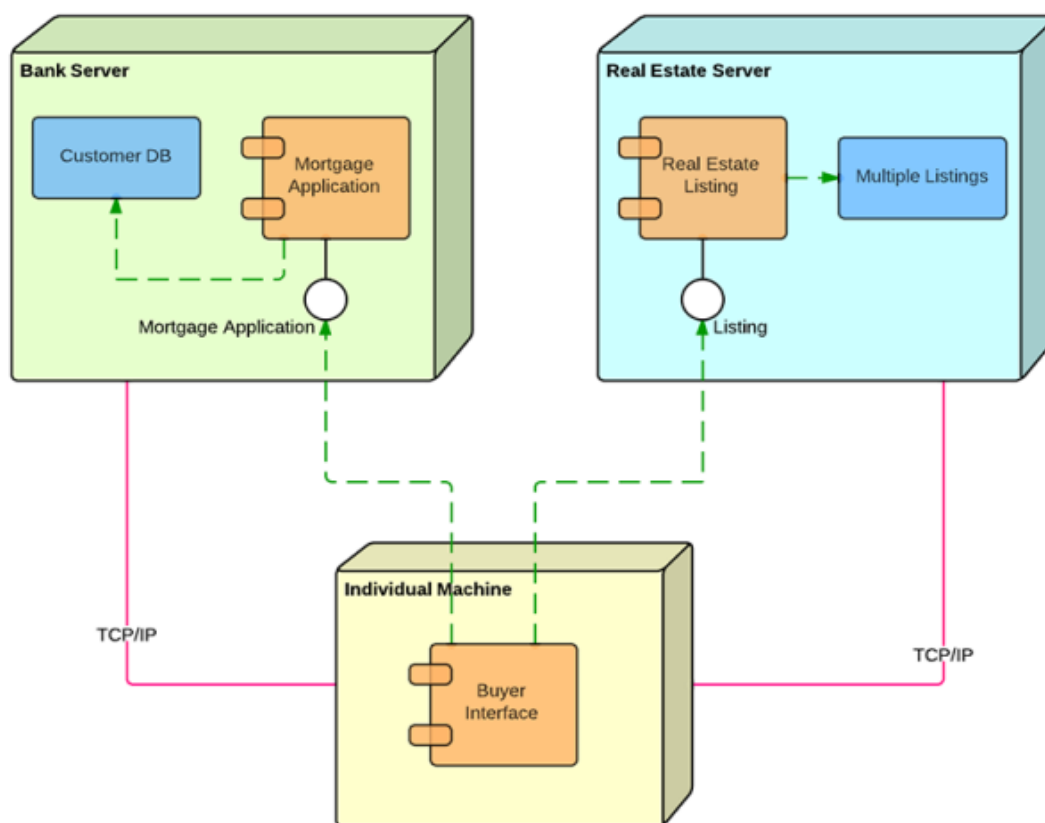
Nota. Primera aproximación al diagrama de componentes. Fuente: Magazine Pro (2023). (<https://diagramasuml.com/componentes/>)

Diagrama de despliegue o distribución.

El diagrama de despliegue muestra cómo se organiza la parte física del sistema, es decir, los equipos o dispositivos donde se ejecutará el software y cómo se conectan entre ellos. Está formado por nodos, componentes y asociaciones. Los nodos representan los recursos físicos, como servidores o computadoras, que pueden ejecutar el software. Las asociaciones indican cómo se conectan los nodos entre sí, por ejemplo, mediante redes como Ethernet. Los componentes son las partes del software que se ubican dentro de estos nodos.

Figura 7

Diagrama de despliegue.



Nota. Diagrama de implementación que representa la arquitectura física de un sistema. Fuente: Lucidchart (2023). (<https://www.lucidchart.com/pages/es/tutorial-de-diagramas-de-despliegue>)

Diagrama de secuencias.

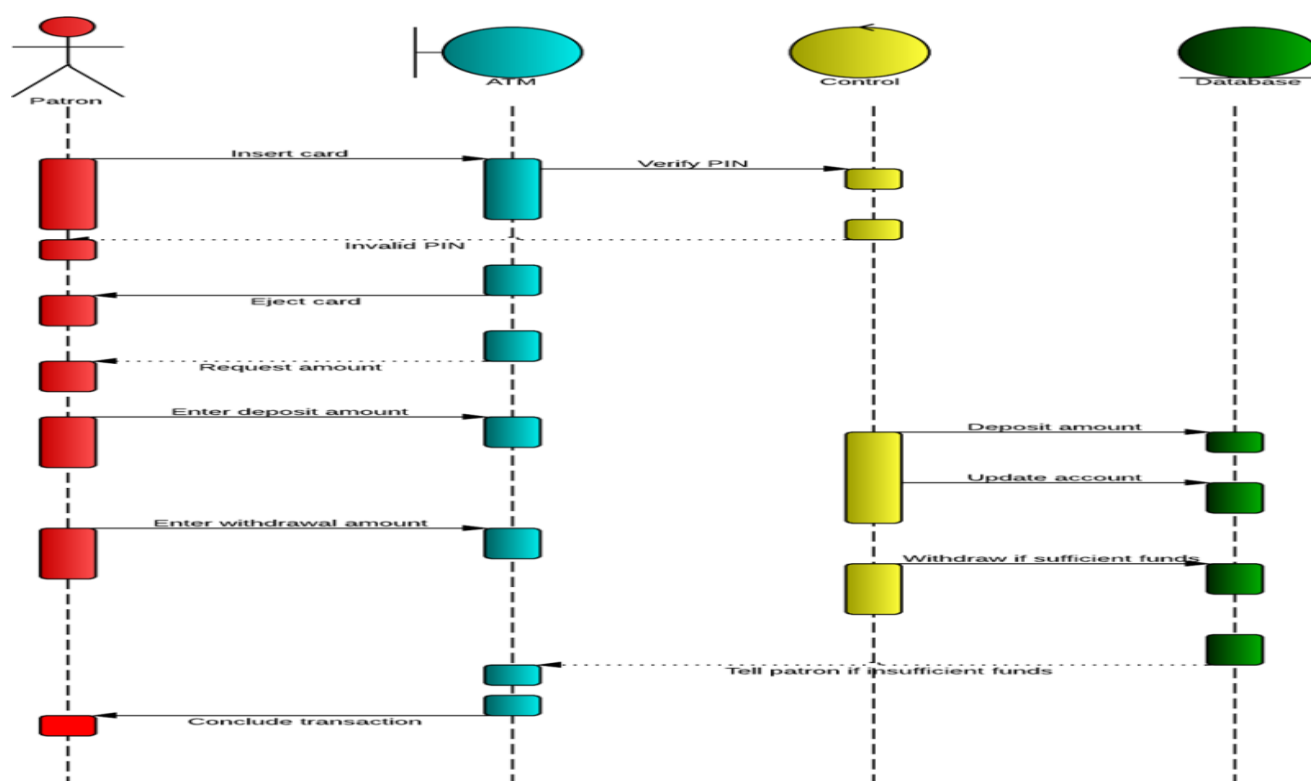
Este diagrama muestra cómo interactúan los objetos dentro del sistema a lo largo del

tiempo. Permite entender qué hace cada objeto, en qué momento y cómo se comunican entre sí.

Los elementos principales son: los roles de clase, que muestran cómo se comporta un objeto en un momento específico; la activación, que indica cuánto tiempo tarda en completar una acción; los mensajes, que son las comunicaciones entre los objetos y pueden ser de distintos tipos, como simples, sincronizados, asíncronos, rechazados o con límite de tiempo; las líneas de vida, que indican cuánto tiempo está presente un objeto en el sistema; y los loops, que muestran cuándo una acción se repite varias veces.

Figura 8

Diagrama de secuencia.



Nota. Diagrama de secuencia que muestra la interacción entre los objetos en el tiempo.
Fuente: Lucidchart (2023). (<https://www.lucidchart.com/pages/es/diagrama-de-secuencia>)

Diagrama de Actividades.

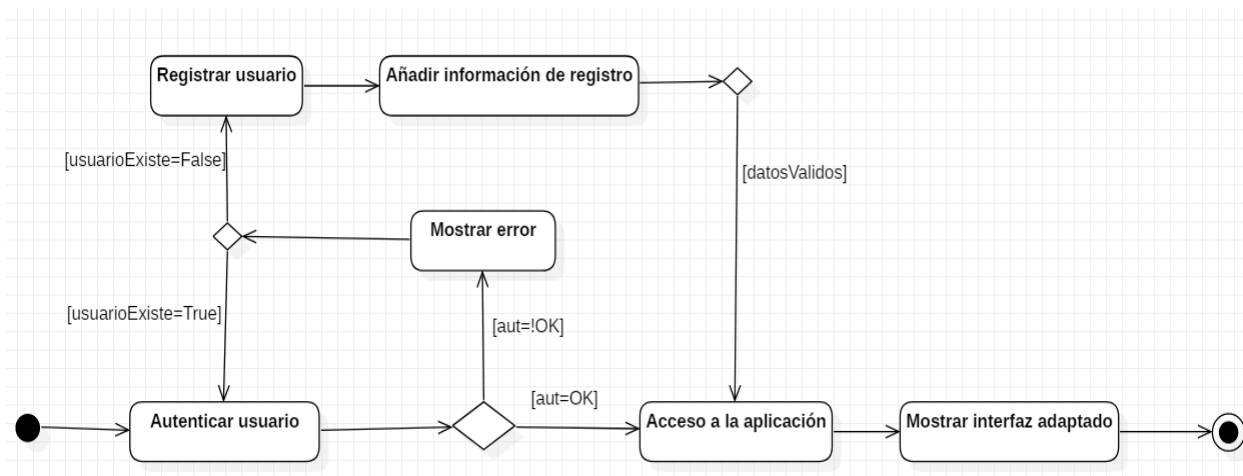
Este diagrama muestra cómo fluye el trabajo dentro del sistema. Cada actividad

representa algo que una clase o componente hace y, además, puede cambiar su estado mientras ocurre.

Se usa principalmente para entender y modelar el flujo de trabajo interno del sistema, mostrando de manera clara qué acciones se realizan y en qué orden.

Figura 9

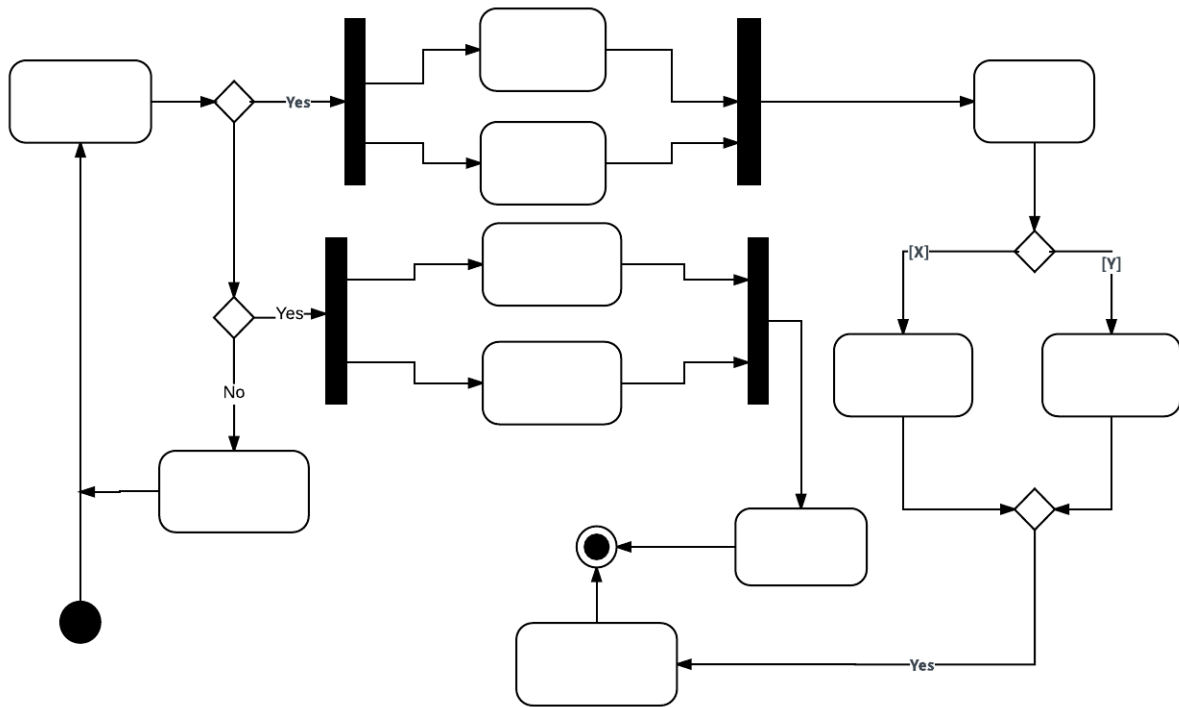
Diagrama de actividades



Nota. Ejemplo de diagrama de actividades utilizado en UML. Fuente: Magazine Pro (2023). (<https://diagramasuml.com/actividades/>)

Diagrama de Interacción.

El diagrama de interacción muestra cómo trabajan juntos los objetos del sistema para realizar una tarea, enfocándose en las interacciones y el orden de las acciones. Sirve para entender cómo se comunican los distintos elementos, qué mensajes se envían y en qué secuencia, ayudando a visualizar el flujo de trabajo y la colaboración entre los componentes.

Figura 10*Diagrama de actividades*

Nota. Ejemplo de diagramas de interacción en UML. Fuente: Lucidchart (2023). (<https://www.lucidchart.com/pages/es/tutorial-diagrama-de-actividades-uml>)

Metodología de Prototipado

El enfoque de prototipos es una técnica para el desarrollo de software que consiste en crear versiones iniciales del sistema de manera gradual, con la finalidad de comprobar funcionalidades, recibir comentarios de los usuarios y hacer modificaciones antes de crear el producto definitivo. Esta metodología es especialmente ventajosa en situaciones donde los requisitos del sistema no están completamente claros desde el comienzo, ya que brinda la oportunidad de explorar de manera más efectiva y tangible las necesidades verdaderas del usuario.

De acuerdo con Campos (2014), “este modelo es básicamente prueba y error ya que si

al usuario no le gusta una parte del prototipo significa que la prueba falló, por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho” (p. 11). Esta dinámica iterativa permite que el desarrollo se adapte de forma flexible, promoviendo una mejora continua del producto y reduciendo el riesgo de construir un software que no cumpla con las expectativas.

El proceso inicia con la definición rápida de los objetivos generales del software, seguido por la identificación de los requisitos conocidos y las áreas que requieren mayor claridad. Posteriormente, se construye un prototipo que es evaluado por el usuario, quien puede sugerir cambios que se incorporan en versiones posteriores. Como señala el autor, “cuanto más próximo se encuentre el prototipo al producto real, mejor será la evaluación” (Campos, 2014, p. 11), lo que evidencia la importancia de acercarse lo más posible a la experiencia final del usuario para obtener observaciones significativas.

Este modelo también permite validar aspectos como la interfaz, la lógica del sistema y la interacción humano-máquina, lo que contribuye a mejorar la calidad del software y a fortalecer la comunicación entre los involucrados en el proyecto.

Etapas Del Modelo De Prototipos

El modelo de prototipos avanza mediante varias fases que posibilitan la creación de versiones iniciales del software, su evaluación junto al usuario y su refinamiento hasta lograr un producto que funcione adecuadamente. Esta estrategia se fundamenta en la retroalimentación continua y en la evolución gradual del sistema, lo que la hace una metodología adaptable y enfocada en el usuario.

Recogida Y Perfeccionamiento De Requisitos. En esta fase se determinan los propósitos principales del software y se reúnen los requisitos existentes. No es necesario tener

una descripción completamente definida desde el principio, lo que permite continuar mientras se aclaran elementos técnicos y funcionales a lo largo del proceso (Campos, 2014, p. 12).

Diseño Y Modelado Ágil. Se elabora un diseño inicial del sistema que facilita la creación de un prototipo funcional sin la necesidad de especificar todos los detalles. Este diseño puede ser simple o elaborado, dependiendo del tipo de interfaz o sistema que se pretende representar.

Desarrollo Del Prototipo. En esta etapa se crea una versión básica del producto, que puede ir desde esquemas en papel hasta simulaciones digitales o incluso código ejecutable. Para hardware, se pueden emplear materiales como espuma, goma o cartón para mostrar físicamente el diseño (Campos, 2014, p. 12).

Revisión Del Prototipo Por El Cliente. El prototipo se muestra al usuario para que lo evalúe y ofrezca su opinión. Cuanto más se asemeje el prototipo al producto final, más exacta será la valoración. Sin embargo, también se pueden obtener buenos resultados con prototipos de menor fidelidad (Campos, 2014, p. 12).

Mejora Del Prototipo. A partir de las sugerencias del usuario, se llevan a cabo modificaciones y optimizaciones en el prototipo. Este proceso puede repetirse varias veces hasta que el usuario esté conforme con el resultado.

Prototipo Final. Una vez que el prototipo ha sido validado, se utiliza como fundamento para desarrollar el software definitivo, integrando las funcionalidades que han sido aprobadas y documentadas.

Figura 11

Ciclo de vida con prototipado



Nota. La imagen muestra cómo es el ciclo de vida del prototipo. Fuente: Clavijero (s. f.). (https://cursos.clavijero.edu.mx/cursos/142_sie/modulo3/contenidos/tema3.1.html)

El proceso inicia con la elaboración de un prototipo que represente el producto final en términos de apariencia y funcionalidad. Dependiendo del tipo de sistema, este prototipo puede ser tan simple como bocetos en papel o tan complejo como una versión operativa del software. En el caso de interfaces físicas, se pueden utilizar maquetas hechas con materiales básicos para simular la experiencia del usuario. La clave está en acercarse lo más posible al producto real, ya que esto mejora la calidad de la evaluación (Campos, 2014, p. 12).

