**Práctica Final**

**de Curso INDRA**

Desarrollo de una Aplicación Java con Scrum y MySQL

# **Sistema de Gestión de Reservas de Salas de Reunión**

## 

Aleix Terés Pareja

Tutor de Practica: Fernando Silvano Gil Peréz

DAW – Desarrollo de Aplicaciones Web

Fecha: 27 de junio de 2025

Repositorio: <https://github.com/Silax17/Practica-Final-de-Curso>

[**Sistema de Gestión de Reservas de Salas de Reunión 1**](#_3j7r5ul9gc69)

[**1. Visión del Proyecto 3**](#_cefr9omgebbn)

[**2. Especificación del Proyecto 3**](#_ux298fnrsjuh)

[2.1 Requisitos funcionales 3](#_w7xtr1bb0i9)

[2.2 Requisitos Técnicos 3](#_ha81uxxa8krf)

[**3. Sprint 1 — Diseño y Preparación (Lunes y Martes) 4**](#_ilc9w446hhn4)

[3.1 Objetivos específicos 4](#_av4dfu71zzx6)

[3.2 Historias de usuario, Tareas y Estimaciones 4](#_oly78snptxmn)

[3.3 Tablero Scrum — Sprint 1: Sistema de Reservas 5](#_ejz7xqol5fxf)

[4. Reflexión Sprint 1 6](#_fhfawq8cvaik)

[**5. Sprint 2 - Desarrollo Funcional (Miércoles a Viernes) 7**](#_evdazsyr8u72)

[5.1 Objetivos específicos 7](#_3zjqwuuj57ia)

[5.2 Historias de usuario, Tareas y Estimaciones 7](#_esfl1yq4g537)

[5.3 Tablero Scrum — Sprint 2 8](#_dzilni7b7koc)

[6. Reflexión Sprint 2 9](#_9ac73u96z3sv)

## **Visión del Proyecto**

El objetivo de este proyecto es desarrollar un sistema de reservas para salas de reuniones en una empresa. La aplicación está construida en Java (versión 11 o superior, con persistencia en una base de datos MySQL y gestión del proyecto mediante Maven. El enfoque se basa en prácticas ágiles con la metodología Scrum, simulando un entorno de trabajo real dividido en dos sprints

## **Especificación del Proyecto**

### *2.1 Requisitos funcionales*

La aplicación desarrollada permite:

1. **Gestión de salas**: Registrar, listar, editar y eliminar salas de reuniones. Cada sala incluye nombre, capacidad y recursos disponibles.
2. **Gestión de empleados**: Administración de empleados con datos como nombre, correo y departamento.
3. **Reservas:** Permite a un empleado reservar una sala para una fecha y franja horaria determinada.
4. **Control de conflictos:** Valida que no existan reservas duplicadas en la misma sala y horario

### *2.2 Requisitos Técnicos*

* **Lenguaje**: Java 11
* **Gestión del proyecto**: Maven
* **Base de datos**: MySQL
* **Persistencia:** JDBC (Java Database Connectivity)
* **Testing:** JUnit y Mockito
* **IDE utilizado:** Eclipse
* **Gestión de versiones:** Git y GitHub
* **Modelo de desarrollo:** Scrum, dividido en 2 sprints, una semana entere

## **Sprint 1 — Diseño y Preparación (Lunes y Martes)**

El Sprint 1 se centró en sentar las bases del proyecto. Durante estos días se diseño la estructura de la base de datos, se creó el modelo entidad-relación, y se preparó el entorno de desarrollo con Maven. Este sprint se enfocó en conseguir una base sólida para las siguientes partes del desarrollo.

