Práctica Final de Curso:

Desarrollo de una Aplicación
Java con Scrum y MySQL

Joel Eguren Guijarro

Contenido

1. Introducción

En este documento se presenta la planificación Scrum relacionada con el proyecto Java para la gestión de reservas de salas de reuniones en una empresa.

El objetivo principal es demostrar los conocimientos adquiridos durante el curso y en las dos semanas de formación en Minsait.

Las tecnologías usadas son:

- **Maven**, para la construcción del proyecto Java y la gestión de tests JUnit.
- **Java**, para la lógica del programa.
- **MySQL**, para la gestión de la base de datos.
- **DBeaver**, para la administración de la base de datos.
- **Trello**, para el tablero de tarjetas Scrum.
- **Git/GitHub**, para el control de versiones y el repositorio.

Este proyecto se desarrolla en un total de cuatro días. El primer día se ha dedicado a la planificación de tareas e historias de usuario en Trello, siguiendo la metodología Scrum. El desarrollo se ha dividido en dos sprints: dos días para el Sprint 1 y un día para el Sprint 2.

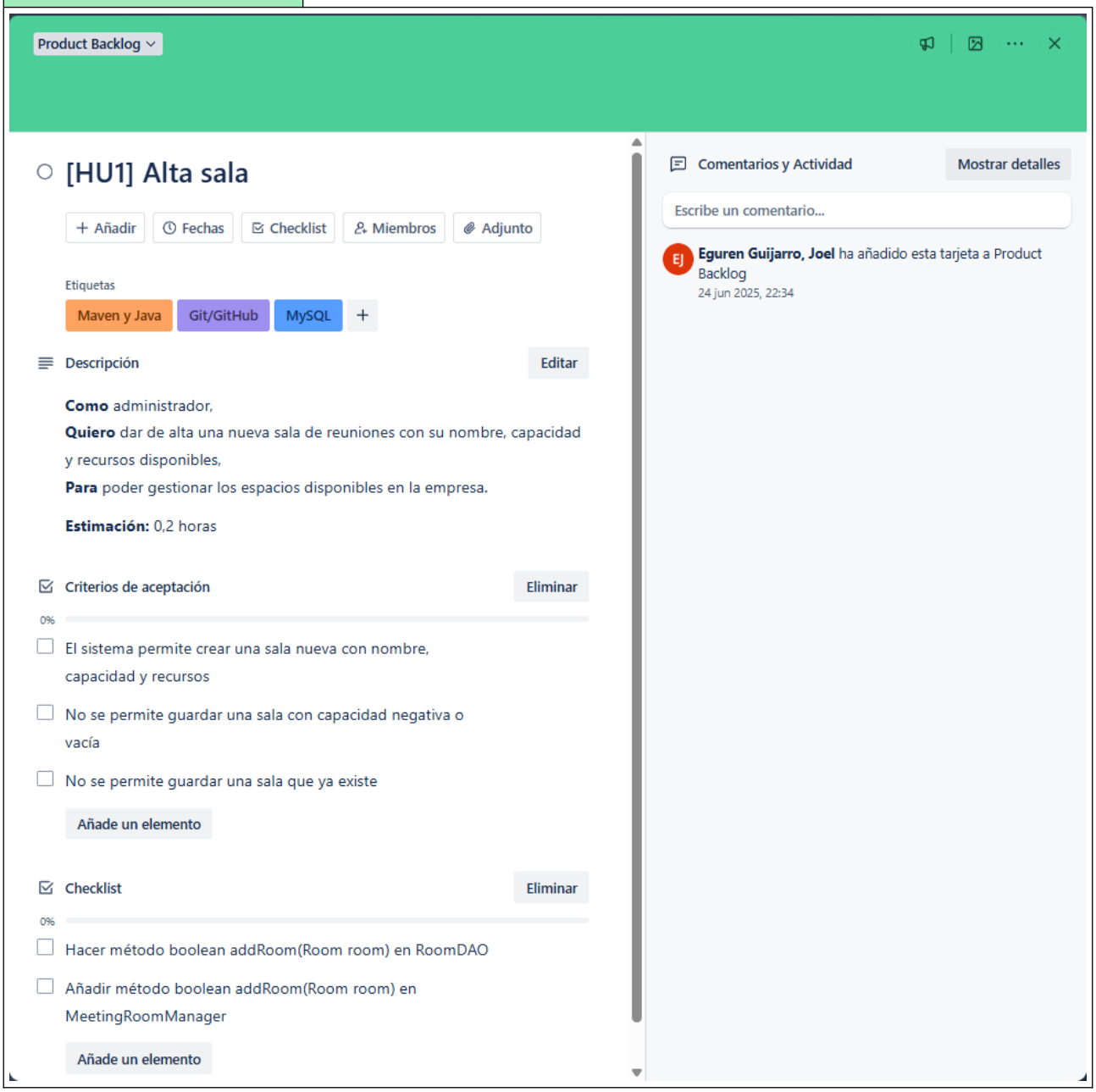
Se ha fijado un límite de *Work In Progress (WIP)* de 2 tarjetas cuando se realizan tareas técnicas sencillas, o de 1 tarjeta en el caso de historias de usuario. Esto permitirá optimizar el flujo de trabajo, porque nos centraremos en menos tareas a la vez.

