

# **Manual Técnico**

## **Introducción**

Este proyecto, denominado **"Integrador"**, consiste en el desarrollo de una plataforma para gestionar los préstamos de salas y equipos audiovisuales. El propósito es optimizar la asignación de recursos, controlar los horarios de uso y ofrecer una administración más eficiente de estos activos institucionales.

En muchas instituciones académicas o centros culturales, la gestión de reservas y préstamos se realiza manualmente o con sistemas fragmentados, lo que genera conflictos de horarios, uso indebido de recursos o falta de trazabilidad. Esta plataforma busca resolver esos problemas mediante una solución centralizada y automatizada.

Este manual técnico cubre la configuración y el uso básico de la plataforma, incluyendo la administración de salas, equipos audiovisuales, registros de préstamos y devoluciones. No incluye personalizaciones avanzadas ni configuraciones específicas por institución.

## **Descripción del Problema**

La gestión de reservas de salas y equipos audiovisuales es frecuentemente desorganizada. Las solicitudes suelen realizarse de forma manual, con seguimiento limitado, sin alertas sobre disponibilidad ni control efectivo del uso o devolución de los recursos.

"Integrador" ofrece una solución que centraliza toda esta gestión, facilitando el registro de préstamos, control de disponibilidad, historial de uso, e identificación de responsables en caso de pérdidas o retrasos en devoluciones.

### **3. Objetivos**

#### **3.1. General**

Desarrollar un Software para la gestión de solicitudes de préstamos, devoluciones, gestión de sanciones, gestión de usuarios y seguimiento de equipos audiovisuales y salas de informática para la universidad del oriente.

#### **3.2. Específico**

Analizar los requerimientos funcionales, con el fin de identificar las funcionalidades para la gestión de los préstamos, devoluciones, seguimiento de equipos y salas, gestión de sanciones, mediante la recolección de información.

Diseñar la arquitectura del Software, con el fin de establecer las clases, interfaces y la base de datos, mediante la elaboración de diagramas UML como casos de uso y de clases, el modelado de la base de datos y la creación de las interfaces gráficas, utilizando herramientas de diseño.

Codificar el Software con el fin de desarrollar las funcionalidades definidas y garantizar el correcto funcionamiento y gestión de la información, mediante la implementación del diseño de software y la adaptación con la base de datos.

### **Justificación**

Actualmente, la gestión de recursos como salas y audiovisuales se realiza con métodos poco eficientes, lo cual provoca desorganización, conflictos de uso, y pérdida de tiempo. Este sistema busca profesionalizar la gestión de estos recursos, incrementando la eficiencia y transparencia, al tiempo que mejora la experiencia del usuario y del personal administrativo.

## **Propuesta del plan de proyecto**

El proyecto "**Integrador**" se basa en el modelo de desarrollo en cascada, siguiendo una secuencia estructurada desde el análisis hasta la entrega final. El sistema incluirá módulos para:

Gestión de salas.

Gestión de equipos audiovisuales.

Registro de préstamos y devoluciones.

Control de disponibilidad.

Roles diferenciados.

Reportes y alertas.

## **Recursos Necesarios**

### **Humanos**

Desarrolladores de software

Analistas de sistemas

Equipo de pruebas

Personal de soporte y capacitación

### **Técnicos**

Computadoras y licencias de desarrollo

Servidores de pruebas. Bases de datos y herramientas de respaldo

## **Etapas de Desarrollo y Entregables**

1. Documento de requisitos
2. Modelo de base de datos y prototipos de interfaz
3. Sistema funcional con gestión de recursos, préstamos y reportes
4. Informe de pruebas y correcciones
5. Manuales de usuario y técnico, y entrega del sistema

## **Análisis de requerimientos del software**

El sistema debe permitir crear, modificar y eliminar salas y equipos audiovisuales.

Debe poder registrar la disponibilidad de salas y equipos, incluyendo horarios, mantenimiento y uso exclusivo.

Los usuarios deben poder registrarse y solicitar préstamos indicando fechas, horarios y recursos deseados.

Los administradores deben aprobar o rechazar solicitudes según disponibilidad y políticas internas.

Debe llevar un registro de préstamos realizados, responsables, fechas de devolución y observaciones.