### **3.1 Objetivos específicos**

* Diseñar el modelo entidad-relación (ER)
* Crear el esquema de base de datos (Salas, Empleados, Reservas)
* Insertar datos de prueba
* Inicializar el proyecto Java con Maven

### **3.2 Historias de usuario, Tareas y Estimaciones**

| ID | Historia de Usuario | Tareas | Estimación |
| --- | --- | --- | --- |
| H1- Registrar salas | Como administrador, quiero registrar salas de reuniones para poder asignarlas | * Crear tabla *Sala* (SQL) | 1h |
| H2- Gestionar empleados | Como administrador, quiero gestionar empleados para tenerlos en el sistema | * Crear tabla *Empleado* (SQL) | 1h |
| H3- Diseñar base de datos | Como desarrollador, necesito diseñar la base de datos para organizar la información | * Diagrama ER * Crear tabla *Reserva* (Relación SQL) * Generar Script SQL * Insertar datos de prueba | 4h |

### **3.3 Tablero Scrum — Sprint 1: Sistema de Reservas**

| Historia de Usuario | Tarea específica | Estado | Estimación |
| --- | --- | --- | --- |
| H1-Registrar salas | Crear tabla Sala (SQL) | Hecho | 1h |
| H2- Gestionar empleados | Crear tabla Empleado (SQL) | Hecho | 1h |
| H3- Diseñar base de datos | Diseñar Diagrama ER | Hecho | 2 h |
| Crear script SQL | Hecho | 1 h |
| Insertar datos de prueba | Hecho | 1h |
| Extra- Setup Inicial | Crear proyecto Maven | Hecho | 0.5 h |

## **Reflexión Sprint 1**

Este sprint se completó en su totalidad. El diseño del modelo ER ayudó a entender claramente las relaciones entre entidades. La creación de las tablas y la configuración inicial del proyecto se realizaron sin complicaciones. El uso de datos de prueba permitió comenzar el desarrollo funcional en el siguiente sprint sin bloqueos.

## 

## 

## 

## 

## **Sprint 2 - Desarrollo Funcional (Miércoles a Viernes)**

Con una base sólida del Sprint 1, el Sprint 2 se busco el desarrollo completo de la lógica del sistema. Se implementaron los CRUD de todas las entidades, validaciones lógicas como la detección de conflictos de horarios en reservas, conexión con la base de datos y pruebas automatizadas. Este sprint marca la finalización del desarrollo del proyecto

### **5.1 Objetivos específicos**

* Completar el CRUD de Salas y Empleados
* Implementar el CRUD para Reservas
* Validar conflictos de horarios en reservas
* Establecer y probar la conexión Java-MySQL
* Implementar pruebas unitarias conJUnit

### **5.2 Historias de usuario, Tareas y Estimaciones**

| ID | Historia de Usuario | Tareas | Estimación |
| --- | --- | --- | --- |
| H1 | Como administrador, quiero registrar salas de reuniones | -Crear clase Sala en Java  -Implementar CRUD completo | 3h |
| H2 | Como administrador, quiero gestionar empleados | -Crear clase Empleado en Java  -Implementar CRUD completo | 3h |
| H4 | Como administrador quiero registrar reservas | -Crear clase Reserva  -CRUD completo  -Validación de horarios | 4h |
| H5 | Como desarrollador, quiero conectar Java con MySQL | -Establecer conexión y probar acceso | 2h |
| H6 | Como desarrollador, quiero verificar el funcionamiento | -Implementar 5 tests con JUnit | 3h |

### 

### **5.3 Tablero Scrum — Sprint 2**

| Historia de Usuario | Tarea específica | Estado | Estimación |
| --- | --- | --- | --- |
| H1-Registrar salas | Crear Sala (Java) | Hecho | 1h |
| Implementar CRUD completo | Hecho | 2h |
| H2-Gestionar empleados | Crear clase Empleado (Java) | Hecho | 1h |
| Implementar CRUD completo | Hecho | 2h |
| H4-Gestionar reservas | Crear clase Reserva | Hecho | 1h |
| Implementar CRUD de reservas | Hecho | 2h |
| Validar conflictos de horario | Hecho | 2h |
| H5-Conexión BD | Establecer conexión Java-MySQL | Hecho | 2h |
| H6-Pruebas unitarias | Implementar 5 pruebas JUnit | Hecho | 3h |

### **Reflexión Sprint 2**

El desarrollo de las funcionalidades se completó según lo planificado. La validación de conflictos en reservas requirió atención especial, per fue superada correctamente. Las pruebas unitarias permitieron comprobar el correcto funcionamiento del sistema. El sistema está listo para ejecutarse correctamente.