Marco Legal

Todo proyecto de desarrollo tecnológico, especialmente aquellos enfocados en la gestión institucional, debe estar respaldado por un marco legal que garantice su legitimidad, el cumplimiento de las normativas vigentes y la protección de los derechos de los usuarios. Este marco establece las disposiciones jurídicas que regulan el tratamiento de la información, la protección de datos personales, la propiedad intelectual y la interoperabilidad entre sistemas. En el contexto educativo, resulta indispensable identificar las leyes, decretos y reglamentos que rigen el desarrollo de software, con el fin de asegurar que el proyecto se ajuste a los principios legales y éticos definidos por el Estado. De acuerdo con esto, las normas que acogen el desarrollo de nuestro proyecto son:

Las universidades en Colombia están bajo la supervisión del Ministerio de Educación Nacional, el cual establece un conjunto de leyes que son aplicables al servicio público educativo en el ámbito de la educación superior.

Ley 30 De 1992

Según la ley 30 de 1992, en su título 3, que se refiere al Régimen Especial De Las Universidades Del Estado y Otras Instituciones De Educación Superior Estatales, el capítulo V establece lo siguiente sobre el Régimen Financiero en relación con los recursos y espacios de las universidades:

Artículo 85: Los ingresos y el patrimonio de las entidades educativas estatales o oficiales de Educación Superior se compondrán de:

- a) Las asignaciones que se le otorguen dentro del presupuesto nacional, departamental, distrital o municipal.
- b) Los bienes tanto muebles como inmuebles que se encuentren en posesión en

este momento, así como aquellos que se obtengan en el futuro, junto con sus frutos y beneficios.

- c) Los ingresos que obtengan por concepto de matrículas, inscripciones y otros derechos.
- d) Los bienes que las personas jurídicas obtengan bajo cualquier concepto.

De acuerdo con el artículo 85 de la ley de 1992, se puede concluir que dependiendo de los bienes o propiedades que posean las universidades (entidades de educación superior que brindan servicios públicos), se puede afirmar que los ingresos y el patrimonio de estas instituciones podrían incrementarse. Esto se debe a que los alquileres y las rentas de los espacios públicos universitarios que no están siendo utilizados en ciertos momentos pueden generar ingresos a través de la explotación de estos recursos.

Artículo 86: Los presupuestos de las universidades nacionales, departamentales y municipales estarán compuestos por contribuciones del Presupuesto Nacional destinadas tanto al funcionamiento como a la inversión, por las aportaciones de los gobiernos locales, y por los ingresos y recursos propios de cada institución.

Según lo indicado en el artículo 86, se establece que los presupuestos de los servicios públicos de educación superior promoverán la generación de sus propios ingresos a través de ingresos propios o mediante el alquiler de sus espacios públicos a terceros.

Ley 1581 de 2012 – Protección de datos personales

Esta ley establece el marco general para el tratamiento de datos personales en Colombia. Su objetivo es garantizar el derecho que tienen todas las personas a conocer, actualizar y rectificar la información que se haya recogido sobre ellas en bases de datos o archivos. Además, exige el consentimiento previo, expreso e informado del titular para

cualquier tratamiento de sus datos (Congreso de Colombia, 2012).

Ley 603 de 2000 – Licenciamiento de software

Esta ley modifica el Código de Comercio y exige que las empresas incluyan en su informe de gestión una declaración sobre el cumplimiento de las normas de propiedad intelectual, especialmente en lo relacionado con el uso legal de software. Su propósito es combatir la piratería y garantizar que los programas utilizados estén debidamente licenciados (Congreso de Colombia, 2000).

En el marco de las disposiciones institucionales, el **Acuerdo No. 000010 de la Universidad de Cundinamarca** establece los derechos pecuniarios para el alquiler de los escenarios deportivos del Centro Académico Deportivo “CAD”. Dicho acuerdo, fundamentado en la autonomía universitaria reconocida por la Constitución Política de Colombia y en el Estatuto General de la institución, regula el uso y la prestación de servicios de estos espacios.

La versión completa del acuerdo se encuentra disponible en la carpeta **Anexos: Anexo c Acuerdo No. 000010 – Universidad de Cundinamarca**.

Marco Tecnológico

El presente proyecto se desarrollará conforme a los lineamientos técnicos definidos por la entidad, con el objetivo de garantizar la interoperabilidad, escalabilidad y sostenibilidad de la solución implementada. A continuación, se detallan los componentes tecnológicos seleccionados para el desarrollo de la aplicación web y la gestión de datos.

Bootstrap

Bootstrap es un framework front-end diseñado para facilitar el desarrollo de interfaces web responsivas mediante el uso de HTML, CSS y JavaScript. Su estructura modular y su sistema de rejilla permiten una rápida maquetación adaptable a distintos dispositivos, lo que lo convierte en una herramienta eficiente para proyectos con enfoque *mobile-first*. Según Castro y Rodríguez (2021), “Bootstrap proporciona una base sólida para el diseño web moderno, reduciendo la complejidad del código y mejorando la experiencia del usuario” (p. 87). En este proyecto, se emplea Bootstrap para garantizar coherencia visual, accesibilidad y eficiencia en el desarrollo de la interfaz gráfica.

Java

Java es un lenguaje de programación basado en objetos, muy común en el desarrollo de aplicaciones tanto empresariales como móviles y en la web. Su mayor ventaja es la posibilidad de ser portátil, gracias a la máquina virtual de Java (JVM), que hace posible la ejecución del mismo código en diversas plataformas sin realizar cambios. Según Abanto Morales (2021), “Java se ha consolidado como una herramienta versátil y robusta, ideal para proyectos que requieren escalabilidad, seguridad y mantenimiento a largo plazo” (p. 21). En el presente proyecto, Java se emplea como lenguaje base para la lógica de negocio, integrándose con patrones arquitectónicos como MVC y DAO, lo que facilita la organización del código y la reutilización de componentes.

JSTL

JSTL (JavaServer Pages Standard Tag Library) es una biblioteca de etiquetas estándar para JSP que permite simplificar la lógica de presentación en aplicaciones web desarrolladas en Java. Proporciona un conjunto de funciones para manipular datos, controlar flujo, acceder a bases de datos y gestionar internacionalización, todo ello sin necesidad de escribir código Java directamente en la página. Como señalan Sierra y Bates (2005), “JSTL permite separar la lógica de negocio de la presentación, fomentando un diseño más limpio y mantenible” (p. 412). En este proyecto, JSTL se emplea para estructurar dinámicamente el contenido en las vistas, facilitando la interacción entre el modelo y la interfaz de usuario bajo el patrón MVC.

MVC

El patrón arquitectónico Modelo-Vista-Controlador (MVC) permite estructurar aplicaciones dividiéndolas en tres componentes independientes: el modelo, que gestiona los datos y la lógica de negocio; la vista, encargada de la presentación visual; y el controlador, que actúa como intermediario entre ambos, gestionando las acciones del usuario. Esta separación de responsabilidades facilita el mantenimiento, la escalabilidad y la reutilización del código. Como explica Reenskaug, creador del patrón, “MVC promueve una arquitectura limpia al desacoplar la lógica de negocio de la interfaz de usuario, permitiendo que cada componente evolucione de forma independiente” (AppMaster, 2023). En el presente proyecto, MVC se implementa para organizar la estructura del sistema, optimizando la interacción entre capas y mejorando la claridad del desarrollo.

DAO

El patrón arquitectónico DAO (Data Access Object) permite separar la lógica de negocio de la lógica de acceso a datos, encapsulando las operaciones de persistencia en clases específicas. Esta separación favorece la mantenibilidad, la reutilización del código y la

independencia entre capas del sistema. Según Reactive Programming (s.f.), “el DAO encapsula toda la lógica de acceso de datos al resto de la aplicación, permitiendo que el sistema evolucione sin afectar la capa de persistencia”. En el presente proyecto, DAO se implementa para gestionar las operaciones CRUD sobre la base de datos, facilitando la integración con tecnologías como JDBC y estructuras como el patrón MVC.

Oracle

Oracle Database es un sistema de gestión de bases de datos relacional (RDBMS) ampliamente utilizado en entornos empresariales por su escalabilidad y seguridad. Diseñado para manejar grandes volúmenes de información, permite la administración eficiente de datos mediante estructuras normalizadas, soporte para transacciones ACID y herramientas avanzadas de recuperación. Según coronel y Morris (2017), “Oracle ofrece una arquitectura sólida que permite la gestión de datos complejos en aplicaciones críticas, con un enfoque en la integridad y el rendimiento” (p. 512). En el presente proyecto, Oracle se emplea como base de datos principal, integrándose con tecnologías como Java y DAO para garantizar la persistencia confiable de la información.

NetBeans

NetBeans es un entorno de desarrollo integrado (IDE) de código abierto, ampliamente utilizado en el desarrollo de aplicaciones basadas en Java. Su diseño modular y su interfaz intuitiva lo convierten en una herramienta robusta para programadores que buscan eficiencia, organización y compatibilidad con múltiples tecnologías. Originalmente desarrollado por Sun Microsystems y actualmente mantenido por Apache Software Foundation, NetBeans ha evolucionado para admitir diversos lenguajes de programación como PHP, HTML, JavaScript, C++ y Python, aunque su enfoque principal sigue siendo el desarrollo en Java.

Una de las principales ventajas de NetBeans es su capacidad para integrar múltiples fases del

ciclo de desarrollo en una sola plataforma: edición de código, compilación, depuración, pruebas, gestión de proyectos y control de versiones. Esto permite a los desarrolladores trabajar de manera más fluida, sin necesidad de recurrir a herramientas externas. Además, su sistema de gestión de proyectos facilita la organización de archivos, módulos y dependencias, lo que resulta especialmente útil en aplicaciones que implementan arquitecturas como Modelo-Vista-Controlador (MVC) y patrones como DAO y VO.

NetBeans también ofrece funciones avanzadas como autocompletado de código, navegación rápida entre clases y métodos, resaltado de sintaxis, y herramientas de depuración que permiten establecer puntos de interrupción, inspeccionar variables y ejecutar el programa paso a paso. Estas características no solo agilizan el desarrollo, sino que también reducen la probabilidad de errores y mejoran la calidad del software. Según IMMUNE Institute (2024), “NetBeans facilita la creación de aplicaciones estructuradas mediante módulos reutilizables, lo que optimiza el desarrollo del código”.

En el contexto del presente proyecto, NetBeans se emplea como plataforma principal para la codificación de componentes Java, la implementación de patrones arquitectónicos y la integración con tecnologías como JSTL para la capa de presentación. Su compatibilidad con servidores como Apache Tomcat y bases de datos como Oracle permite una configuración eficiente del entorno de ejecución, asegurando que el sistema sea escalable, mantenible y alineado con estándares profesionales.

Apache Tomcat

Apache Tomcat es un contenedor de aplicaciones Java y un servidor de internet que facilita la ejecución de tecnologías como Servlets y JSP, las cuales son fundamentales para la creación de aplicaciones web interactivas. Su ligereza, versatilidad y su capacidad de integración con el ecosistema Java lo hacen una herramienta perfecta para

entornos educativos y de negocio. Según HostGator (2025), “Tomcat proporciona funcionalidades adecuadas tanto para proyectos sencillos como para sistemas de mayor complejidad”. En este proyecto, se emplea Tomcat como plataforma de ejecución para implementar la lógica de negocio creada en Java utilizando el patrón MVC.

Software Libre

El software libre se refiere a programas informáticos que otorgan a los usuarios la libertad de ejecutar, estudiar, modificar y distribuir el código fuente sin restricciones legales o técnicas. Esta filosofía promueve la colaboración, la transparencia y el acceso equitativo al conocimiento tecnológico. Según Stallman (2010), “el software libre es una cuestión de libertad, no de precio. Para entender el concepto, hay que pensar en libertad de expresión, no en cerveza gratis” (p. 41). En el presente proyecto, se prioriza el uso de herramientas libres como NetBeans, Apache Tomcat y XAMPP, lo que permite una implementación flexible, ética y sostenible del sistema.

La elección de cada tecnología responde a criterios de pertinencia, funcionalidad y coherencia con los objetivos del proyecto. Se priorizaron herramientas que no solo fueran compatibles entre sí, sino que también ofrecieran estabilidad, escalabilidad y facilidad de mantenimiento. Por ejemplo, el uso de Java como lenguaje base garantiza una estructura sólida y orientada a objetos, ideal para aplicaciones que requieren modularidad y reutilización de código. Su integración con patrones como MVC y DAO permite una organización clara entre capas, lo que facilita el desarrollo y la depuración.

Asimismo, se seleccionó Bootstrap para la capa de presentación, por su capacidad de generar interfaces responsivas y estéticamente uniformes sin necesidad de codificación extensa. Esto permite optimizar tiempos de desarrollo y asegurar una experiencia de usuario coherente en distintos dispositivos. Complementariamente, JSTL se incorporó para simplificar

la lógica en las vistas JSP, evitando el uso de código Java incrustado y promoviendo buenas prácticas de separación entre lógica y presentación.

El uso de Oracle como sistema de gestión de bases de datos responde a la necesidad de manejar información estructurada de forma segura y eficiente, mientras que Apache Tomcat se emplea como servidor de aplicaciones por su compatibilidad con el ecosistema Java y su facilidad de configuración. Para el entorno de desarrollo, se eligió NetBeans, por su integración nativa con tecnologías Java, su interfaz intuitiva y sus herramientas de depuración que agilizan el ciclo de desarrollo.

Además, se optó por XAMPP como entorno local de pruebas, lo que permite validar funcionalidades antes de su despliegue en producción, reduciendo riesgos y asegurando la calidad del sistema. Finalmente, se valoró el uso de software libre como principio rector, no solo por razones éticas y económicas, sino también por la flexibilidad que ofrece al momento de adaptar y escalar el proyecto según futuras necesidades.

Marco Geográfico

El área geográfica donde se llevará a cabo la implementación de nuestro proyecto será en la universidad de Cundinamarca extensión Soacha, que se encuentra ubicada en la parte central del país, sobre la cordillera oriental, al sur de la sabana de Bogotá, Administrativamente hace parte al municipio de Soacha, en lo que se refiere a la división política del departamento de Cundinamarca.

Figura 12

Mapa Soacha



Nota: Relieve del municipio de Soacha. **Fuente:** Corporación Ambiental caminando el territorio. (2014)

Figura 13

Universidad de Cundinamarca



Nota: Extensión Soacha. **Fuente:** Universidad de Cundinamarca. (2023)

Capítulo 8. Estructura Temática

La estructura temática de este proyecto se orienta a la construcción de un **prototipo de software** para el control de alquiler de los recursos disponibles en el Campus Universitario de la Universidad de Cundinamarca, extensión Soacha. Este prototipo busca materializar los requerimientos previamente definidos y validar su viabilidad en términos de funcionalidad, usabilidad y pertinencia institucional. En este capítulo se presentan las herramientas empleadas, el proceso de diseño y las principales funcionalidades implementadas, con el propósito de demostrar la factibilidad de una solución tecnológica que optimice la gestión y disponibilidad de los recursos universitarios.

Fase 1 Análisis (Recolección y Refinamiento De Requisitos)

En esta fase de Recolección y Refinamiento de Requisitos, orientada específicamente al desarrollo del software para el control de alquiler de los recursos del campus universitario, se recopilan las necesidades funcionales, operativas y normativas que guiarán la construcción del sistema. A través de entrevistas, análisis documental y validación con los actores clave, se definen los procesos de reserva, disponibilidad, autorización y seguimiento de los recursos institucionales. Este levantamiento inicial permite establecer una visión clara del alcance del proyecto, garantizar su alineación con las políticas universitarias y sentar las bases para un diseño técnico coherente, eficiente y legalmente sustentado.

Levantamiento De Información

Durante esta etapa se aplicaron diversas técnicas metodológicas para identificar, comprender y documentar los requisitos del sistema de control de alquiler de recursos en el campus universitario de la Universidad de Cundinamarca, extensión Soacha. El propósito fue obtener una visión clara de la problemática actual, los procedimientos vigentes y las funcionalidades necesarias para el desarrollo del software.

Técnicas De Entrevista

Se llevaron a cabo reuniones con personal administrativo, en las que se formularon preguntas orientadas a conocer cómo se realiza actualmente el proceso de alquiler de recursos, qué dificultades se presentan en la gestión manual y qué funcionalidades consideran necesarias para optimizar el sistema. Esta interacción directa permitió identificar puntos críticos del procedimiento, necesidades específicas de control y trazabilidad, así como criterios institucionales que deben ser considerados en el diseño del software.

Mesas De Trabajo

Se organizaron espacios colaborativos con actores clave del proceso, donde se discutieron los flujos operativos, las limitaciones del modelo actual y las expectativas frente a una solución tecnológica. Estas sesiones facilitaron la construcción colectiva de propuestas, permitiendo priorizar funcionalidades como la visualización de disponibilidad, la reserva automatizada, la generación de reportes y el control de autorizaciones.

Análisis De Documentación

Se revisaron reglamentos internos, formatos de solicitud y documentos administrativos relacionados con la gestión de recursos. Este análisis permitió contextualizar los requisitos dentro del marco normativo vigente, identificar restricciones institucionales y asegurar que el sistema propuesto se alinee con las políticas y procedimientos establecidos por la universidad.

