

# Software De Préstamo De Audiovisuales Y Salas De Informática

## Audiovisual And Computer Room Loan Software

Mayron Santiago Serrano Hernández, Duvan Leandro Pedraza González, Haykel Steller Martínez Carvajal

Artículo recibido en mes 06 de año 2025; artículo aceptado en mes 06 de año 2025

Serrano Hernández, M. S, Pedraza Gonzalez, D. L, Martinez Carvajal, H. S (2025) Proyecto integrador [Integrados]. Volumen 1

### Resumen

Este artículo presenta el desarrollo de un proyecto integrador que consolida las competencias adquiridas en el programa de Ingeniería de Sistemas, enfocándose en la conceptualización, diseño, implementación y evaluación de un software innovador para la gestión de préstamos de salas y equipos audiovisuales. El sistema ofrece herramientas para la administración integral préstamos de salas y equipos, devoluciones, soporte, sanciones, incorporando funcionalidades clave como la gestión de roles de usuario (administrador y usuario).

Durante el desarrollo del proyecto, se emplearon metodologías fundamentales de la ingeniería de software. En particular, se adoptó la metodología en cascada, conocida por su enfoque secuencial y lineal, la cual garantizó una planificación estructurada que permitió cumplir con los requisitos del cliente y las especificaciones definidas desde las fases iniciales. Este enfoque fue complementado con un diseño técnico robusto que incluye diagramas UML, modelado relacional de bases de datos y principios de programación SOLID, asegurando que el sistema sea escalable, mantenible y adaptable a futuras necesidades.

El impacto del proyecto es significativo en el ámbito de la prestación de equipos y salas en el ámbito estudiantil, al optimizar procesos críticos como la gestión de sala o quipos, la gestión de devoluciones y sanciones. Gracias a su diseño intuitivo y su funcionalidad avanzada, el software permite a los usuarios, tanto administradores como usuarios, gestionar información clave de manera ágil, segura y confiable, promoviendo la eficiencia operativa.

**Palabras clave:** Software, SOLID, Base De Datos, Diseño

### Abstract

This article presents the development of an integrative project that consolidates the competencies acquired in the Systems Engineering program, focusing on the conceptualization, design, implementation and evaluation of an innovative software for the management of rooms and audiovisual equipment loans. The system offers tools for the integral administration of room and equipment loans, returns, support, sanctions, incorporating key functionalities such as the management of user roles (administrator and user).

During the development of the project, fundamental software engineering methodologies were used. In particular, the waterfall methodology was adopted, known for its sequential and linear approach, which ensured a structured planning that allowed meeting the client's requirements and the specifications defined from the initial phases. This approach was complemented by a robust technical design that includes UML diagrams, relational database modeling and SOLID programming principles, ensuring that the system is scalable, maintainable and adaptable to future needs.

The impact of the project is significant in the area of equipment and room provision in the student environment, by optimizing critical processes such as room or equipment management, returns management and sanctions. Thanks to its intuitive design and advanced functionality, the software allows users, both administrators and users, to manage key information in an agile, secure and reliable way, promoting operational efficiency.

Translated with DeepL.com (free version)

**Keywords:** Software, SOLID, Database, Design

### Introducción

La gestión eficiente del préstamo de recursos tecnológicos, como equipos audiovisuales y salas de informática, representa un reto clave para las

instituciones universitarias. Estas operaciones requieren una organización rigurosa para asegurar el acceso equitativo, el uso responsable y el mantenimiento adecuado de los recursos. Sin embargo, la falta de herramientas tecnológicas especializadas dificulta la coordinación entre usuarios, personal encargado y sistemas de control, lo que puede generar retrasos, conflictos en la programación y pérdidas materiales. Contar con un sistema de software que permita automatizar y controlar estas operaciones es esencial para optimizar el uso de recursos, mejorar la experiencia del usuario y garantizar la trazabilidad de cada solicitud.

En respuesta a esta problemática, este proyecto plantea el diseño y desarrollo de un sistema integral para la gestión de préstamos de equipos audiovisuales y salas de informática en el entorno universitario. Este software está orientado a cubrir las necesidades específicas de las instituciones educativas, destacándose por su capacidad para automatizar procesos operativos, controlar accesos según roles y asegurar el correcto uso y seguimiento de los recursos. A diferencia de soluciones genéricas, esta herramienta integra funcionalidades de gestión de solicitudes, devoluciones, sanciones y soporte técnico, permitiendo además la generación de reportes personalizados que facilitan la toma de decisiones administrativas y el análisis del uso de recursos tecnológicos en la institución.