(Enlace al tablero Scrum en Trello)

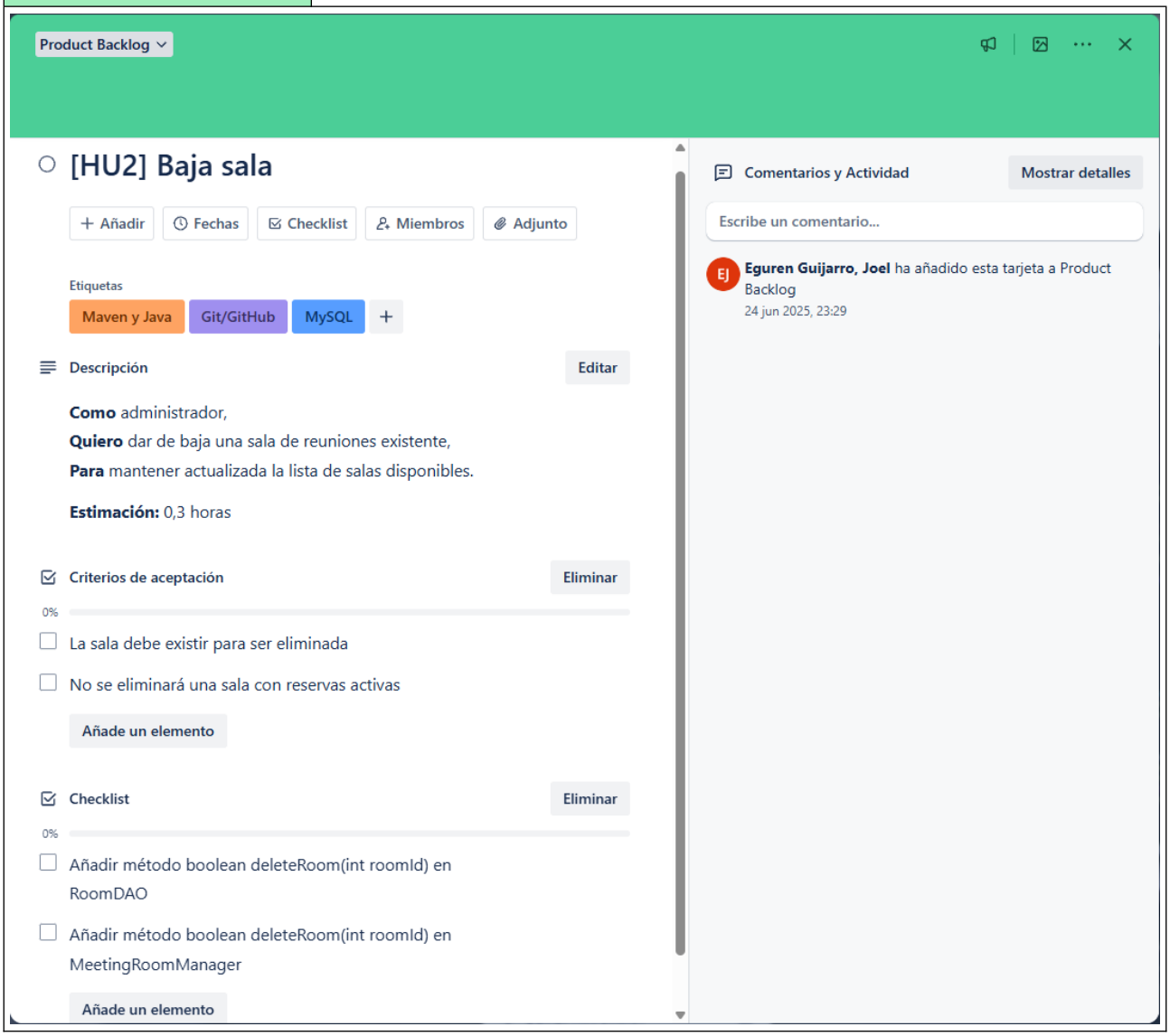
2. Historias de usuario

A continuación, se presentan todas las historias de usuario comprendidas en el proyecto usando Trello, el orden de las tarjetas se indica en cada Sprint.