Casos de uso

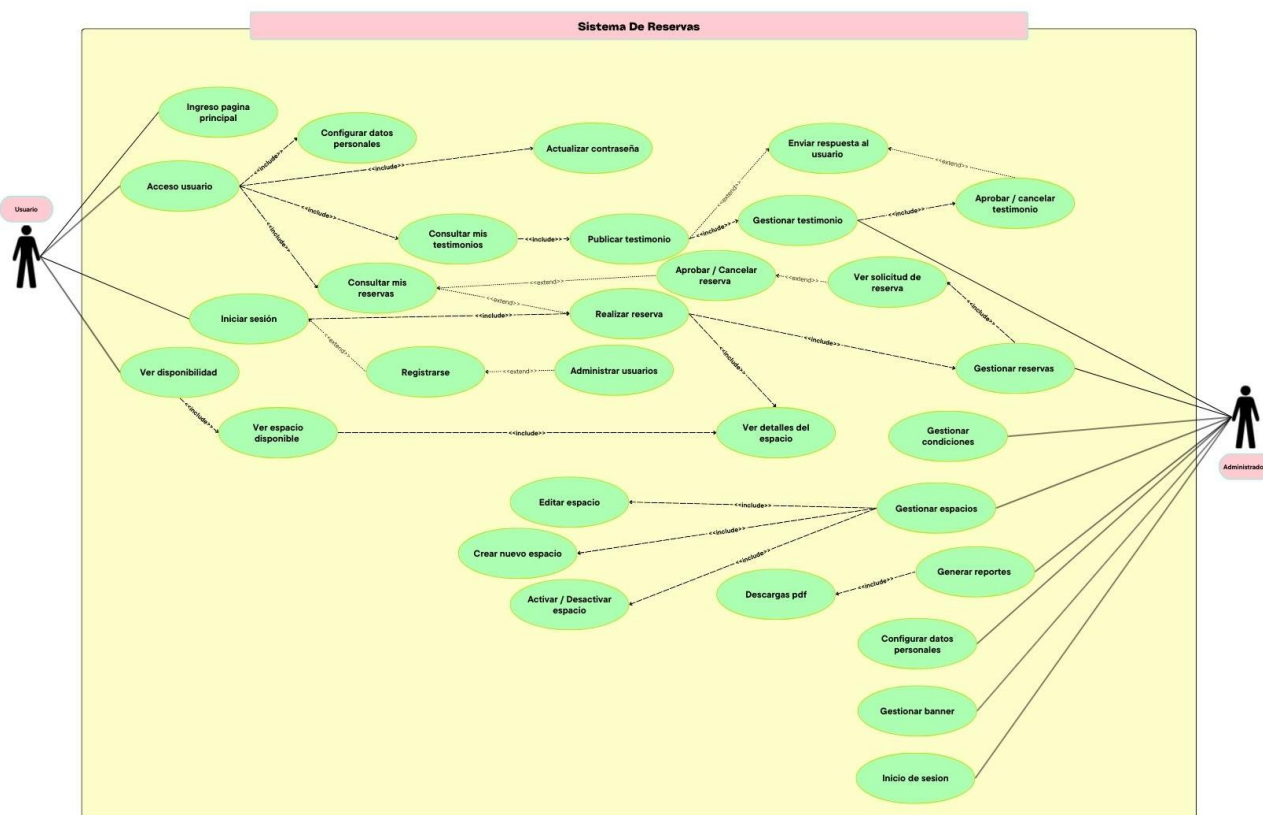
Los casos de uso permiten representar de manera estructurada las interacciones entre los usuarios y el sistema, facilitando la comprensión de los procesos funcionales que debe ejecutar el software. En el contexto del proyecto Software para el Control de Alquiler de los Recursos Disponibles del Campus Universitario de la Universidad de Cundinamarca, Extensión Soacha, estos casos reflejan las acciones clave que los usuarios realizarán dentro de la plataforma, asegurando trazabilidad entre los requerimientos funcionales y el diseño técnico.

En particular, se describe la interacción entre dos actores principales: el Usuario, quien accede a la página principal para verificar la disponibilidad de espacios, consultar sus características, realizar reservas, administrar sus datos personales y contraseña, revisar el estado de sus reservas y testimonios, publicar nuevos testimonios y cerrar sesión; y el Administrador, quien ingresa directamente al sistema para gestionar usuarios, banners informativos, condiciones de uso, espacios (creación, edición, activación/desactivación), reservas (aprobación, cancelación o marcación como pendiente con respuesta al usuario), testimonios (aprobación o rechazo con retroalimentación), y generar reportes en formato PDF.

Estas interacciones se plasman gráficamente en el diagrama de casos de uso que se presenta a continuación, como parte del diseño estructurado del sistema.

Diagrama 1

Caso de uso sistema general gestión de reservas



Nota Interacción entre Usuario y Administrador para consultar, reservar y gestionar espacios, reservas y testimonios. Fuente: Autoría Propia.

Tabla 2

Plantilla Casos de Uso Sistema de Reservas

SECCIÓN	DETALLES
ACTORES PRINCIPALES	-Usuario, Administrador
OBJETIVO	-Permitir al usuario reservar espacios, publicar testimonios y gestionar su información personal; permitir al administrador gestionar espacios, reservas y testimonios.
PRECONDICIONES	-El usuario debe tener acceso a internet y a la página principal. -El usuario debe contar con una cuenta o estar dispuesto a registrarse. -El administrador debe tener credenciales válidas. -Los espacios disponibles deben estar registrados en el sistema.
FLUJO NORMAL (USUARIO)	-Ingresa a la página principal. -Verifica disponibilidad de espacios. -Selecciona el espacio que desea reservar. -Visualiza los detalles completos del espacio. -Inicia sesión o se registra si no tiene cuenta. -Configura datos personales o actualiza contraseña si lo desea. -Realiza la reserva, que se envía al administrador. -Publica un testimonio sobre su experiencia, que el administrador revisa.
FLUJO NORMAL (ADMINISTRADOR)	-Gestiona reservas: revisa solicitudes, aprueba, cancela o deja pendiente. -Gestiona espacios: crea nuevos espacios, edita, activa o desactiva. -Gestiona testimonios: revisa, aprueba o cancela. -Genera reportes en PDF sobre reservas.
FLUJO NORMAL (SISTEMA)	-Envía notificaciones al usuario sobre el estado de reservas y testimonio. -Actualiza la información de reservas, testimonios y espacios.
FLUJOS ALTERNATIVOS / EXTENSIONES	-Usuario no registrado: se registra y continúa el flujo normal. -Usuario actualiza datos personales antes de reservar. -Administrador rechaza o cancela reserva o testimonio. -Espacio no disponible: el sistema informa al usuario.
POSTCONDICIONES	-Las reservas quedan registradas y asociadas a los usuarios. -Los testimonios quedan registrados y gestionados por el administrador. -Los espacios pueden ser agregados, editados o desactivados. -Los usuarios reciben confirmación o actualización sobre sus reservas -Los usuarios reciben confirmación o actualización sobre sus testimonios. -El administrador tiene visibilidad completa de reservas, testimonios y espacios.

Nota: Esta tabla resume el caso de uso del sistema de reservas, mostrando actores, objetivos, flujo normal, flujos alternativos y postcondiciones, para guiar la gestión de reservas, testimonios y espacios. Fuente: Autoría Propia.

Requerimientos Funcionales

Los requerimientos funcionales representan las capacidades específicas que debe tener el sistema para cumplir con los objetivos operativos del proyecto *Software para el Control de Alquiler de los Recursos Disponibles del Campus Universitario de la Universidad de Cundinamarca, Extensión Soacha*. Estos requisitos fueron definidos a partir del análisis realizado durante la fase de levantamiento de información, mediante entrevistas al personal administrativo, mesas de trabajo colaborativas y revisión de documentación institucional.

El enfoque estuvo orientado a comprender cómo se realiza actualmente el proceso de alquiler, qué dificultades enfrentan los usuarios en su gestión manual, y qué funcionalidades son necesarias para optimizar la eficiencia, trazabilidad y control del sistema. Los requerimientos funcionales establecidos para nuestro proyecto fueron:

Tabla 3

Requerimiento funcional RF1: Gestión de usuarios

Código	Requerimiento funcional	Descripción	Prioridad
RF1	Gestión de usuarios	Permitir el registro, edición y eliminación de usuarios, con autenticación mediante credenciales.	Alta

Nota: administra el registro, edición y eliminación de usuarios. Fuente: Autoría Propia.

Tabla 4*Requerimiento funcional RF2: Gestión de recursos*

Código	Requerimiento funcional	Descripción	Prioridad
RF2	Gestión de recursos	Registrar y administrar los recursos disponibles (salones, laboratorios, equipos, etc.) y mostrar su disponibilidad en tiempo real.	Alta

Nota: Registra y muestra la disponibilidad de recursos en tiempo real. Fuente: Autoría Propia.

Tabla 5*Requerimiento funcional RF3: Reservas de recursos*

Código	Requerimiento funcional	Descripción	Prioridad
RF3	Reservas de recursos	Permitir que los usuarios realicen solicitudes de alquiler indicando fecha, hora de inicio y hora de fin.	Alta

Nota: Solicitar reservas indicando fecha, hora de ingreso y salida. Fuente: Autoría Propia.

Tabla 6*Requerimiento funcional RF4: Aprobación de reservas*

Código	Requerimiento funcional	Descripción	Prioridad
RF4	Aprobación de reservas	Permitir a los administradores aprobar, rechazar y finalizar la reserva.	Alta

Nota: Permite aprobar, rechazar o finalizar la solicitud de reserva. Fuente: Autoría Propia.

Tabla 7*Requerimiento funcional RF5: Control de disponibilidad*

Código	Requerimiento funcional	Descripción	Prioridad
RF5	Control de disponibilidad	Bloquear automáticamente los recursos que ya estén reservados en un horario específico.	Alta

Nota: Bloquea recursos ya reservados para evitar conflictos. Fuente: Autoría Propia.

Tabla 8*Requerimiento funcional RF6: Historial de alquileres*

Código	Requerimiento funcional	Descripción	Prioridad
RF6	Historial de alquileres	Generar un registro de alquileres por usuario y por recurso, con opción de consulta y filtrado.	Alta

Nota: Genera un registro de alquileres con opción de consulta. Fuente: Autoría Propia.

Tabla 9*Requerimiento funcional RF7: Notificaciones*

Código	Requerimiento funcional	Descripción	Prioridad
RF7	Notificaciones	Informar al usuario sobre el estado de sus solicitudes (aceptada, rechazada, pendiente).	Alta

Nota: Informa a los usuarios sobre el estado de sus solicitudes. Fuente: Autoría Propia.

Tabla 10*Requerimiento funcional RF8: Reportes*

Código	Requerimiento funcional	Descripción	Prioridad
RF8	Reportes	Generar reportes de utilización de recursos (frecuencia, disponibilidad y demanda).	Alta

Nota: Genera reportes de utilización de recursos. Fuente: Autoría Propia.

Requerimientos No Funcionales

Los requerimientos no funcionales establecidos para el proyecto Software para el Control de Alquiler de los Recursos Disponibles del Campus Universitario de la Universidad de Cundinamarca, Extensión Soacha definen las condiciones de calidad, seguridad y rendimiento bajo las cuales debe operar el sistema. Estos requisitos fueron identificados como resultado del análisis técnico y contextual realizado durante la fase de recolección de información, y son fundamentales para garantizar que el software sea confiable, eficiente y adaptable al entorno administrativo del campus. Los requerimientos no funcionales establecidos para nuestro proyecto fueron:

Tabla 11*Requerimiento no funcional RNF1: Usabilidad*

Código	Requerimiento no funcional	Descripción	Prioridad
RNF1	Usabilidad	El sistema debe tener una interfaz intuitiva y fácil de usar por usuarios con conocimientos básicos en informática.	Alta

Nota: El sistema tendrá una interfaz intuitiva, fácil de usar. Fuente: Autoría Propia.

Tabla 12*Requerimiento no funcional RNF2: Idioma*

Código	Requerimiento no funcional	Descripción	Prioridad
RNF2	Idioma	El prototipo debe estar disponible en idioma español.	Alta

Nota: El sistema debe estar disponible en idioma español. Fuente: Autoría Propia.

Tabla 13*Requerimiento no funcional RNF3: Rendimiento*

Código	Requerimiento no funcional	Descripción	Prioridad
RNF3	Rendimiento	Las consultas deben responder en un tiempo máximo de 50 segundos.	Alta

Nota: El sistema debe ofrecer tiempos de respuesta ágiles. Fuente: Autoría Propia.

Tabla 14*Requerimiento no funcional RNF4: Capacidad de usuarios*

Código	Requerimiento no funcional	Descripción	Prioridad
RNF4	Capacidad de usuarios	El sistema debe soportar al menos 50 usuarios concurrentes en la fase de prototipo.	Alta

Nota: El prototipo debe ser capaz de manejar una concurrencia. Fuente: Autoría Propia.

Tabla 15*Requerimiento no funcional RNF5: Accesibilidad*

Código	Requerimiento no funcional	Descripción	Prioridad
RNF5	Accesibilidad	Debe funcionar en navegadores web comunes (Chrome, Edge) sin necesidad de instalación adicional.	Alta

Nota: Se garantiza accesibilidad desde navegadores estándar. Fuente: Autoría Propia.

Tabla 16*Requerimiento no funcional RNF6: Seguridad*

Código	Requerimiento no funcional	Descripción	Prioridad
RNF6	Seguridad	Las credenciales de usuario deben protegerse mediante cifrado básico y con control de permisos.	Alta

Nota: Se busca proteger los datos de acceso de los usuarios. Fuente: Autoría Propia.

Tabla 17*Requerimiento no funcional RNF7: Escalabilidad*

Código	Requerimiento no funcional	Descripción	Prioridad
RNF7	Escalabilidad	El diseño debe permitir la evolución hacia un sistema completo en fases posteriores.	Alta

Nota: El sistema tendrá una interfaz intuitiva, fácil de usar. Fuente: Autoría Propia.

Requerimientos No Funcionales De Software Y Hardware

En esta sección se presentan las especificaciones mínimas de software y hardware necesarias para asegurar el funcionamiento estable y eficiente del sistema dentro del entorno tecnológico del campus universitario.

Tabla 18

Requerimientos de hardware para usuarios finales

Componente	Requerimiento mínimo	Requerimiento recomendado
Procesador	1.6 GHz	Intel Core i3 o superior
Memoria RAM	2 GB	4 GB o más
Almacenamiento	10 GB libres en disco	50 GB en SSD
Pantalla	Resolución mínima 1366 x 768	Resolución Full HD (1920 x 1080)
Conectividad	Acceso a red LAN o Wi-Fi estable	Wi-Fi institucional de 10 Mbps o más
Periféricos	Teclado y mouse	Teclado, ratón y auriculares

Nota: Requerimientos de hardware para el uso del sistema en equipos de los usuarios finales.

Fuente: Autoría Propia.

Tabla 19

Requerimientos de software para usuarios finales

Componente	Requerimiento
Sistema operativo	Windows 7 o superior
Navegadores soportados	Google Chrome (versión 90+), Microsoft Edge (versión 90+)
Aplicaciones adicionales	Lector de PDF para reportes exportados
Seguridad	Navegador actualizado con soporte HTTPS/TLS

Nota: Requerimientos de software para usuarios finales. Fuente: Autoría Propia.

Fase 2 Diseño (Modelado, Diseño Rápido)

Durante la fase de diseño, se adopta la metodología de prototipos como enfoque iterativo para representar visual y funcionalmente el sistema antes de su desarrollo completo. Esta estrategia permite validar tempranamente los requerimientos mediante representaciones tangibles que facilitan la retroalimentación de los usuarios y el refinamiento progresivo de la solución. Como señala Pressman (2010), “el prototipo sirve como un mecanismo para identificar errores en los requisitos, mejorar la comprensión del sistema y explorar alternativas de diseño” (p. 245).

En esta etapa se emplearán herramientas como mockups para ilustrar la interfaz gráfica y la navegabilidad del sistema. Se desarrollarán casos de uso que describen las interacciones entre los actores y el sistema, complementados por diagramas UML que formalizan la estructura y comportamiento del software: el diagrama de clases modelará las entidades y sus relaciones; el diagrama de actividades representará los flujos de procesos; el diagrama de estados mostrará los cambios de estado de los objetos; y el diagrama de flujos permitirá visualizar la lógica secuencial de las operaciones. Estos artefactos no solo garantizan trazabilidad técnica, sino que también alinean el diseño con los estándares institucionales y las expectativas funcionales.