El sistema debe generar alertas de vencimiento, reportes de uso y disponibilidad.

Debe contar con roles diferenciados: administradores (gestión total) y usuarios (solicitudes y seguimiento).

El sistema debe contar con autenticación segura y cumplir normativas de protección de datos.

La interfaz debe ser intuitiva y accesible para que los usuarios puedan operar el sistema sin capacitación avanzada.

## **CASOS DE USOS**

### **Registro de Usuarios (Solicitantes)**

Este caso de uso lo realiza el Administrador, encargado de registrar a los usuarios que podrán solicitar préstamos. Se registran con los siguientes datos:

Nombre, apellido, email, contraseña, rol.

### **Gestión de Préstamos**

Caso de uso que realizan los Usuarios para registrar solicitudes de préstamo, y los Administradores para aprobarlas o gestionarlas. Incluye:

tipo de equipo o sala, tiempo de uso.

### **Registro de Devoluciones**

Realizado por el Usuario o validado por el Administrador. Permite dejar constancia de que el recurso fue devuelto, con detalles como:

fecha, hora, observaciones.

### **Gestión de Soporte Técnico**

Tanto Usuarios como Administradores pueden reportar incidentes relacionados con los equipos o salas. Se gestionan mediante:

fecha, hora, observaciones.

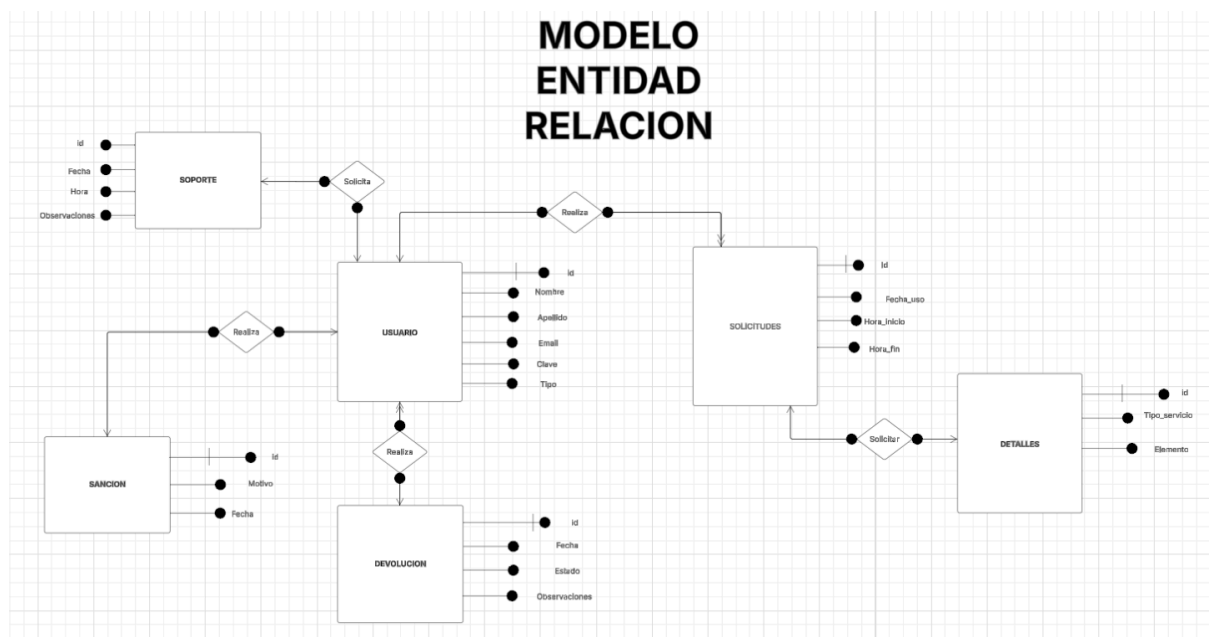
### **Gestión de Sanciones**

Solo el Administrador puede aplicar sanciones por mal uso, daño de equipos, o incumplimiento en devoluciones. Este caso de uso incluye:

