**Documentación Scrum Jan Altisent**

# **Proyecto:** Gestor de Reservas de Salas de Reuniones en una Empresa

### **Alumno:** Jan Altisent Brescó



## Introducción

El proyecto consiste en la creación de una aplicación en Java que actúe como una Gestión de Reservas de Salas de Reuniones en una Empresa. Esta tiene que ser de tipo CRUD completo (Create, Read, Update, Delete), lo cual permitirá aplicar este control que logre la principal funcionalidad del proyecto. Crear, listar, actualizar y eliminar las salas según los trabajadores y reservas. La aplicación consta además de la gestión de conflictos en caso de una posibilidad de reservar doblemente una sala. Asegurar que no puede estar reservada por más de una persona en una misma fecha y horario.

Este documento ha sido escrito mediante el uso de la metodología Scrum. Dividiendo el proyecto en 2 Sprints, cada uno diferenciando los objetivos y las tareas correspondientes, siendo organizadas en un tablero Kanban personalizado.

## Tablero Kanban

Un tablero Kanban funciona como una herramienta que permite una visualización que ayuda al usuario a organizar y hacer una gestión de las tareas del proyecto actual. Este se compone en 3 tipos de columnas divididas, donde se representan los diferentes estados del trabajo.

* Tareas pendientes de realización (To Do)
* Tareas en progreso de realización (In Progress)
* Tareas completadas (Done)

## Sprints

El proyecto se ha dividido en 2 sprints, anteriormente mencionado, usando la metodología Scrum, junto a una herramienta visual como función de soporte mediante el método Kanban.

### **Sprint 1: Diseño y Ejecución Base del Proyecto**

Este Sprint del proyecto tiene como objetivo la realización de las bases necesarias para el desarrollo de la Gestión del Sistema de Reservas de Salas de Reuniones en una Empresa.

Mediante la ejecución de este primer Sprint, en el IDE Eclipse, es necesaria la estructura Maven del proyecto, realizado en el código de programación Java, permitiendo una correcta organización del código fuente, dependencias y recursos.

Siguiendo el patrón de trabajo MVC (Modelo-Vista-Controlador) que estructura el proyecto, donde se organizan las clases.

Se ha diseñado la base de datos en MySQL, donde se almacena la información de las salas, trabajadores y las reservas.

**Objetivos:**

1. ***Configuración del proyecto Maven funcional:*** El uso de la herramienta Maven permite generar una estructura en el proyecto en Java. Facilitando una gestión de dependencias, compilación, pruebas y organización del código. La correcta creación del proyecto Maven asegura las futuras integraciones de bibliotecas externas necesarias en el proyecto gestionando de manera sencilla y escalable, MySQL y JUnit5.
2. ***Implementación CRUD en la Sala de Reuniones:*** La realización de tipo CRUD (Create, Read, Update, Delete), será fundamental para gestión de datos. La principal funcionalidad del proyecto es crear, hacer una visualización y realizar actualizaciones y eliminar registros. Cumpliendo las necesidades de la empresa para gestionar sus espacios de reuniones.
3. ***Creación de un Script SQL que contenga una estructura de las tablas y datos de prueba:*** La creación de la base de datos donde se almacenará toda la información que se genere cuando la aplicación se utilice es clave para garantizar una coherencia y tener una integridad. El Script permitirá al usuario recrear la base de datos y utilizar la aplicación.
4. ***Conexión entre la aplicación Java y la base de datos MySQL:*** El objetivo anterior no funcionará si no existe una conexión entre la aplicación y la base de datos, por lo que no se podrá realizar una gestión de los datos. Las operaciones CRUD podrán ser ejecutadas y persistir de forma correcta en la base de datos. Haciendo la aplicación operativa.
5. ***Diagrama Entidad-Relación (E-R):*** La creación de este diagrama permite una visualización esquemática de la estructura de las tablas y sus relaciones. Ayudando a poder detectar al instante errores que pueden surgir en el desarrollo y que la estructura de la aplicación sea lógica y consistente. A parte de ser una especie de guía para el programador, será de gran utilidad a la hora de documentar el proyecto y facilitar la ejecución.