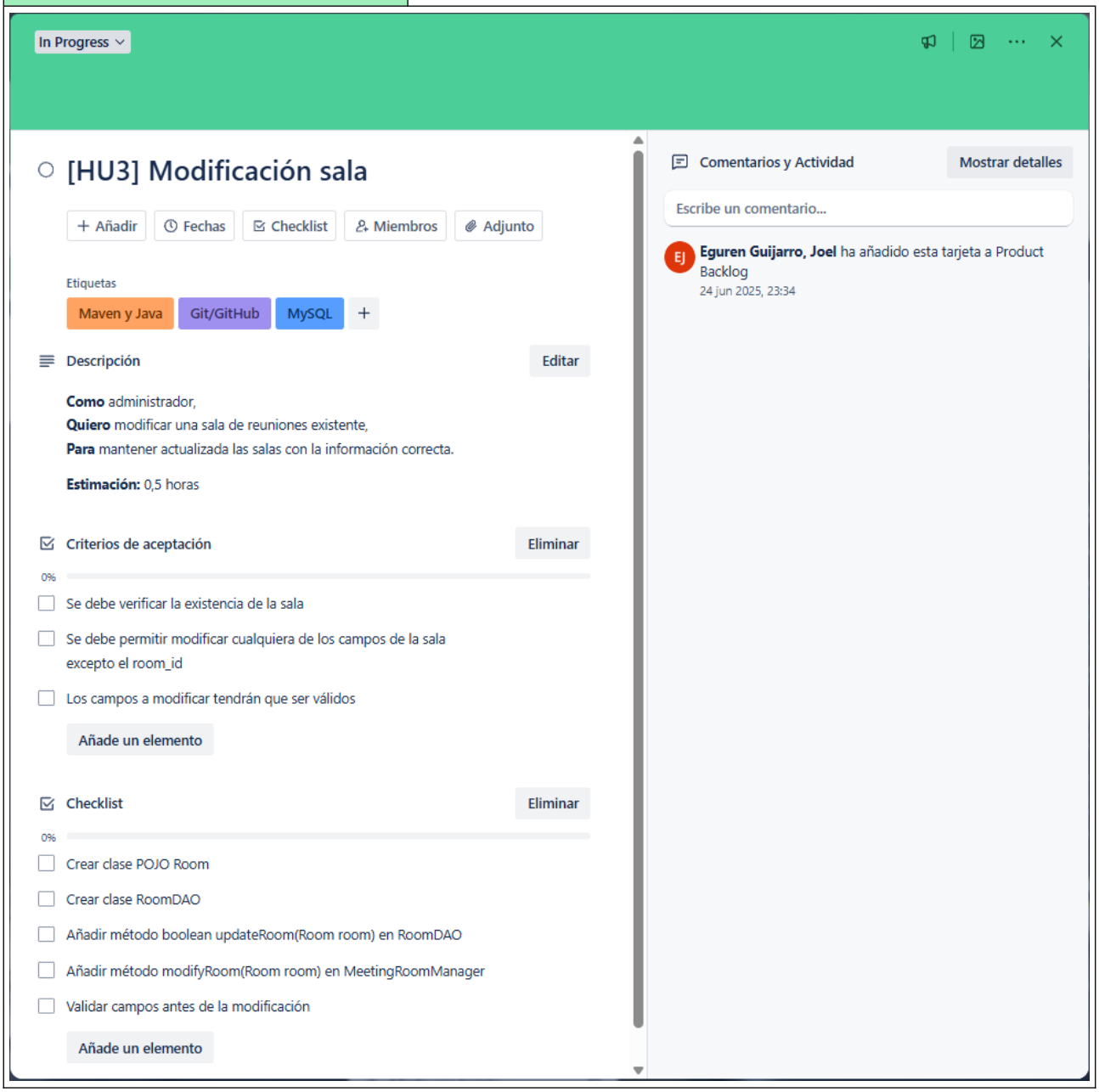
2.1. HU1 – Alta sala