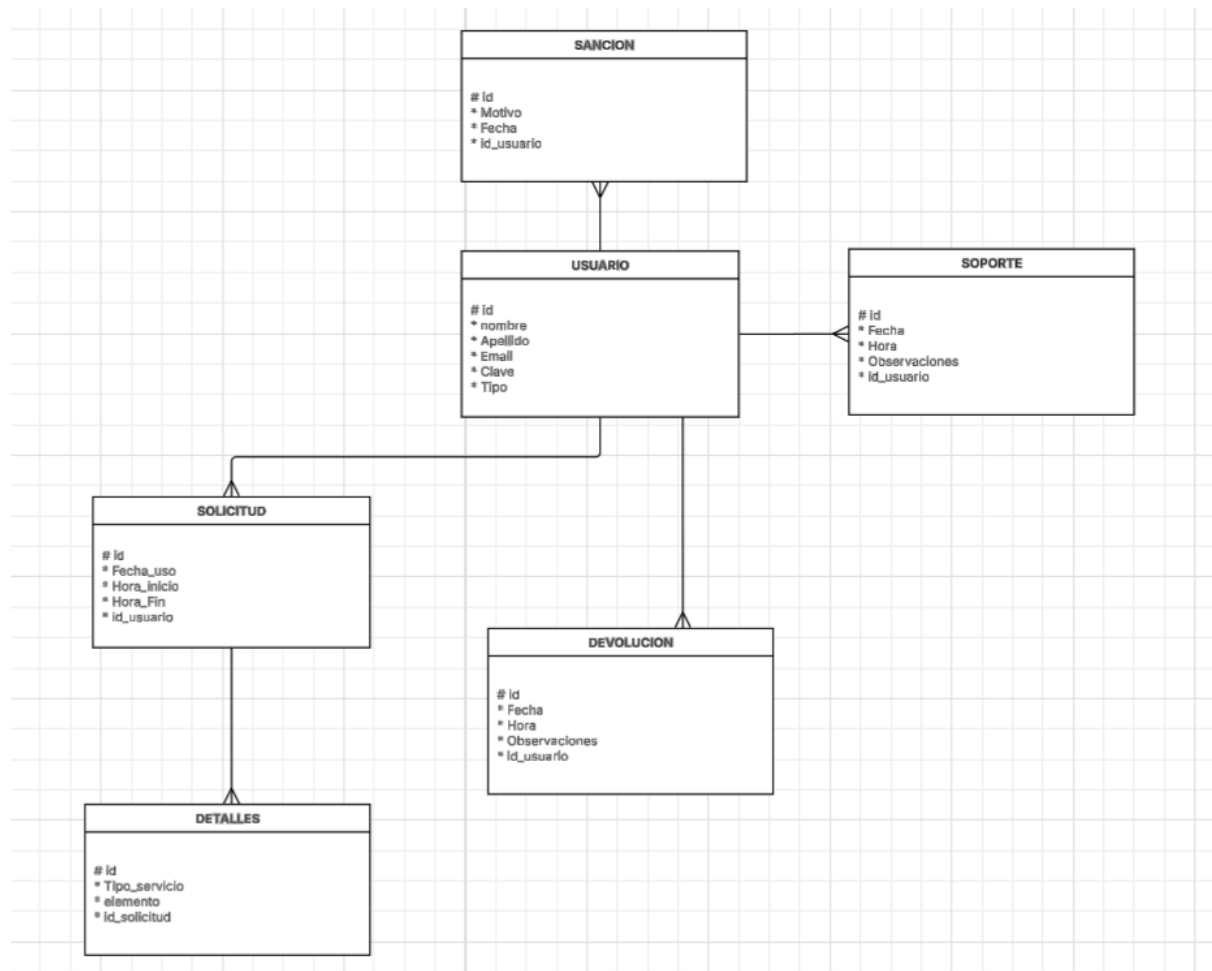
Fecha, motivo.

## Modelamiento De La Base De Datos

### modelo entidad-relación



### Diagrama relacional



## Usuario

Los usuarios pueden ser solicitantes o administradores del software. Es fundamental tener su información registrada para cualquier acción dentro del software.

- Representa a toda persona que interactúa con el software.
- Se registra con datos como nombre, correo, tipo y credenciales.
- Está relacionado con solicitudes, devoluciones, sanciones y reportes de soporte.