Mockups

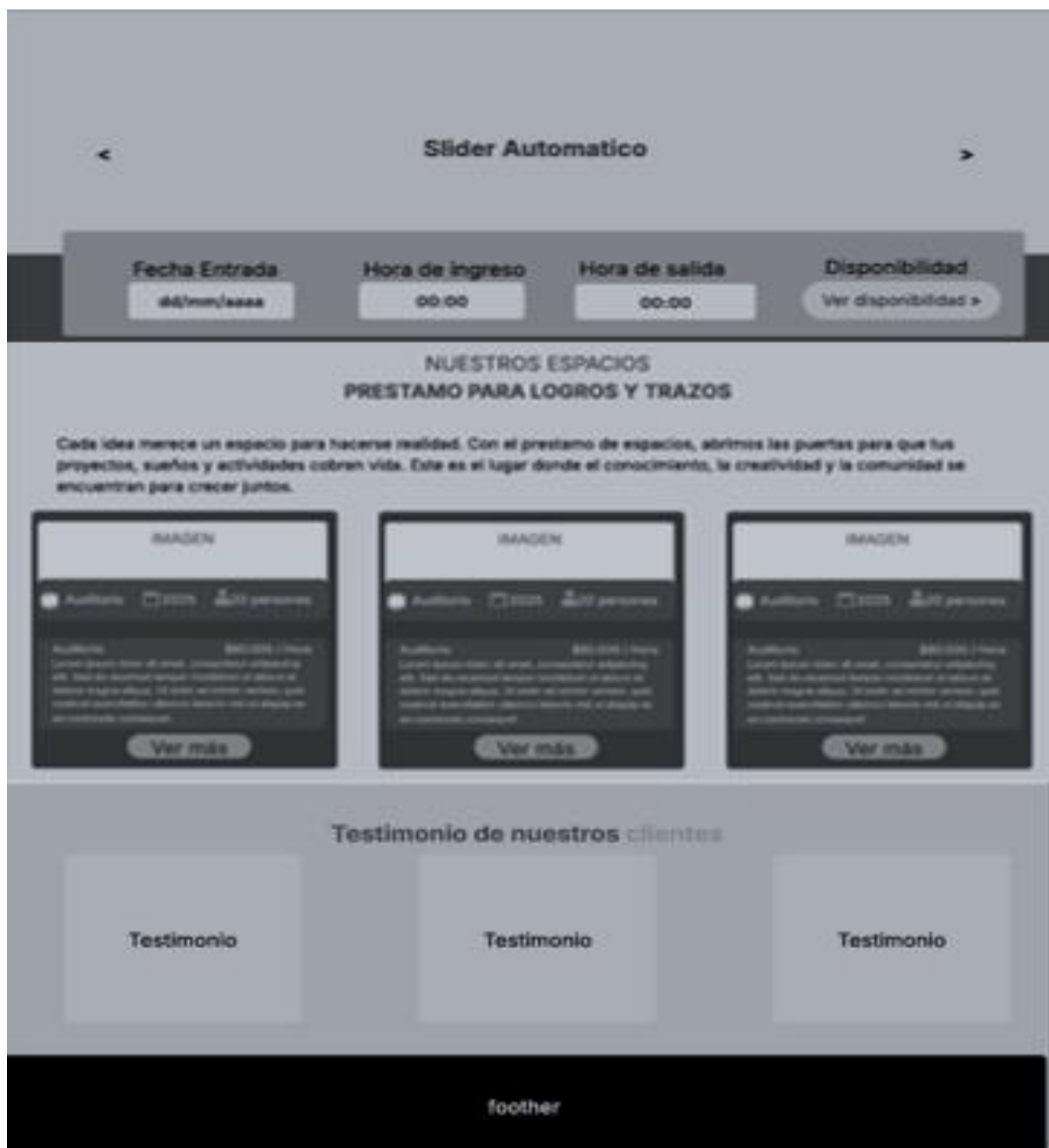
Los mockups permiten visualizar cómo se estructurarán los contenidos, los espacios disponibles, los menús y las funcionalidades que ofrecerá la aplicación (**ver Anexo a**). Este software ha sido diseñado para optimizar la gestión integral de alquiler de espacios, proporcionando una experiencia intuitiva y eficiente tanto para administradores como para usuarios finales. Facilita el control de la disponibilidad de los espacios, la reserva en tiempo real y el seguimiento detallado de clientes y reservas. Cada pantalla ha sido concebida para

garantizar una navegación fluida, claridad en la información y una interacción profesional, asegurando que la aplicación cumpla con los más altos estándares de usabilidad y diseño.

En el mockup de la interfaz principal se representa la interfaz de un sistema institucional para la reserva de espacios, orientado a facilitar la gestión eficiente de espacios disponibles, como auditorios y ambientes multifuncionales. La estructura combina elementos visuales claros con funcionalidades interactivas que permiten al usuario consultar disponibilidad, explorar opciones y realizar reservas de forma intuitiva. En la parte superior se encuentra una barra de navegación con accesos a inicio, lista de espacios, contacto e inicio de sesión. El módulo central incluye un formulario para seleccionar fecha de entrada, hora de ingreso y salida, junto con un botón para verificar disponibilidad. Al consultar, el sistema muestra un mensaje positivo confirmando que hay espacios disponibles, o de lo contrario un mensaje que indicará que no hay espacios disponibles, seguido de un calendario automático que indica los días reservados y libres. Los espacios disponibles se presentan en tarjetas informativas que detallan tipo de espacio, capacidad, disponibilidad, costo por hora y una breve descripción del uso recomendado, con opción de ver más información. Luego, en otro apartado, aparecerán tarjetas con los espacios que se encuentran publicados, organizadas de forma visualmente coherente para facilitar la exploración. Posteriormente, se incluye una sección responsiva que destaca los beneficios de la reserva de espacios, como disponibilidad garantizada, organización eficiente, apoyo institucional y flexibilidad de uso. Finalmente, la sección de testimonios de clientes refuerza la credibilidad del servicio, ofreciendo espacio para compartir experiencias positivas y fomentar la confianza en el sistema.

Figura 14

Pantalla principal de consulta y reserva de espacios disponibles

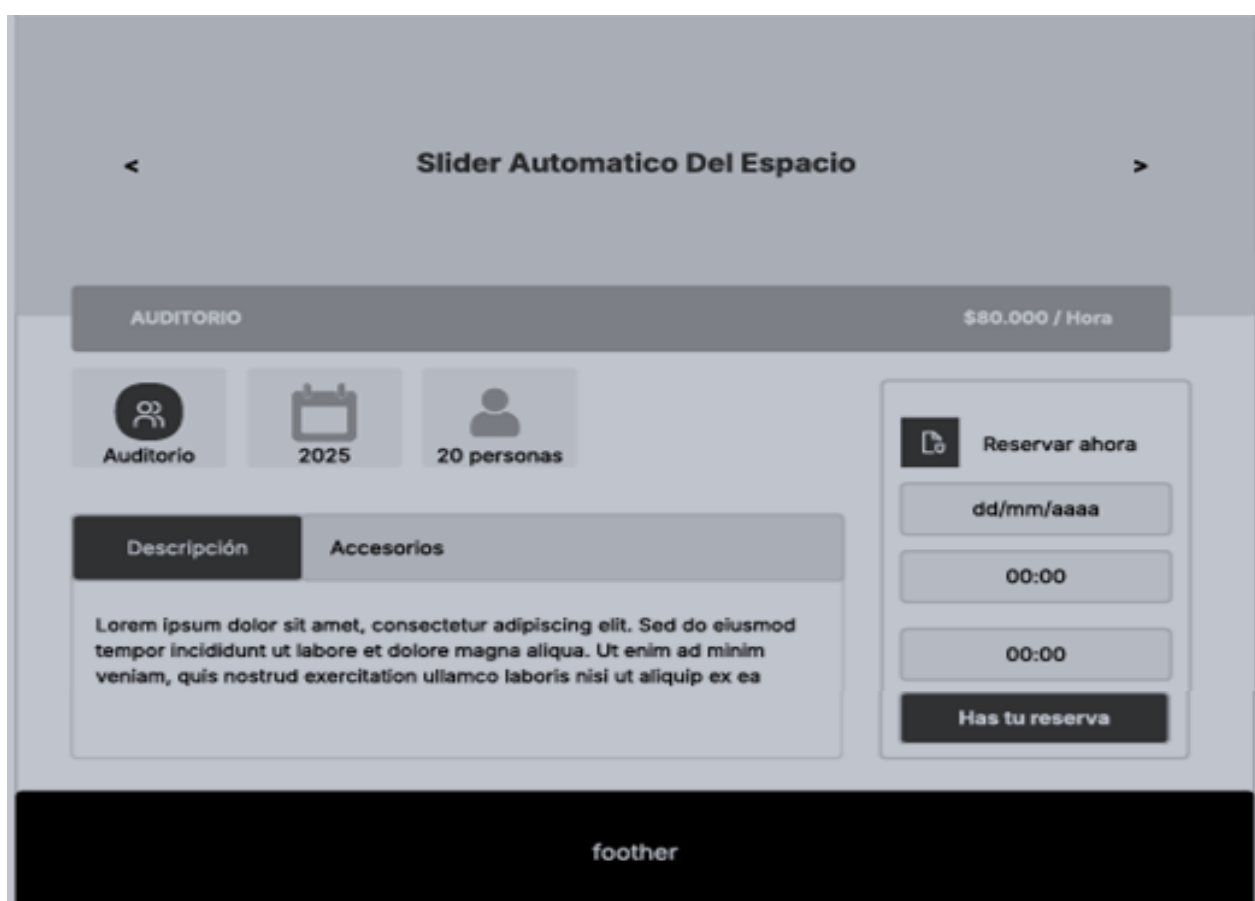


Nota: La figura muestra la interfaz principal del sistema de reserva de espacios, con los módulos de consulta de disponibilidad, tarjetas informativas y secciones de beneficios y testimonios.
Fuente: Autoría Propia

Este mockup muestra la vista detallada de un espacio dentro del sistema de reservas. La interfaz incluye una barra de navegación superior, un slider destacado del espacio, y una tarjeta con información clave como tipo, capacidad, año de referencia y tarifa por hora. El usuario puede consultar la descripción del espacio, revisar accesorios disponibles y realizar la reserva mediante un formulario con campos para fecha, hora y mensaje personalizado. La estructura está diseñada para facilitar la interacción directa y brindar una experiencia clara, funcional y orientada a la acción.

Figura 15

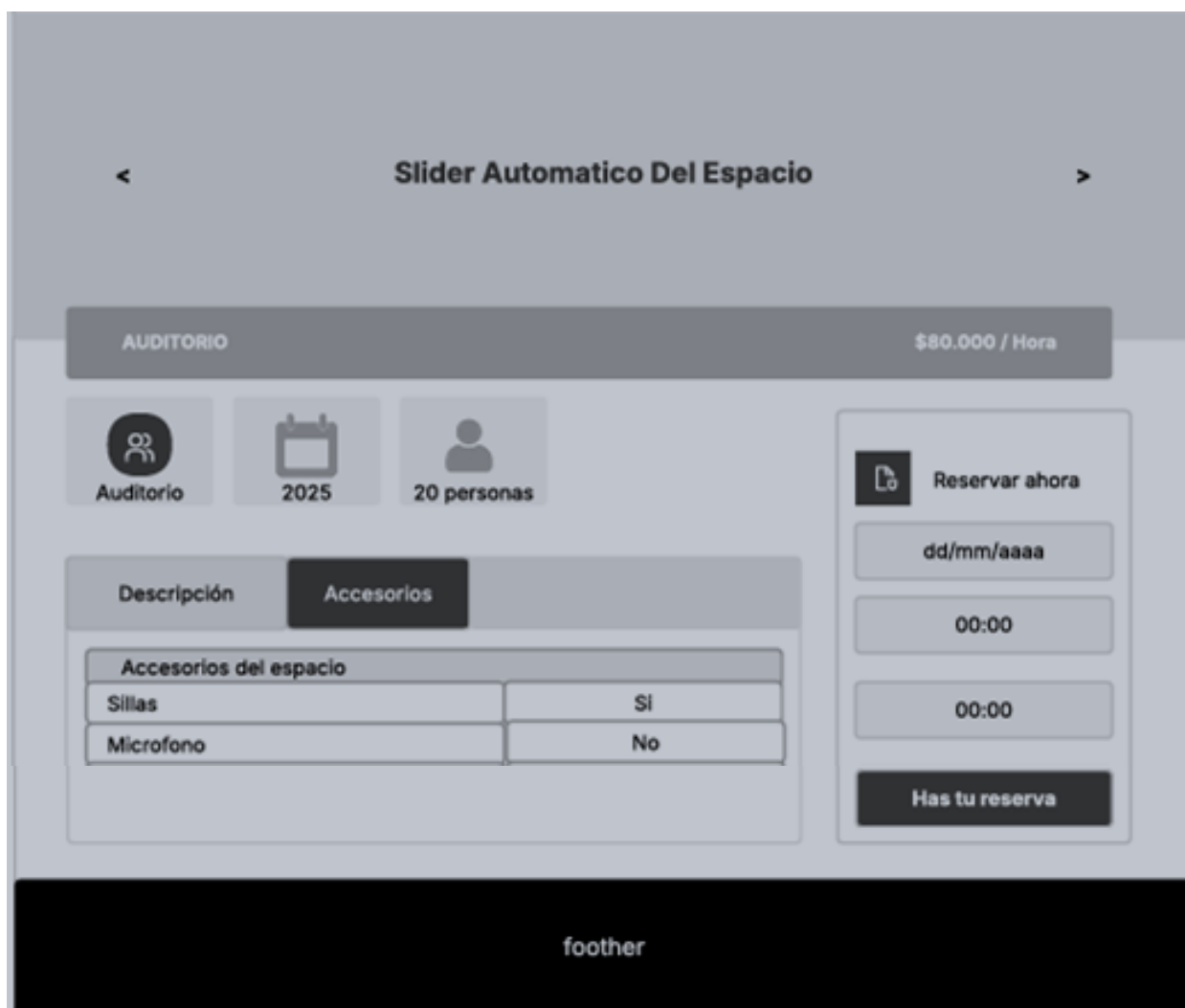
Vista detallada de espacio disponible en el sistema de reservas



Nota: El mockup presenta la interfaz de visualización individual de un espacio institucional disponible para reserva. Incluye información clave del espacio y un formulario lateral que permite al usuario completar la solicitud de manera rápida y personalizada. Autoría Propia.

Figura 16

Vista de accesorios disponibles para espacio en reserva



Nota: Interfaz que permite consultar los elementos complementarios disponibles en el espacio antes de realizar la reserva. Autoría Propia.

Este mockup muestra la interfaz de inicio de sesión con campos para correo electrónico y contraseña, botón de ingreso, y enlaces para registro y recuperación de contraseña.

Figura 17

Vista de interfaz para inicio de sesión

The image shows a mobile application interface for a login screen. At the top, there is a header with the text "Slider Automatico Del Espacio" centered between two navigation arrows. Below the header is a large, light gray rectangular area containing the login form. The form is titled "Iniciar Sesión!". It features two input fields: "Correo electronico*" and "Contraseña*". Below these fields is a rounded rectangular button labeled "INGRESAR". Underneath the button, there are two lines of text: "¿Aún no tienes una cuenta? Regístrate aquí" and "¿Olvidaste tu contraseña?". At the bottom of the screen, there is a dark gray footer area with the word "footer" centered.

Nota: Interfaz que permite al usuario autenticarse mediante correo electrónico y contraseña para acceder a las funcionalidades del sistema. Autoría Propia.

Este mockup muestra la interfaz de registro de usuario. Incluye un formulario con campos para nombre completo, identificación, correo electrónico, teléfono y contraseña.

Figura 18

Vista de interfaz para registro de usuario

The image shows a mobile application interface for user registration. At the top, there is a header bar with the text "Slider Automatico Del Espacio" and navigation arrows on either side. Below the header is a central registration form titled "Registro". The form contains six input fields, each with a label and an asterisk indicating it is required: "Nombre completo*", "Numero identificación*", "Correo electronico*", "Numero de telefono*", "Contraseña*", and "Confirmar contraseña*". Below these fields is a large, rounded "Regístrate" button. Underneath the button, there is a link that says "¿Ya tienes una cuenta? Ingresar aquí". At the bottom of the screen, there is a dark footer bar with the word "footer" centered.

Nota: Permite crear una cuenta ingresando datos personales y credenciales de acceso. Autoría Propia.

La interfaz permite al usuario editar sus datos personales como nombre, correo electrónico, número de identificación y teléfono. También ofrece acceso a funciones como actualización de contraseña, gestión de reservas, publicación de testimonios y cierre de sesión.

Figura 19

Vista de configuración de usuario en la plataforma de reservas

The screenshot shows the user configuration page in the ReservaEspacios platform. The page has a dark sidebar on the left with the following navigation items: Dashboard, Mis Reservas, Mis Testimonios, Notificaciones, Mi Perfil, and Contáctenos. The main content area is light gray and features a welcome message "¡Bienvenido!" at the top right, along with the user name "Bienvenido, samuel" and a "Cerrar Sesión" button. Below the welcome message are three action cards: "Tus espacios" (Consultar los espacios asignados o disponibles), "Tus testimonios" (Revisar o editar los testimonios compartidos), and "Realiza tu reserva" (Reservar un espacio en pocos clics). The "Mis Reservas" section contains a table with the following data:

ID	RECURSO	FECHA	HORA	ESTADO
122	62	2025-10-28	09:01 - 10:07	Pendiente
121	61	2025-10-27	08:00 - 10:00	Pendiente
123	62	2025-10-29	12:22 - 14:28	Pendiente

The "Notificaciones" section shows "No tienes notificaciones" and a button "Marcar todas como leídas".

Nota: Permite al usuario actualizar su información personal y gestionar opciones de cuenta. Autoría Propia.

Figura 20

Vista de actualización del perfil de usuario

The screenshot shows the user profile update page in the ReservaEspacios platform. The page has a dark sidebar on the left with the following navigation items: Dashboard, Mis Reservas, Mis Testimonios, Notificaciones, Mi Perfil, and Contáctenos. The main content area is light gray and features a form titled "Editar Perfil" at the top right, along with the user name "Bienvenido, samuel" and a "Cerrar Sesión" button. The form contains the following fields:

- Nombre completo: samuel
- Correo electrónico: sam@gmail.com
- Teléfono: 32435435
- Nueva contraseña: *****

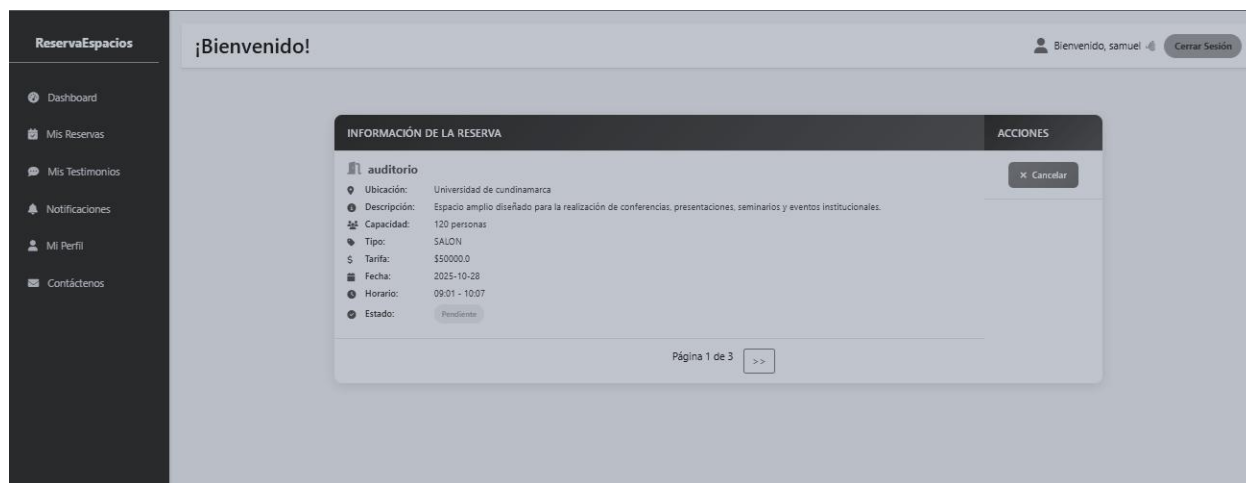
Below the fields are two buttons: "Guardar Cambios" (highlighted in dark gray) and "Cancelar" (light gray).