Los objetivos principales de este proyecto incluyen: (1) **analizar los requerimientos funcionales** con el fin de identificar las funcionalidades necesarias para la gestión de préstamos, devoluciones, seguimiento de equipos y salas, y la administración de sanciones, mediante la recolección y análisis de información clave del proceso institucional; (2) **diseñar la arquitectura del software** que permita estructurar adecuadamente las clases, interfaces y base de datos, utilizando herramientas de modelado como diagramas de casos de uso, diagramas de clases y prototipos de interfaces gráficas; y (3) **codificar el sistema** para implementar las funcionalidades definidas, garantizando un funcionamiento correcto, una integración eficaz con la base de datos y una interfaz adecuada para el usuario final.

Al ofrecer una solución que optimiza la gestión de los procesos internos relacionados con el préstamo de recursos tecnológicos, este sistema promueve la eficiencia administrativa y mejora significativamente la experiencia tanto del personal institucional como de los usuarios finales. Su implementación permite a la universidad mantener un control riguroso sobre el uso de equipos y espacios, reducir errores humanos, y responder de forma ágil a las necesidades académicas y operativas. De esta manera, el sistema tiene el potencial de transformar la forma en que se gestionan los recursos tecnológicos, alineándose con las exigencias de una

educación superior moderna y tecnológicamente equipada.

## Metodología

### Tipo de estudio

Este proyecto se basa en un **enfoque no experimental**, ya que no se manipularon variables de forma directa, sino que se analizaron las condiciones existentes en los procesos manuales actuales de **gestión de préstamos y devoluciones de equipos y salas**, con el fin de identificar oportunidades de mejora en la eficiencia operativa y el control institucional.

El diseño del estudio es de tipo **descriptivo**, pues busca detallar de forma precisa las características, necesidades y problemáticas que enfrentan los usuarios del sistema (personal administrativo, docentes y encargados de soporte) en relación con la gestión de recursos tecnológicos y espacios institucionales.

La investigación sigue un enfoque **cuantitativo**, centrado en la recolección y análisis de datos objetivos como la frecuencia de solicitudes, tiempos de préstamo, tasas de incumplimiento o sanciones, entre otros. Esta información se utilizó para fundamentar decisiones durante el diseño y desarrollo del software, asegurando que la solución responda a las necesidades reales de los usuarios y de la institución.

El estudio está orientado a la **creación de un producto funcional**, cuyo objetivo es implementar un sistema que automatice los procesos de solicitud, aprobación, devolución, seguimiento de equipos, control de sanciones y soporte técnico. Además, se contemplan aspectos como el control de acceso mediante roles, la generación de reportes y el uso eficiente de los recursos disponibles en el entorno universitario.

### Herramientas tecnológicas

Para garantizar un desarrollo eficiente y estructurado del proyecto, se emplearon diversas herramientas tecnológicas que facilitaron la implementación, organización y control del sistema. Estas herramientas incluyeron:

**Lenguaje de programación:** Se utilizó **Java** como lenguaje principal para el desarrollo del software, aprovechando su robustez, orientación a objetos y compatibilidad con entornos de desarrollo como NetBeans. Java permitió construir una estructura modular, facilitando la escalabilidad y el mantenimiento del sistema.

**Gestión de bases de datos:** Se implementó una **base de datos Oracle**, seleccionada por su fiabilidad y capacidad

para manejar grandes volúmenes de datos de forma segura. Oracle permitió gestionar adecuadamente la información relacionada con usuarios, préstamos, devoluciones, equipos, salas y sanciones, manteniendo la integridad y consistencia de los datos.

**Control de versiones:** Se utilizó **GitHub** como plataforma para el almacenamiento del código fuente y control de versiones del sistema. Esto permitió mantener un respaldo constante del proyecto, registrar el historial de cambios y facilitar la colaboración y seguimiento del desarrollo en cada una de las fases.