## Solicitud

Corresponde al registro formal del préstamo de un recurso (sala o equipo) por parte de un usuario.

- Define cuándo se usará un recurso (fecha, hora de inicio y fin).

- Está asociada a un usuario solicitante.
- Cada solicitud puede tener varios detalles (recursos específicos).

## Detalles

Especifica los elementos individuales (equipos o salas) incluidos en una solicitud.

- Permite registrar qué tipo de servicio se pidió y cuál fue el recurso.
- Relaciona directamente una solicitud con los elementos solicitados.
- Define si el préstamo fue de tipo sala, audiovisual u otro.

## Devolución

Registro del momento en que el usuario devuelve un recurso al software.

- Permite comprobar que el recurso fue devuelto correctamente.
- Incluye observaciones sobre el estado del recurso.
- Asociado al usuario responsable.

## Sanción

Registra penalizaciones impuestas a los usuarios que incumplen normas.

- Puede aplicarse por devolución tardía, daño de recursos u otras infracciones.
- Está directamente relacionada con el usuario sancionado.
- Incluye motivo, fecha y tipo de sanción.

## Soporte

Permite a los usuarios reportar incidentes técnicos o problemas con recursos.

- Gestionado también por administradores para seguimiento.



- Incluye fecha, hora, observaciones y usuario que lo reporta.
- Ayuda a mantener el estado funcional de los recursos del sistema.

## **Código documentado**

<https://github.com/duvanleandro/ProyectoIntegradorUDI/tree/main>

En el repositorio de GitHub se encuentra almacenada toda la información esencial para comprender y analizar el funcionamiento del software desarrollado. Esta sección incluye documentos detallados y archivos técnicos que abarcan desde la descripción inicial del proyecto hasta el diseño, implementación y pruebas del sistema.

Dentro de esta rama, se pueden encontrar elementos clave como el análisis de requerimientos, diagramas UML para modelar los diferentes aspectos del software, especificaciones de diseño, detalles sobre la configuración de la base de datos y el código fuente documentado. Además, esta documentación proporciona una guía clara que detalla los objetivos del sistema, las metodologías utilizadas durante el desarrollo (como el enfoque en cascada), y evidencia del cumplimiento de buenas prácticas de programación y diseño.

## **Código Fuente Versión Release**

### **Descripción de la versión final del código.**

En esta versión final se puede ver que el Software se puede hacer el login y dependiendo del rol del usuario, ingresará al panel de administrador o usuario y claramente los dos cumplen con funcionalidades diferentes, el administrador puede gestionar diferentes funcionalidades y el usuario registrar préstamos, devoluciones o soporte técnico, además de esto también se puede ver que se puede eliminar y modificar, esto incluye que se pueda ver las tablas donde se tiene todos los datos y así poder eliminarlos o editarlos.

## Instrucciones de instalación y ejecución del sistema.

El sistema se encuentra en un archivo rar para descargar, una vez descargado se extrae para después abrirlo en el Netbeans, donde dice open folder, se busca donde se guardó el archivo y se abre, una vez se haga esto se puede iniciar el Software donde la primera interfaz gráfica que se puede ver es el login, se ingresa con un usuario y contraseña y se especifica si es administrador o usuario, con esto ya se podría ingresar a las funcionalidades de este software y todo lo que contiene, la versión que se está utilizando es la 19, también hay que tener en cuenta que se debe tener el jdk.

## REFERENCIAS:

Northware. (s.f.). *Requerimientos en el desarrollo de software y aplicaciones*.  
Northware. Recuperado el 2 de junio de 2025, de Visure Solutions. (s.f.). *Requirements definition*.

Visure Solutions. Recuperado el 2 de junio de 2025, de <https://visuresolutions.com/es/blog/requirements-definition/>

DiagramasUML. (s.f.). *Diagrama de clases*. DiagramasUML. Recuperado el 2 de junio de 2025, de <https://diagramasuml.com/diagrama-de-clases/>

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-programarconjava>

GanttPRO. (s. f.). ¿Qué es el método de cascada?. Recuperado de <https://blog.ganttpro.com/es/metodologia-de-cas>

[cada/#:~:text=ágiles%20y%20cascada-,¿Qué%2](#)

[0es%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/>