Nota: Permite modificar la contraseña asociada a la cuenta de forma segura. Autoría Propia.

El mockup presenta el historial de reservas realizadas por el usuario, incluyendo nombre del espacio, fecha, hora de ingreso y salida, junto al estado de cada solicitud.

Figura 21

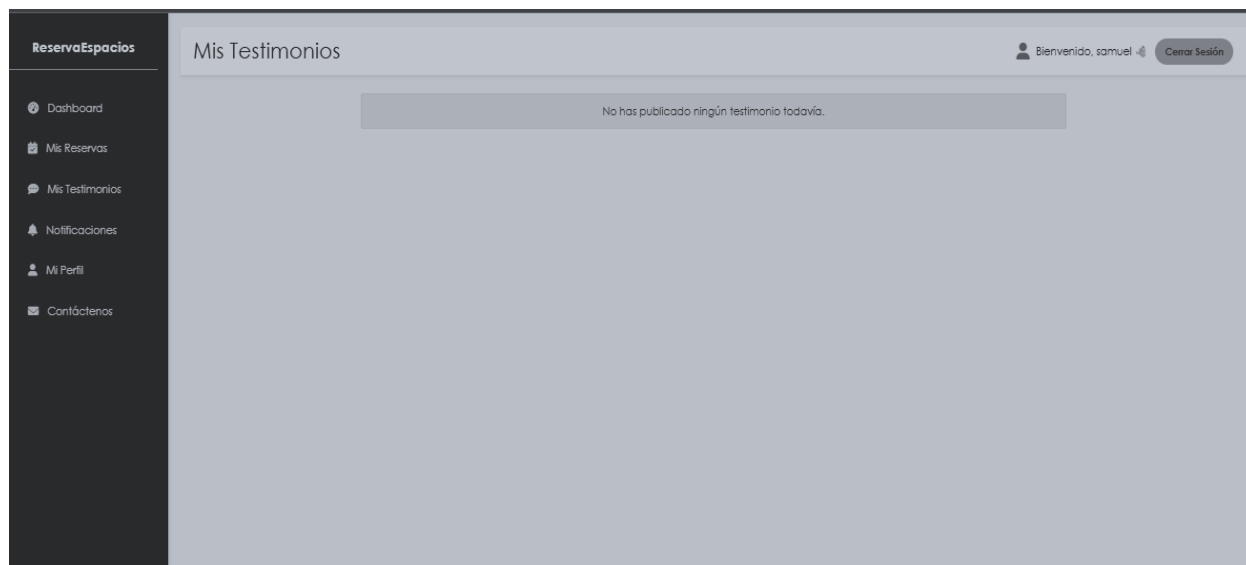
Vista de reservas del usuario



Nota: Permite al usuario consultar el estado de sus reservas y verificar detalles como fecha, horario y espacio asignado. Autoría Propia.

Figura 22

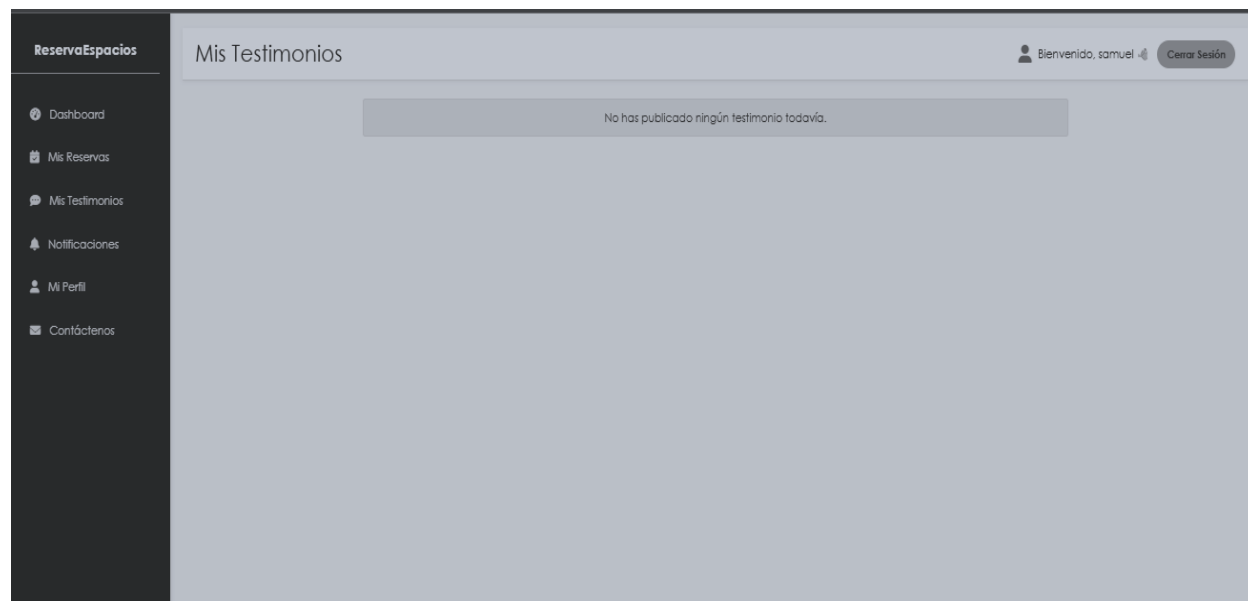
Vista de publicación de testimonio del usuario



Nota: Facilita la expresión de experiencias personales dentro del sistema de reservas. Autoría Propia.

Figura 23

Vista de testimonios del usuario



Nota: Permite consultar los testimonios enviados y su estado dentro del sistema. Autoría Propia.

Figura 24

Vista del panel administrativo del sistema de reservas



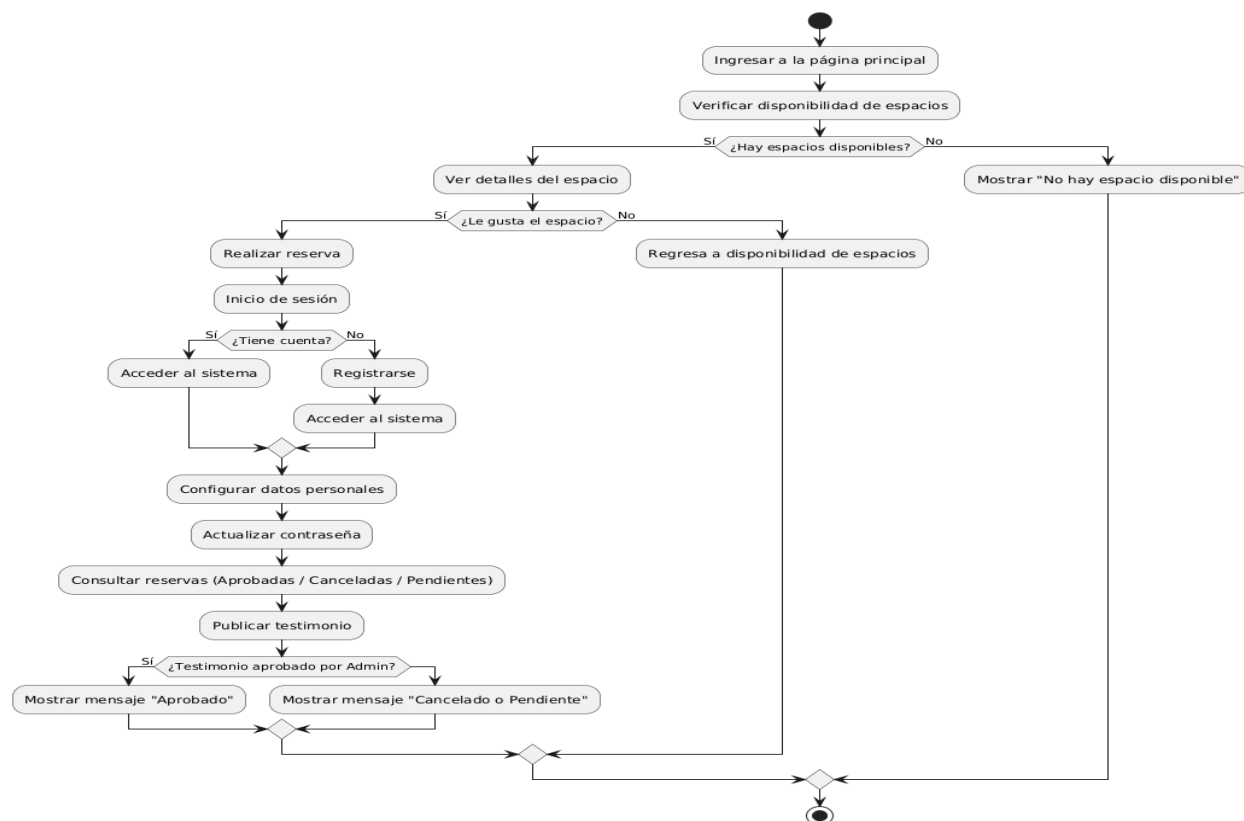
Nota: Permite visualizar indicadores generales y acceder a la gestión de usuarios, espacios, reservas y testimonios. Fuente: Autoría Propia.

Diagrama de actividades

En el siguiente diagrama de actividades *Diagrama 1*, se representa el flujo lógico de acciones que ejecutan determinadas clases dentro del sistema. Este modelo permite visualizar el comportamiento dinámico del proceso, desde su estado inicial hasta la finalización, incluyendo decisiones, bifurcaciones y sincronizaciones. Cada actividad refleja una operación significativa que contribuye al cumplimiento de un objetivo funcional, y su secuencia evidencia cómo interactúan los componentes del sistema en distintos escenarios. Este tipo de representación es útil para identificar puntos críticos, validar la coherencia del flujo y anticipar posibles ajustes en futuras versiones del sistema.

Diagrama 2

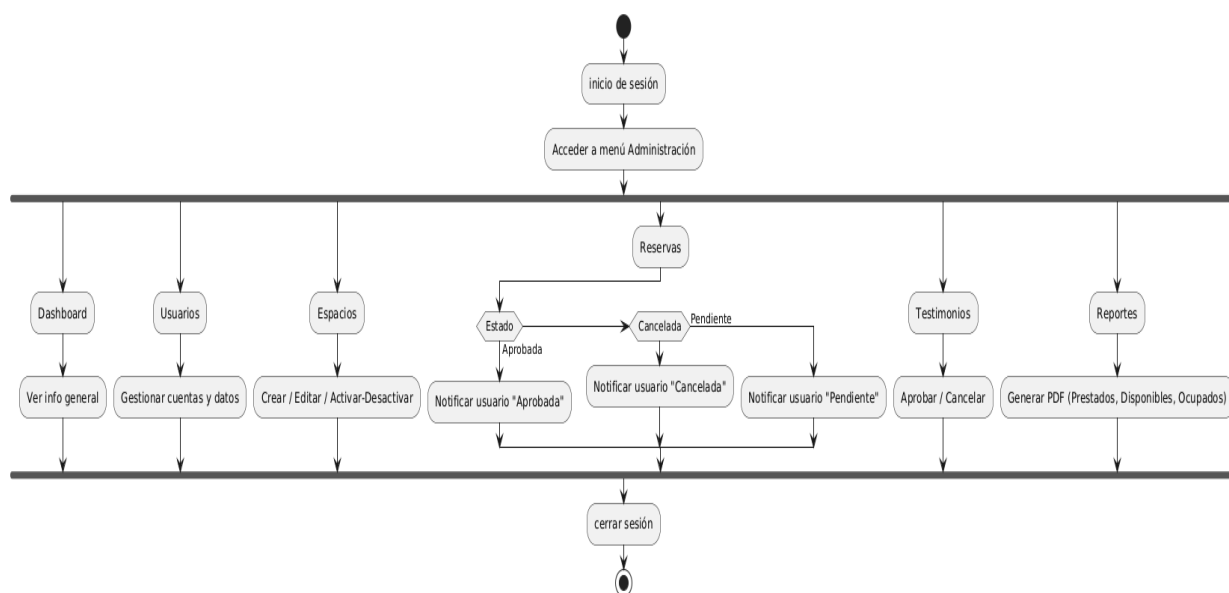
Diagrama de actividades del usuario en el sistema de reservas



Nota. El diagrama muestra las acciones que realiza el usuario desde el acceso a la página principal, la verificación de espacios y la reserva, hasta la consulta de sus datos y testimonios.
Fuente: Autoría Propia.

Diagrama 3

Diagrama de actividades del administrador en el sistema de reservas



Nota: El diagrama detalla el flujo de actividades del administrador, desde el inicio de sesión hasta el cierre de sesión, incluyendo la gestión de usuarios registrados, reservas solicitadas, testimonios publicados, la administración de espacios (crear, editar, activar o desactivar), la modificación de información del banner y la generación de reportes PDF sobre la disponibilidad y uso de los espacios. Autoría propia.

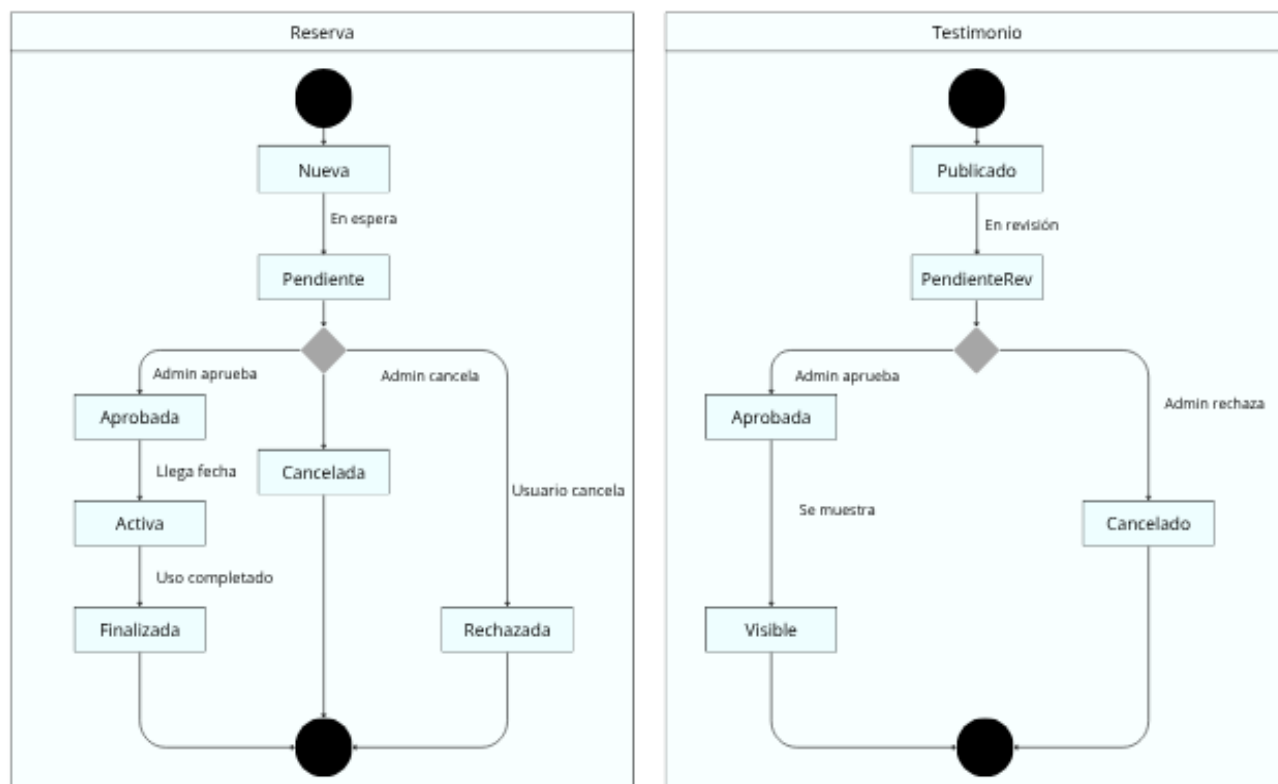
Diagrama de estados

Para comprender el funcionamiento del sistema de reservas, se diseñaron diagramas de estados UML, los cuales permiten representar gráficamente los posibles estados de cada componente y las transiciones que ocurren según las acciones del usuario o del administrador.

En los siguientes apartados se presentan los diagramas de estados correspondientes a los módulos principales del sistema: reservas, testimonios, administrador, usuario y espacios. Cada figura detalla de manera organizada cómo se gestionan los cambios de estado y las decisiones que influyen en el ciclo de vida de cada elemento.