## Procedimiento

Para alcanzar los objetivos propuestos en esta investigación, se siguió un procedimiento metodológico dividido en tres fases principales:

### 1. Análisis de requerimientos funcionales

Se realizó la identificación y recolección de información relevante para definir las funcionalidades necesarias del Software, enfocándose en la gestión de préstamos, devoluciones, seguimiento de equipos y salas, y la administración de sanciones. Este proceso se llevó a cabo mediante entrevistas con los usuarios finales y la revisión documental de los procedimientos actuales. Se utilizó un protocolo estructurado para la obtención de requisitos que permitió asegurar la validez y la exhaustividad de la información recolectada.

### 2. Diseño de la arquitectura del software

Con base en los requerimientos definidos, se diseñó la arquitectura del Software utilizando diagramas UML, incluyendo diagramas de casos de uso y diagramas de clases. Asimismo, se modeló la base de datos mediante diagramas entidad-relación para garantizar la integridad y la eficiencia en el manejo de la información. Para la elaboración de estos modelos se emplearon herramientas especializadas de diseño como [nombre de la herramienta, por ejemplo, Visual Paradigm o StarUML, versión], que permitieron generar esquemas claros y precisos. Además, se diseñaron las interfaces gráficas de usuario, priorizando la usabilidad y funcionalidad.

### 3. Codificación e implementación del software

En esta etapa, se llevó a cabo la programación del software, implementando las funcionalidades definidas en la fase de diseño. Se utilizó el entorno de desarrollo integrado NetBeans [versión 17], empleando Java como lenguaje de programación, y Oracle como sistema gestor de base de datos. La conexión y manipulación de datos se realizó a través de JDBC. Las pruebas funcionales se efectuaron en múltiples sesiones, realizando de ensayos para validar la correcta gestión de préstamos, devoluciones y sanciones. Durante el desarrollo se mantuvo un control de versiones utilizando

[Git], y se aplicaron pruebas unitarias y de integración para garantizar la calidad del software.

## Resultados

El sistema desarrollado cumplió con los requerimientos funcionales definidos por el cliente, logrando:

**Interfaz amigable y fácil de usar:** El diseño del sistema se centró en ofrecer una experiencia intuitiva para los usuarios, facilitando la rápida adaptación al software, independientemente de su nivel técnico. Esto permitió que los empleados puedan gestionar préstamos, devoluciones y el seguimiento de equipos y salas sin dificultades, optimizando la productividad desde las primeras etapas de implementación.

**Gestión de roles y seguridad:** Se implementó un sistema de gestión de roles que garantiza que cada usuario acceda únicamente a las funcionalidades y datos pertinentes a su perfil, protegiendo la información sensible y minimizando riesgos asociados a accesos no autorizados o mal uso del sistema.

### Centralización y organización de la información:

Toda la información relacionada con equipos, salas, usuarios y sanciones está centralizada y organizada dentro del sistema, lo que mejora la precisión de los datos y facilita la consulta y actualización en tiempo real, garantizando una gestión más efectiva y controlada de los recursos.

## Agradecimientos

De manera especial, agradecemos a Todo Code, canal de YouTube, por el valioso material y tutoriales que nos apoyaron en la codificación y el aprendizaje del lenguaje Java, contribuyendo significativamente al desarrollo técnico del software.

## Referencias

IBM. (s. f.). Creating a Java project. Recuperado de <https://www.ibm.com/docs/es/integration-bus/10.0?topic=node-creating-java-project>

ubSpot. (s. f.). Cómo programar con Java. Recuperado de <https://blog.hubspot.es/website/como-programarcon-java>

GanttPRO. (s. f.). ¿Qué es el método de cascada?. Recuperado de <https://blog.ganttpro.com/es/metodologia-de-cascada/#:~:text=ágiles%20y%20cascada-,¿Qué%20es%20el%20método%20de%20cascada%3F,se%20haya%20completado%20la%20anterior.>

Apache NetBeans. (s. f.). Connecting to Oracle database. Recuperado de

<https://netbeans.apache.org/tutorial/main/kb/docs/ide/oracle-db/>

SafetyCulture. (s. f.). Metodología en cascada: guía y ejemplos. Recuperado de <https://safetyculture.com/es/temas/metodologia-en-cascada/>