### **Sprint 2: Desarrollo Avanzado i Finalización del Proyecto**

Este segundo Sprint del proyecto se centra en la consolidación de la funcionalidad completa del sistema de reservas, implementación de pruebas unitarias mediante JUnit 5, documentación del proceso y preparación de entrega final. El éxito de este Sprint cae en poder garantizar la mejor calidad estructural y tener un mantenimiento eficiente del código. Contando con una integración excelente mediante la herramienta Maven, una documentación genérica Scrum estructurada claramente en base a ensayos, pruebas y detección de errores.

**Objetivos:**

1. ***Creación y Ejecución de pruebas unitarias JUnit 5:*** Las clases desarrolladas en la aplicación del bloque de testeo se someten a una exhaustiva revisión de detección de errores previamente su ejecución final funcional. Se ejecuta de forma externa al bloque de código principal.
2. ***Detección y prevención de conflictos de reserva(reserva doble):*** Implementar un control de conflictos será una práctica esencial para evitar que una sala pueda ser reservada por más de un usuario en una misma hora y horario.
3. ***Migración estructural del proyecto Maven:*** La herramienta Maven, anteriormente citado, permite una gestión correcta de dependencias como MySQL o JUnit. Este objetivo mejorará la portabilidad del proyecto y facilitará la estándar industrial pedido en lenguaje Java.
4. ***Reorganización del proyecto en formato estructural estándar:*** Supondrá de una práctica que mejorará la visualización del proyecto y separará las responsabilidades de producción y pruebas. El IDE Eclipse actúa de forma automática en esta función, diferenciando la herramienta Maven del bloque de código en estado de pruebas.
5. ***Elaboración de documentación Scrum i informe del proceso según la metodología:*** Redactar una planificación como función de gestionar las tascas y reflexionar y proponer mejoras literarias propias de un entorno ágil.
6. ***Preparación*** [***README.md***](http://readme.md) ***i subida del proyecto en fase completa a GitHub:*** La herramienta GitHub proporciona un control de versiones y poder demostrar la evolución y la finalización del proyecto. Para el usuario que quiera acceder a la visualización de la aplicación, este documento actuará como una especie de introducción o resumen sobre su funcionamiento y cómo será ejecutada.

# Reflexión Final

En este proyecto, se ha llevado a cabo con éxito la gran mayoría de los objetivos obligatorios establecidos en la planificación de la práctica. La estructura del proyecto se ha logrado.

* Una correcta configuración de un entorno Maven permite su conexión con Java mediante sus dependencias.
* El diseño de su base de datos en MySQL con su correspondiente Script de creación de las tablas, junto a su desarrollo de las funcionalidades básicas tipo CRUD para la gestión de salas, trabajadores y reservas.
* Implementación de detección de conflictos para evitar el caso de una doble reserva.
* Pruebas unitarias mediante JUnit 5 en función de mejorabilidad tecnológica del código.

En esta documentación Scrum se refleja una estrategia de planificación, ejecución y revisión del proyecto. Finalmente, el proyecto ha sido preparado para su entrega y subida a GitHub, facilitando su mantenimiento y permitiendo futuras mejoras.

Lamentablemente, a pesar de estos avances, no se ha podido lograr que la aplicación funcione correctamente debido a problemas de conexión entre Java y la base de datos MySQL. Esta dificultad técnica es debido a la configuración del acceso desde la aplicación y de la herramienta SQLDeveloper. Sin poder ser comprendida ni ser solucionada dentro del plazo de la práctica. Por desgracia, la aplicación debido a este aspecto permaneció impedida de su operación, aunque su arquitectura y funcionalidad del sistema han sido correctamente implementadas y testeadas.

Finalmente, la creación de este proyecto sirve para identificar la importancia de la integración y un correcto funcionamiento y manejo de las bases de datos en proyectos reales en un futuro. Evidenciando áreas de mejora en el conocimiento y gestión de conexiones JDBC en Java. En futuras ocasiones, y si puede ser con un margen de tiempo más extenso y corrigiendo y aprendiendo de los errores, se garantiza una plena funcionalidad del sistema.