Figura 25

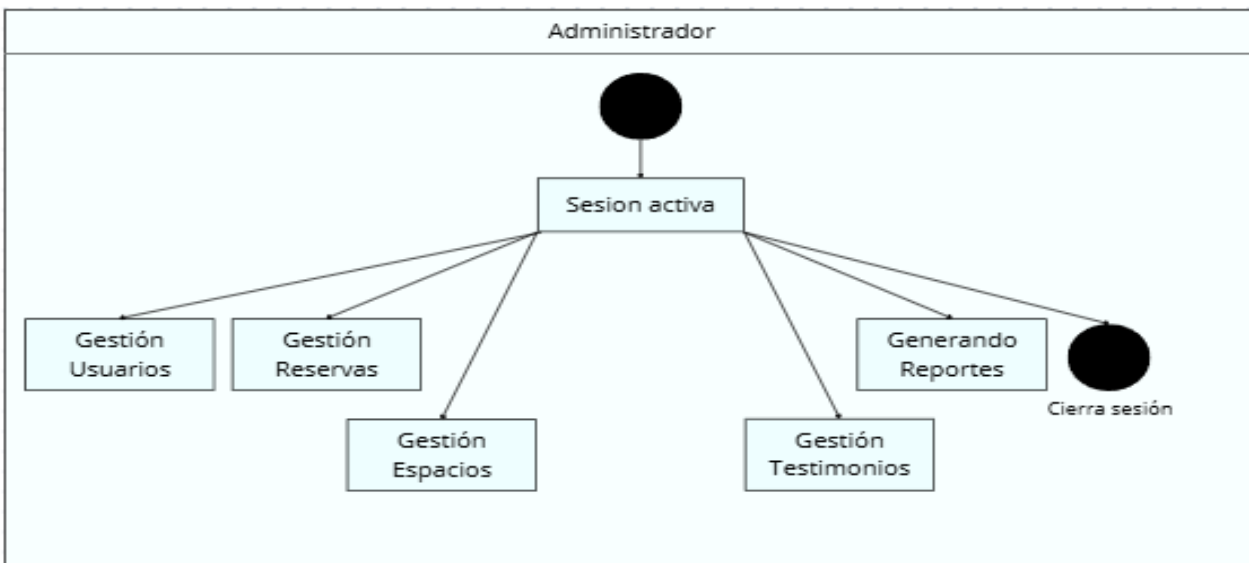
Diagrama de estados de reservas y testimonios



Nota. El diagrama representa, por un lado, los estados de una reserva en el sistema (nueva, pendiente, aprobada, cancelada, rechazada, activa y finalizada) y, por otro, los estados de un testimonio publicado por el usuario (pendiente, aprobado, cancelado y visible). Autoría propia.

Figura 26

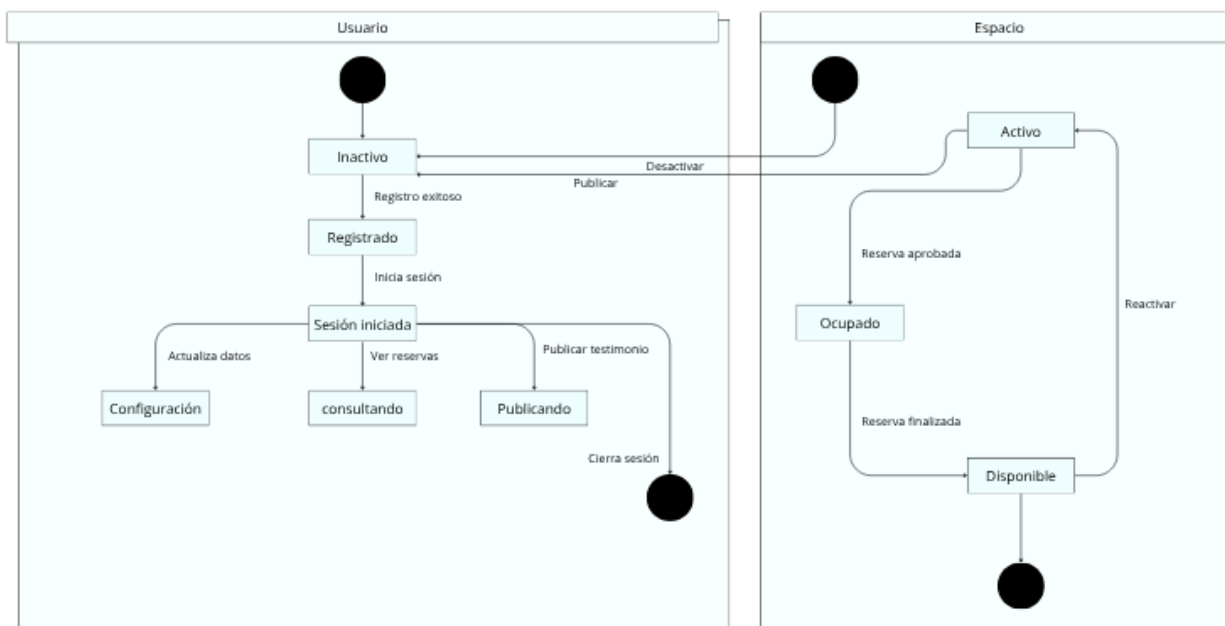
Diagrama de estados del módulo de administrador



Nota. Ilustra los estados y acciones principales del administrador en sesión activa, incluyendo la gestión de usuarios, reservas, espacios, testimonios y reportes. Autoría propia.

Figura 27

Diagrama de estados del módulo de usuario y espacios



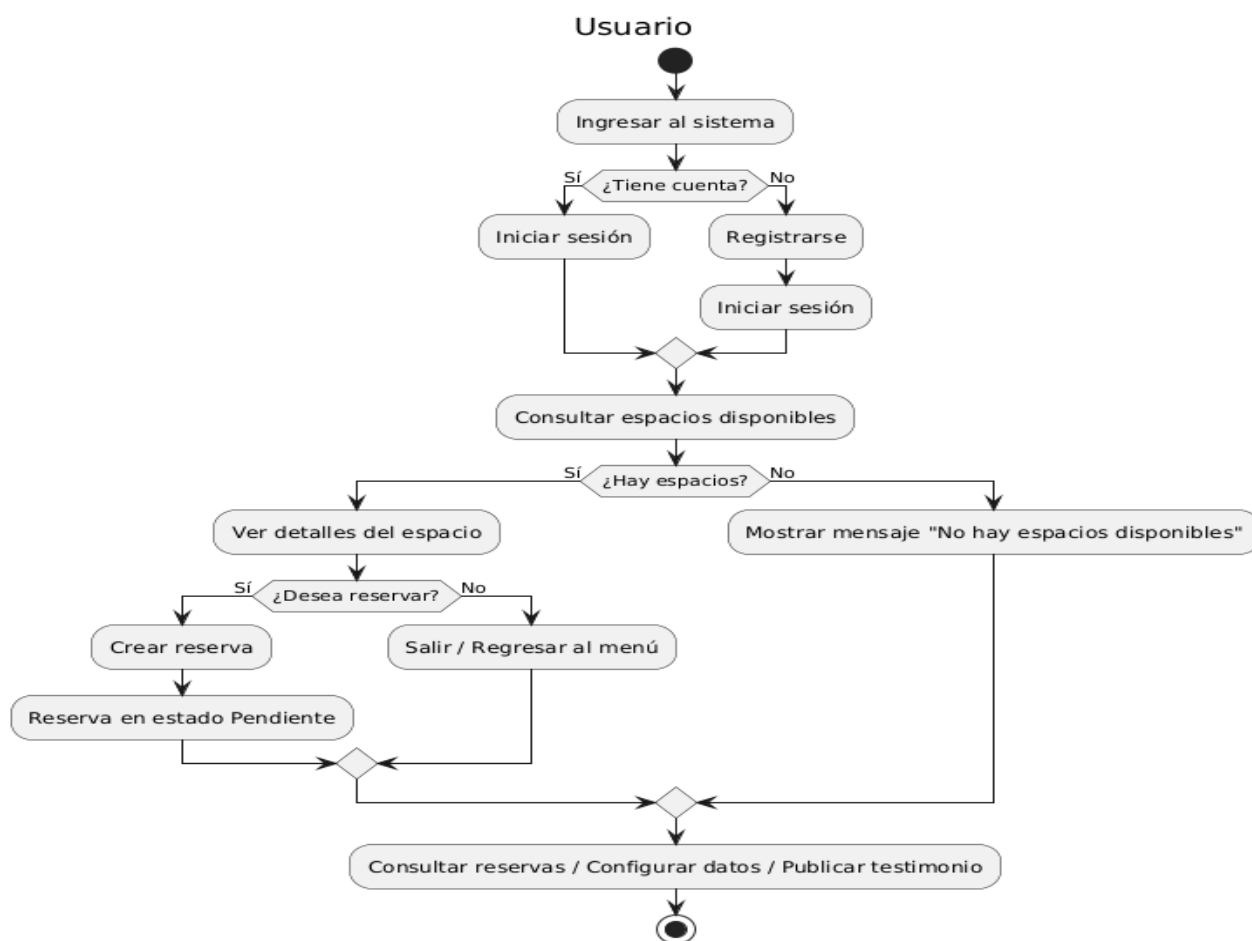
Nota. Se describen los estados del usuario en el sistema (registro, inicio y cierre de sesión, interacciones) y los del espacio (inactivo, activo, ocupado, disponible). Autoría propia.

Diagrama de flujo

Los diagramas de flujo permiten representar de manera ordenada los procesos del sistema de reservas, mostrando las actividades y decisiones que intervienen en la interacción del usuario y en la gestión realizada por el administrador. Estos diagramas facilitan la comprensión del funcionamiento interno del sistema y de las rutas que puede seguir cada acción.

Figura 28

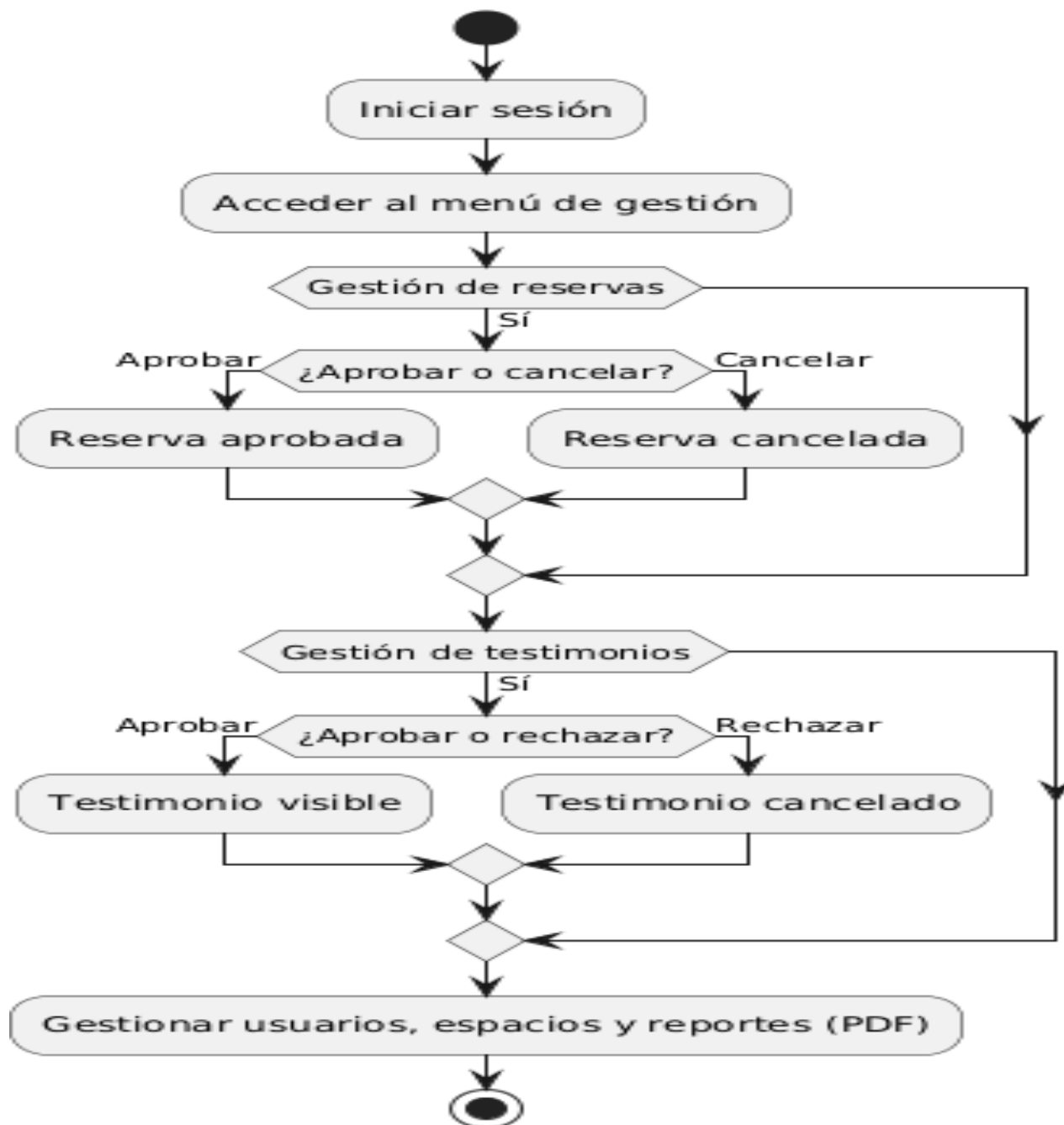
Diagrama de flujo del módulo de usuario



Nota. El diagrama ilustra el proceso que sigue el usuario desde el ingreso al sistema, considerando el registro o inicio de sesión, la consulta de espacios disponibles, la creación de reservas y la publicación de testimonios. Autoría propia.

Figura 29

Diagrama de flujo del módulo de administrador



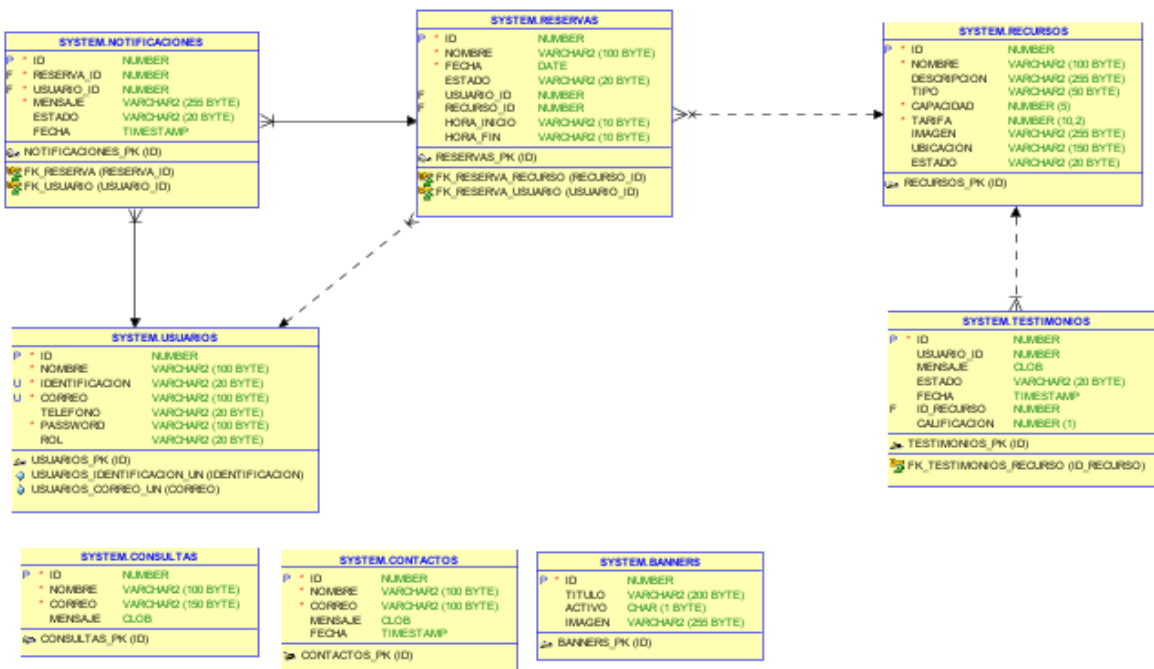
Nota. Se representan las acciones principales del administrador al iniciar sesión en el sistema, incluyendo la gestión de reservas, testimonios, usuarios, espacios y la generación de reportes. Autoría propia.

Modelo entidad relación

Dentro del desarrollo del sistema de reservas, el modelo entidad–relación cumple la función de representar cómo se estructuran y relacionan los datos en un nivel conceptual. A través de este modelo se identifican las entidades centrales (usuario, administrador, espacio, reserva y testimonio), sus atributos característicos (como nombre, identificación, teléfono, estado de la reserva, disponibilidad del espacio, entre otros) y las relaciones existentes entre ellas. De esta manera, se establece un esquema organizado que facilita la gestión de la información y constituye un insumo esencial para el diseño físico de la base de datos.

Figura 30

Modelo entidad–relación del sistema de reservas



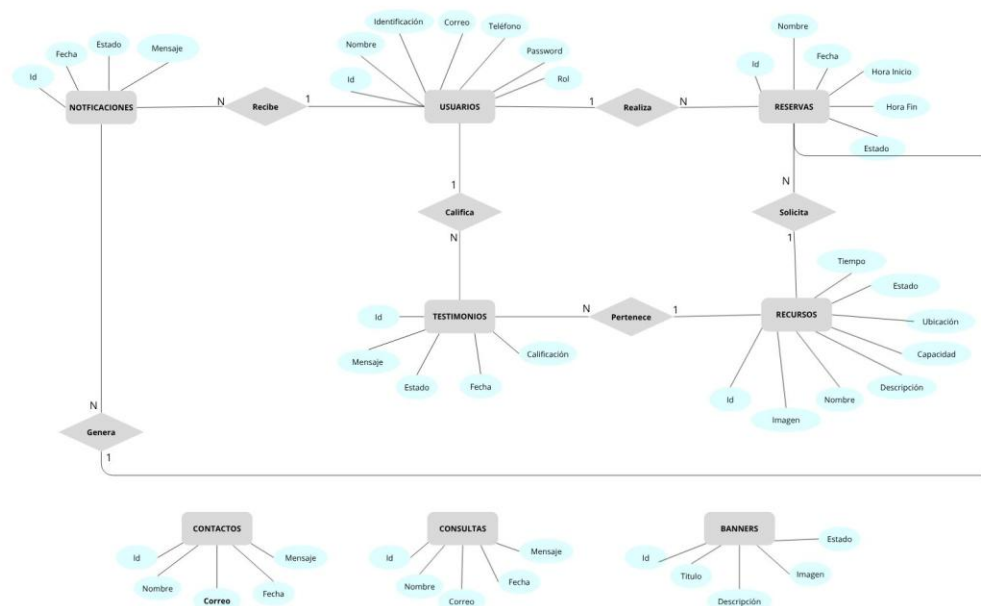
Nota. El diagrama representa las entidades principales del sistema (usuarios, administradores, reservas, espacios y testimonios), sus atributos más relevantes y las relaciones que permiten organizar la información de manera estructurada. Autoría propia.

Diseño conceptual base de datos

El diseño conceptual de la base de datos constituye una etapa clave en el desarrollo del sistema de reservas, ya que permite transformar los requerimientos funcionales en una estructura lógica organizada. En este diseño se definen las entidades principales, sus atributos y las relaciones entre ellas, lo que asegura la correcta representación de la información necesaria para la gestión de usuarios, reservas, espacios, testimonios y procesos administrativos. De esta manera, el modelo conceptual sirve como base para la posterior creación del modelo lógico y físico de la base de datos, garantizando consistencia, integridad y eficiencia en el manejo de los datos.

Figura 31

Diseño conceptual de la base de datos del sistema de reservas



Nota. El diseño muestra la estructura conceptual de la base de datos, incluyendo las entidades identificadas, sus atributos principales y las relaciones que garantizan la integridad de la información dentro del sistema de reservas. Autoría propia.

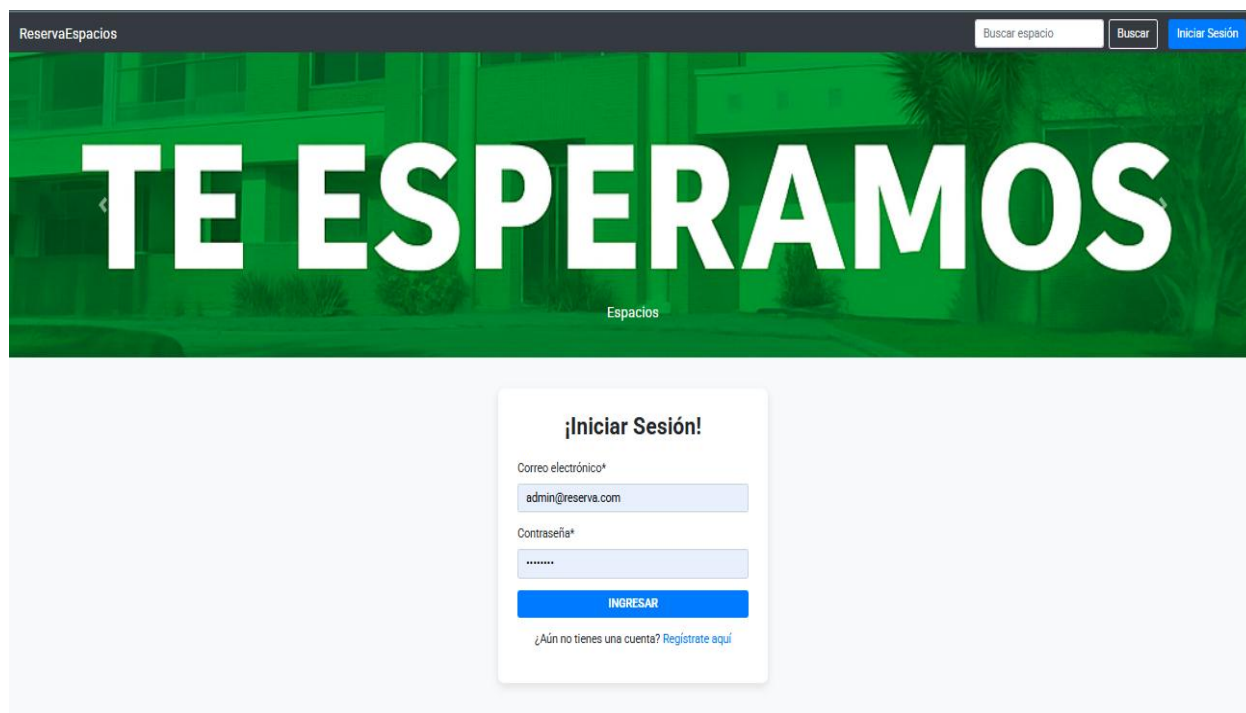
Fase 3 Desarrollo (Construcción del prototipo)

La fase de desarrollo constituye una de las etapas más relevantes del proyecto *Software para el Control de Alquiler de los Recursos Disponibles del Campus Universitario*, puesto que en ella se materializan los requerimientos establecidos durante el análisis y el diseño. En este proceso se implementaron los módulos funcionales que permiten gestionar los recursos del campus: registro, consulta y administración de alquileres.

Además, se construyó la base de datos encargada de centralizar y proteger la información, garantizando integridad, disponibilidad y acceso oportuno a los datos relacionados con los procesos de reserva. Paralelamente, se diseñó una interfaz gráfica intuitiva orientada a mejorar la experiencia del usuario, facilitando la interacción tanto del personal administrativo como de los solicitantes de espacios.

Gracias a ello, se logró desarrollar una solución tecnológica eficiente, alineada con las necesidades detectadas y capaz de optimizar el control, la asignación y el seguimiento de los recursos disponibles dentro de la universidad.

. En esta pantalla el administrador debe ingresar sus credenciales para acceder al sistema. El formulario solicita un correo electrónico y contraseña previamente registrados en la plataforma. Una vez validados los datos, el usuario es dirigido al módulo administrativo donde podrá gestionar recursos, solicitudes y usuarios. En caso de no contar con una cuenta, se dispone de un enlace para registro.

Figura 32*Inicio de sesión del administrador*

ReservaEspacios

Buscar espacio

Buscar

Iniciar Sesión

TE ESPERAMOS

Espacios

¡Iniciar Sesión!

Correo electrónico*

admin@reserva.com

Contraseña*

.....

INGRESAR

[¿Aún no tienes una cuenta? Regístrate aquí](#)

Nota: Formulario principal para el acceso seguro al sistema. Autoría propia.

Al ingresar al sistema, el administrador accede al Panel de Administración, donde puede visualizar un resumen general de la plataforma. En la parte superior se presentan indicadores con el total de reservas, espacios registrados, testimonios y usuarios activos.

Desde el menú lateral se puede navegar a las diferentes funciones administrativas como gestión de usuarios, recursos, reservas, testimonios, consultas, reportes y configuración del sistema. Además, se incluye un filtro para consultar reservas según su estado, así como acceso directo a la lista completa de solicitudes registradas en la plataforma.

Figura 33

Vista del panel de administrador

Panel de Administrador

Reservas: 3, Espacios: 2, Testimonios: 0, Clientes: 4

Filtrar por estado: -- Todos --

Listado de Reservas

ID	Usuario	Recurso	Fecha	Hora	Estado	Acciones
123	samuel	auditorio	2025-10-29	12:22 - 14:28	PENDIENTE	✓ Aprobar, ✖ Puntar, 🗑 Finalizar
122	samuel	auditorio	2025-10-28	09:01 - 10:07	PENDIENTE	✓ Aprobar, ✖ Puntar, 🗑 Finalizar
121	samuel	sala de computo	2025-10-27	08:00 - 10:00	PENDIENTE	✓ Aprobar, ✖ Puntar, 🗑 Finalizar

Nota. Se observa el listado de reservas con sus estados y las acciones disponibles para su administración. Autoría propia.

Figura 34

Gestión de usuarios por parte del administrador.

Lista de Usuarios

Buscar por nombre o documento

ID	NOMBRE	IDENTIFICACIÓN	CORREO	TÉLEFONO	ROL	ACCIONES
1	marco	1000663315	admin@reserva.com	3000000000	ADMIN	✎ Editar, 🗑 Eliminar
2	samuel	12345667	samuel@gmail.com	34542324	ADMIN	✎ Editar, 🗑 Eliminar
21	pepe	1111	pepe@gmail.com	3123285895	USER	✎ Editar, 🗑 Eliminar
23	samuel	243432	sam@gmail.com	32435435	USER	✎ Editar, 🗑 Eliminar

Nota. Se muestra el listado de usuarios registrados y las acciones disponibles para su administración. Autoría propia.

En esta sección, el administrador puede actualizar su información personal y de acceso al sistema. El formulario permite modificar datos como nombre, correo electrónico, teléfono y contraseña. De este modo, se garantiza que cada administrador mantenga su perfil seguro y actualizado dentro de la plataforma.

Figura 35

Edición del perfil del administrador.

The screenshot displays the 'SistemaReserva' interface. At the top, there is a navigation bar with 'Reservas', 'Recursos', 'Usuarios', and 'Reportes'. The user 'marco' is logged in, with a 'Cerrar Sesión' button. The left sidebar is titled 'Administración' and contains several menu items: 'Usuarios', 'Configuración' (highlighted), 'Banner', 'Gestionar Reservas', 'Gestionar Testimonios', 'Gestionar Consultas', 'Nuevo Espacio', 'Gestionar Espacios', and 'Reportes'. The main content area shows a modal window titled 'Editar Perfil Administrador'. This modal contains the following fields: 'Nombre completo' with the value 'marco', 'Correo electrónico' with 'admin@reserva.com', 'Teléfono' with '3000000000', and 'Nueva contraseña' which is masked with dots. At the bottom of the modal are two buttons: 'Guardar cambios' and 'Cancelar'.

Nota. Se muestra el formulario destinado a modificar la información personal y credenciales del administrador. Autoría propia.

Esta sección permite al administrador administrar los espacios disponibles para alquiler dentro del campus universitario. Se presenta un listado con información clave del recurso como nombre, ubicación, capacidad, tarifa y estado de disponibilidad. Además, se ofrecen acciones para editar o eliminar cada recurso, así como la opción de registrar nuevos espacios, asegurando una gestión actualizada del inventario.

Figura 36

Gestión de recursos del sistema.

ID	IMAGEN	NOMBRE	TIPO	UBICACIÓN	CAPACIDAD	TARIFA	ESTADO	DISPONIBLE	ACCIONES
61		sala de computo	SALON	universidad de Cundinamarca	50	\$ 100000.0	DISPONIBLE		Editar Eliminar
62		auditorio	SALON	Universidad de cundinamarca	120	\$ 50000.0	DISPONIBLE		Editar Eliminar

Nota. Se observa el listado de espacios registrados junto con sus características y acciones disponibles. Autoría propia.

Figura 37

Formulario de registro de un nuevo recurso.

Registrar Nuevo Recurso

Nombre
Ej: Laboratorio A

Tipo: Seleccione Capacidad: Ej: 30 Estado: Disponible

Ubicación: Ej: Edif. B, Piso 1 Tarifa (COP): Ej: 50000

Descripción: Breve descripción Imagen: [Elegir archivo](#) No se ha seleccionado ningún archivo

Disponible para reservas

[Guardar](#) [Cancelar](#)

Nota. Se visualizan los campos requeridos para ingresar la información del recurso y su disponibilidad. Autoría propia.

El sistema permite a los usuarios consultar la disponibilidad de los espacios del campus seleccionando fecha y horarios. Una vez verificada la disponibilidad, se muestran los recursos habilitados para reserva junto con su información relevante: ubicación, capacidad, tarifa y estado. Desde esta vista, el usuario puede explorar las opciones y avanzar.

Figura 38

Consulta de disponibilidad de espacios.

The screenshot shows the 'ReservaEspacios' website interface. At the top, there is a navigation bar with the text 'ReservaEspacios' on the left, a search input field labeled 'Buscar espacio', a 'Buscar' button, and an 'Iniciar Sesión' button. Below the navigation bar is a large green banner with the text 'TE ESPERAMOS' in white. Underneath the banner is a teal-colored search form with four fields: 'Fecha Entrada' (with a date input 'dd/mm/aaaa' and a calendar icon), 'Hora de ingreso' (with a time input '---:--' and a clock icon), 'Hora de salida' (with a time input '---:--' and a clock icon), and a 'Verificar disponibilidad' button labeled 'Ver disponibilidad'.

Nota. Se muestra el formulario para seleccionar fecha y horarios con el fin de validar disponibilidad antes de reservar. Autoría propia.

Figura 39

Catálogo de espacios disponibles para reserva.

The screenshot displays a catalog of available spaces. The header reads 'Espacios que impulsan tus ideas' with a 'Disponibles' badge. Below the header, two space cards are shown, both from 'Universidad de Cundinamarca' and marked as 'Disponibles'.
 The first card is for a 'sala de computo' (computer room), described as a room equipped with the latest technology for academic activities, with access to specialized software and navigation. It has a capacity of 50 people and a price of \$100,000.00 COP. It has no ratings yet and a 'Reservar Ahora' button.
 The second card is for an 'auditorio' (auditorium), described as a large space designed for conferences, presentations, seminars, and institutional events. It has a capacity of 120 people and a price of \$50,000.00 COP. It also has no ratings yet and a 'Reservar Ahora' button.

Nota. Se muestran los recursos publicados con sus características principales y acceso al proceso de reserva. Autoría propia.

Esta sección permite a los usuarios conocer las experiencias y opiniones de quienes ya han realizado reservas dentro del campus. Aquí se muestran únicamente los testimonios que han sido aprobados por el administrador, garantizando la veracidad y pertinencia del contenido.

Figura 40

Sección de testimonios de usuarios.



Nota. Se visualiza el apartado destinado a mostrar testimonios aprobados sobre los espacios reservados en la universidad. Autoría propia.

Esta interfaz permite que los usuarios ingresen al sistema utilizando su correo electrónico y contraseña, con el fin de acceder a las funcionalidades de reserva de espacios dentro de la universidad. Además, se incluye un enlace para registrarse en caso de no contar aún con una cuenta, lo que facilita su incorporación al sistema.

Figura 41

Inicio de sesión para usuarios.

ReservaEspacios

Espacios que impulsan tus ideas

Prestamo de espacios

¡Iniciar Sesión!

Correo electrónico*

Contraseña*

[¿Aún no tienes una cuenta? Regístrate aquí](#)

Nota. Se visualiza el formulario de autenticación dirigido a los usuarios que desean acceder para realizar reservas. Autoría propia.

Figura 42

Dashboard del usuario.

ReservaEspacios

¡Bienvenido!

Bienvenido, samuel

Tus espacios
Consulta los espacios asignados o disponibles.

Tus testimonios
Revisa o edita tus testimonios compartidos.

Realiza tu reserva
Reserva un espacio en pocos clics.

Mis Reservas

ID	RECURSO	FECHA	HORA	ESTADO
122	62	2025-10-28	09:01 - 10:07	Pendiente
121	61	2025-10-27	08:00 - 10:00	Pendiente
123	62	2025-10-29	12:22 - 14:28	Pendiente

Notificaciones

No tienes notificaciones

Nota. Se muestran accesos principales y el listado de reservas recientes del usuario dentro del sistema. Autoría propia.

Figura 43

Detalle de la información de una reserva del usuario.

The screenshot displays the 'ReservaEspacios' application interface. On the left is a dark green sidebar with navigation options: Dashboard, Mis Reservas, Mis Testimonios, Notificaciones, Mi Perfil, and Contáctenos. The main content area has a white header with '¡Bienvenido!' and a user profile 'Bienvenido, samuel' with a 'Cerrar Sesión' button. Below this is a modal window titled 'INFORMACIÓN DE LA RESERVA' with a green header and 'ACCIONES' on the right. The reservation details are as follows:

Icono	Detalle
	auditorio
	Ubicación: Universidad de cundinamarca
	Descripción: Espacio amplio diseñado para la realización de conferencias, presentaciones, seminarios y eventos institucionales.
	Capacidad: 120 personas
	Tipo: SALON
	Tarifa: \$50000.0
	Fecha: 2025-10-28
	Horario: 09:01 - 10:07
	Estado: Pendiente

At the bottom of the modal, there is a pagination indicator 'Página 1 de 3' and a '>>' button. A red 'Cancelar' button is located in the top right corner of the modal.

Nota. Se muestra la descripción del recurso reservado junto con la opción de cancelar la solicitud si es necesario. Autoría propia.

Figura 44

Edición del perfil del usuario.

The screenshot shows the 'Editar Perfil' page in the 'ReservaEspacios' application. The sidebar and header are consistent with the previous figure. The main content area features a white form titled 'Editar Perfil' with the following fields:

- Nombre completo:** Input field containing 'samuel'.
- Correo electrónico:** Input field containing 'sam@gmail.com'.
- Teléfono:** Input field containing '32435435'.
- Nueva contraseña:** Input field with masked characters '.....'.

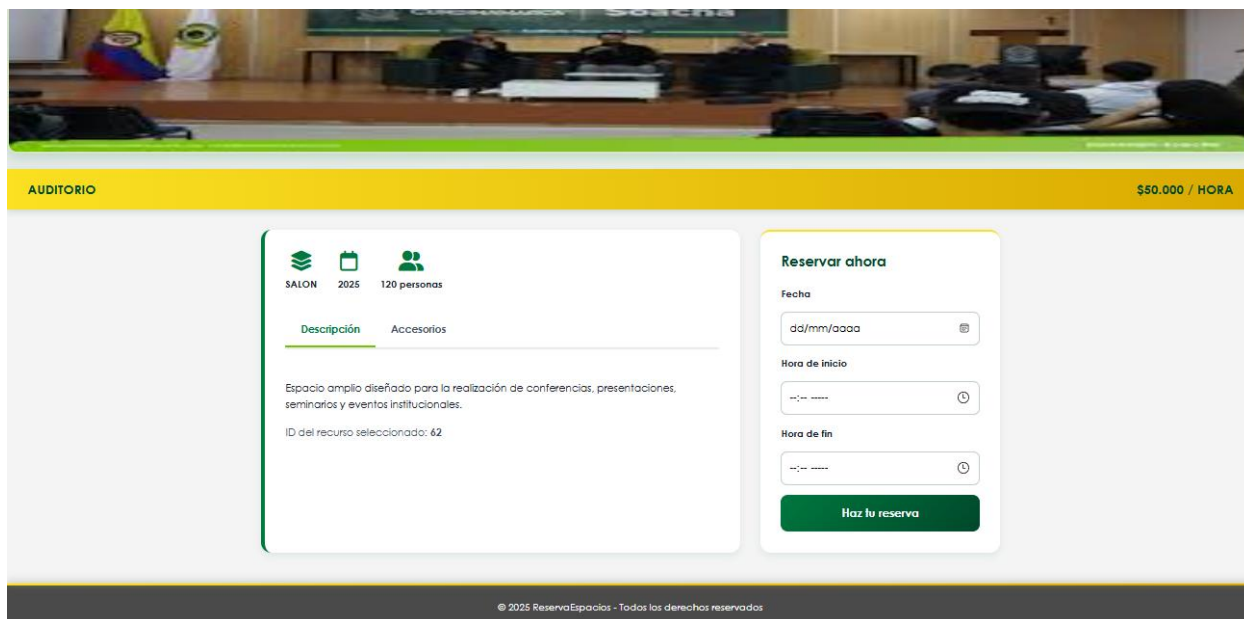
At the bottom of the form are two buttons: a green 'Guardar Cambios' button and a grey 'Cancelar' button.

Nota. Se visualiza el formulario destinado a la actualización de la información personal del usuario. Autoría propia.

En esta interfaz, el usuario puede visualizar la información detallada del espacio seleccionado, incluyendo tipo, capacidad, tarifa y descripción del uso del recurso dentro de la universidad. Al mismo tiempo, se habilita el formulario de solicitud de reserva, donde el usuario selecciona la fecha y el horario deseado antes de registrar su solicitud en el sistema.

Figura 45

Detalle del recurso y formulario de reserva.



Nota. Se muestra la información del espacio y el formulario que permite al usuario registrar una reserva según disponibilidad. Autoría propia.

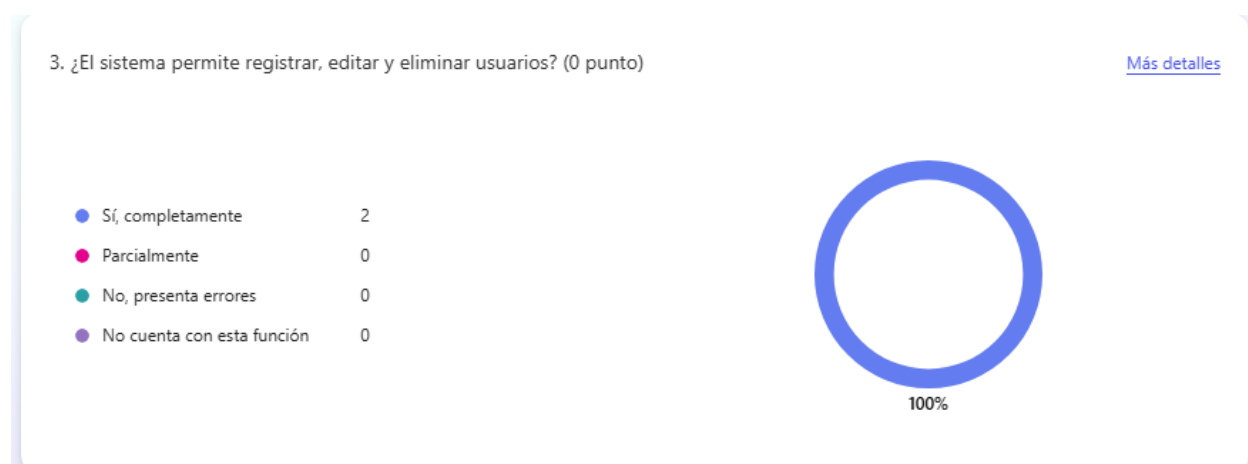
Capítulo 9. Resultados

Para evaluar el desempeño del sistema, se aplicó una encuesta al personal administrativo encargado de la gestión del préstamo de espacios en el campus universitario. A continuación, se presenta el análisis de los aspectos evaluados:

En la **Figura 46**, el 100% de los encuestados afirmó que el sistema permite registrar, editar y eliminar usuarios de manera completa y funcional. Esto evidencia que el módulo de gestión de usuarios cumple con los requerimientos definidos y aporta control total sobre el acceso al sistema. Ver resultados en la carpeta **Anexos** ver **Anexo a**. Encuesta instrumento diagnóstico

Figura 46

Evaluación del módulo de gestión de usuarios.

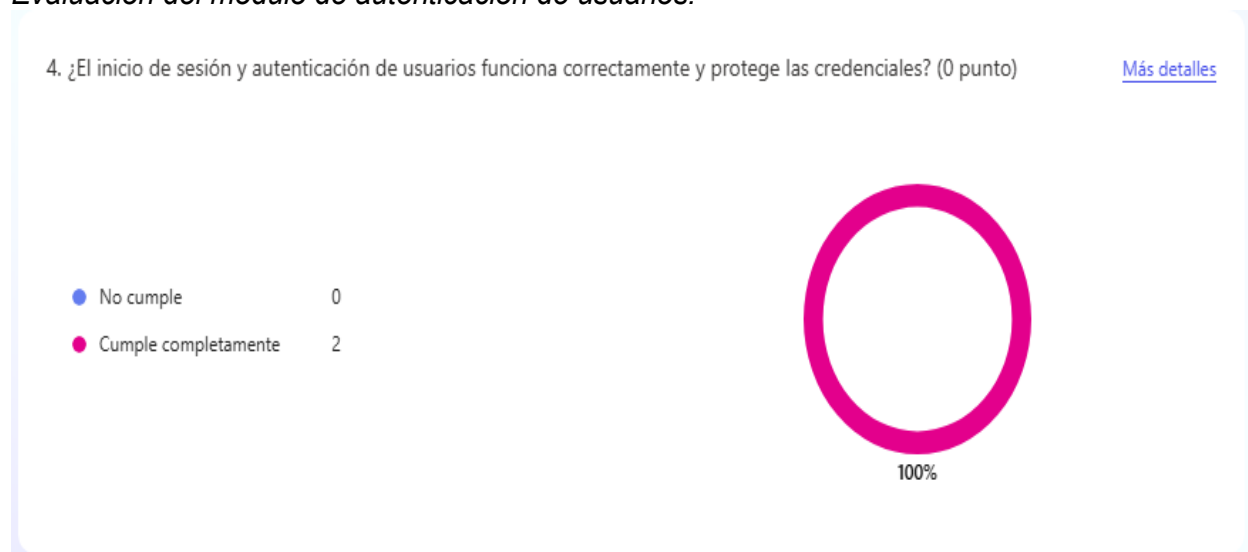


Nota. Resultados de la encuesta sobre la funcionalidad de registro, edición y eliminación de usuarios. Autoría propia.

En la **Figura 47**, el 100% de los participantes indicó que el sistema realiza correctamente el proceso de inicio de sesión, protegiendo la información de acceso de los usuarios.

Figura 47

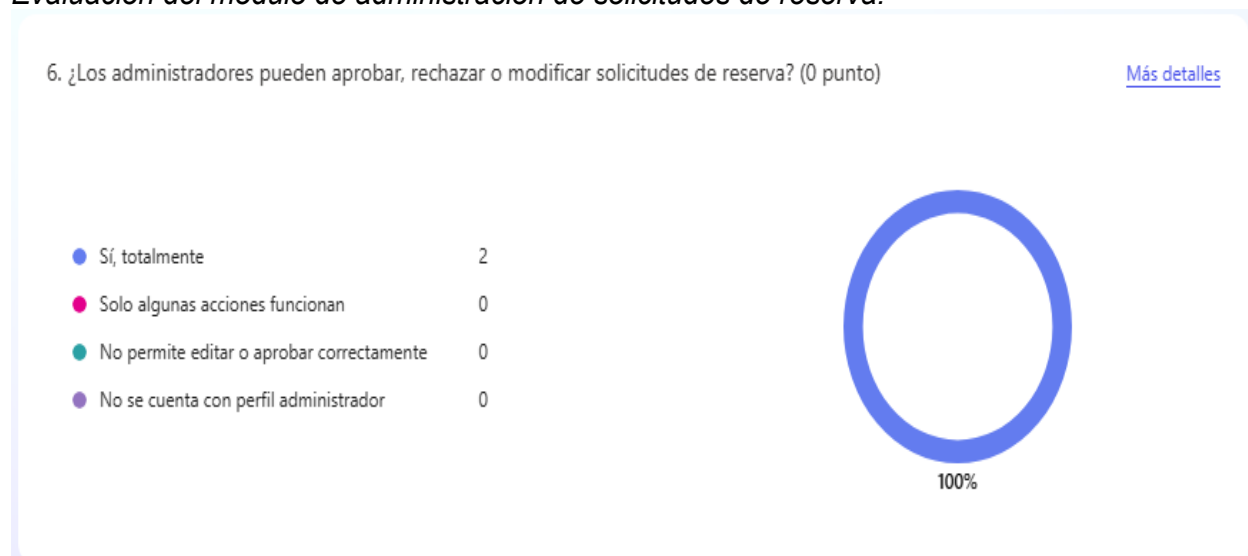
Evaluación del módulo de autenticación de usuarios.



Nota. Resultados de la encuesta sobre el funcionamiento del inicio de sesión y protección de credenciales. Autoría propia

Figura 48

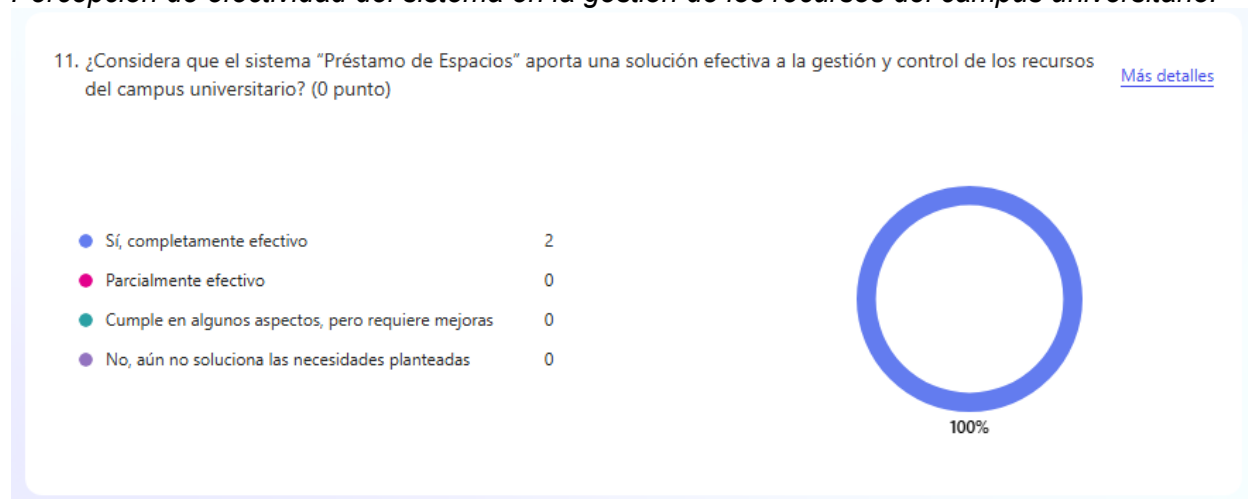
Evaluación del módulo de administración de solicitudes de reserva.



Nota. Resultados de la encuesta aplicada al personal encargado de la gestión, donde el 100% confirmó que el administrador puede aprobar, rechazar o modificar las solicitudes de reserva sin inconvenientes. Autoría propia.

Figura 49

Percepción de efectividad del sistema en la gestión de los recursos del campus universitario.



Nota. Resultados de la encuesta aplicada al personal encargado de la gestión, donde el 100% considera que el sistema es completamente efectivo para el control y administración de los espacios. Autoría propia.

Capítulo 10. Discusión

La implementación de soluciones tecnológicas dentro de la Universidad de Cundinamarca permite optimizar procesos que actualmente se gestionan de forma manual y que pueden generar retrasos e inconsistencias en la asignación de espacios. Con el desarrollo del Software para el Control de Alquiler de los Recursos Disponibles del Campus Universitario, se evidencia cómo la tecnología contribuye a agilizar y mejorar la calidad del servicio ofrecido, garantizando un proceso más organizado, eficiente y transparente.

El sistema centraliza la información relacionada con reservas, usuarios y disponibilidad en tiempo real, lo cual facilita la labor del personal administrativo encargado de gestionar los espacios. A su vez, ofrece a los estudiantes y docentes una herramienta práctica que les permite acceder a los recursos institucionales de manera autónoma y sin depender de procedimientos manuales.

Adicionalmente, la interfaz intuitiva del software y su facilidad de uso mejoran la experiencia de quienes interactúan con la plataforma, permitiendo que la gestión de los espacios universitarios sea más accesible, controlada y moderna. El impacto positivo reflejado en los resultados de la encuesta confirma la pertinencia de este proyecto como una solución tecnológica que fortalece la organización interna y la correcta administración de los recursos dentro de la universidad.

Capítulo 11. Conclusiones

- El desarrollo del *Software para el Control de Alquiler de los Recursos Disponibles del Campus Universitario* permitió establecer una solución tecnológica capaz de mejorar significativamente la gestión actual del préstamo de espacios dentro de la Universidad de Cundinamarca, Extensión Soacha, respondiendo a una necesidad real identificada en la universidad.
- El personal administrativo que evaluó el sistema manifestó una percepción positiva sobre su utilidad, facilidad de uso y aporte al fortalecimiento del control institucional, destacando que la plataforma facilita la organización y optimiza el manejo de los recursos disponibles en el campus.
- Finalmente, se concluye que este proyecto constituye un aporte importante para la modernización de los procesos institucionales, contribuye a la calidad del servicio prestado y sienta las bases para futuras mejoras orientadas a la transformación digital dentro de la universidad.

Capítulo 12. Recomendaciones

Para continuar fortaleciendo y expandiendo el alcance del *Software para el Control de Alquiler de los Recursos Disponibles del Campus Universitario*, se recomienda integrar nuevas funcionalidades que permitan mejorar todavía más la experiencia de los usuarios y la eficiencia en la gestión administrativa.

Asimismo, se recomienda desarrollar una versión móvil del software compatible con diferentes sistemas operativos, como Android e iOS, permitiendo a los usuarios realizar solicitudes y administrar sus reservas desde cualquier dispositivo y lugar. Esto aumentaría significativamente la accesibilidad y cobertura de la herramienta.

Finalmente, se plantea la posibilidad de integrar módulos complementarios como estadísticas avanzadas, mapas interactivos del campus para ubicar los espacios, y mecanismos de firma digital para formalizar las reservas. Estas mejoras futuras permitirán una evolución constante del sistema y una adaptación a las nuevas necesidades académicas y administrativas de la universidad.

Capítulo 13. Bibliografía

Abuchar Porras, R. (2023). *Metodologías ágiles para el desarrollo de software*. Editorial académica (si aplica).

Reyes, L. (2017). *Solución de software para apoyar el proceso de gestión de espacios físicos de la Universidad Distrital*. Universidad Distrital Francisco José de Caldas.

González, J. (2019). *Gestión de espacios educativos asistida por software*.

Universidad de Bogotá Jorge Tadeo Lozano. (2015). *Sistema de gestión de espacios académicos*.

Universidad de los Andes. (s.f.). *Sistema de préstamo de espacios y equipos de la Facultad de Artes y Humanidades*. <https://facartes.uniandes.edu.co> (o URL correspondiente)

Pontificia Universidad Javeriana. (s.f.-a). *Portal de reservas de espacios PUJ*. <https://www.javeriana.edu.co>

Ministerio de Educación Nacional. (1992). *Ley 30 de 1992*. Diario Oficial de Colombia.

Congreso de Colombia. (2012). *Ley 1581 de 2012 – Protección de datos personales*. Diario Oficial de Colombia.

Congreso de Colombia. (2000). *Ley 603 de 2000 – Licenciamiento de software*. Diario Oficial de Colombia.

Montaño, P. (2018). *Optimización en la gestión de recursos físicos universitarios*.

Turban, E., Pollard, C., & Wood, G. (2018). *Sistemas de información en las organizaciones*. McGraw-Hill.

Pressman, R. S., & Maxim, B. R. (2020). *Ingeniería de software: Un enfoque práctico*. McGraw-Hill.

Laudon, K., & Laudon, J. (2016). *Sistemas de información gerencial*. Pearson.

Chiavenato, I. (2017). *Introducción a la teoría general de la administración*. McGraw-Hill.

Elmasri, R., & Navathe, S. (2017). *Sistemas de gestión de bases de datos*. Pearson.

Sharma, A., & Kumar, D. (2020). *Tecnologías de apoyo al desarrollo de aplicaciones web*.

Sommerville, I. (2011). *Ingeniería del software*. Pearson.

Piattini, M., & Calero, C. (2002). *Análisis y diseño orientado a objetos con UML*. RA-MA Editorial.

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing* (Special Publication 800-145). National Institute of Standards and Technology.

Rodríguez, D. (2019). *Software de apoyo administrativo universitario*.

Ruiz-Vanoye, J. A., et al. (2017). *Metodologías de desarrollo de software*. Alfaomega.

TutorialsPoint. (s.f.). *CASE Tools overview*.

https://www.tutorialspoint.com/es/software_engineering/case_tools_overview.htm

Magazine Pro. (2023). *Diagramas de componentes y actividades UML*.

<https://diagramasuml.com/componentes/>

Lucidchart. (2023). *Diagrama de despliegue UML*.

<https://www.lucidchart.com/pages/es/tutorial-de-diagramas-de-despliegue>

Lucidchart. (2023). *Diagrama de secuencia UML*.

<https://www.lucidchart.com/pages/es/diagrama-de-secuencia>

Corporación Ambiental Caminando el Territorio. (2014). *Relieve del municipio de Soacha*. (Mapa)

Universidad de Cundinamarca. (2023). *Extensión Soacha. Información institucional*.

Capítulo 14. Anexos

Anexo a. Encuesta instrumento diagnóstico

Anexo b. Acuerdo No. 000010 – Universidad de Cundinamarca.

Anexo c. Manual Técnico

Anexo d. Manual de usuario