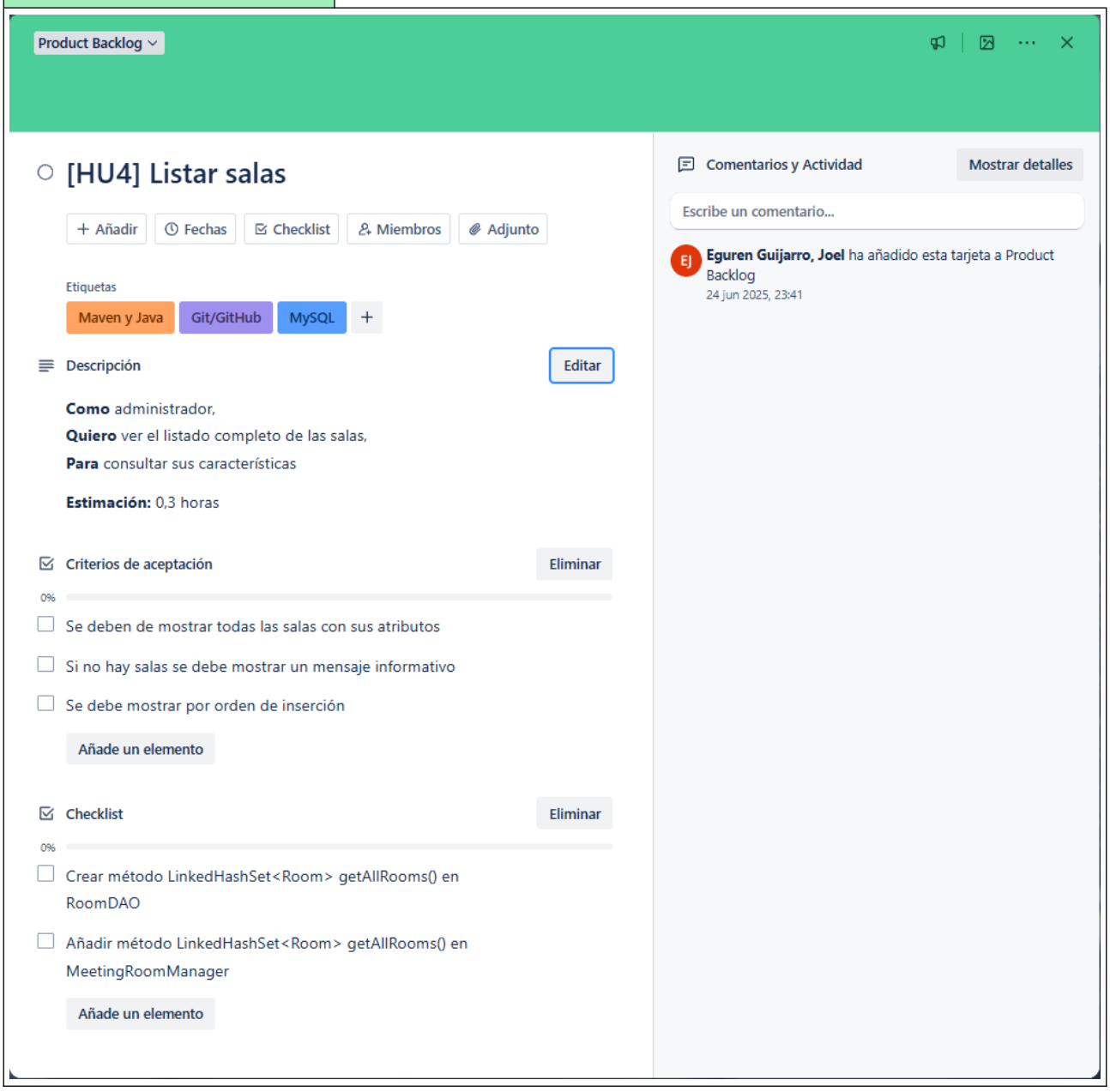
2.2. HU2 – Baja sala



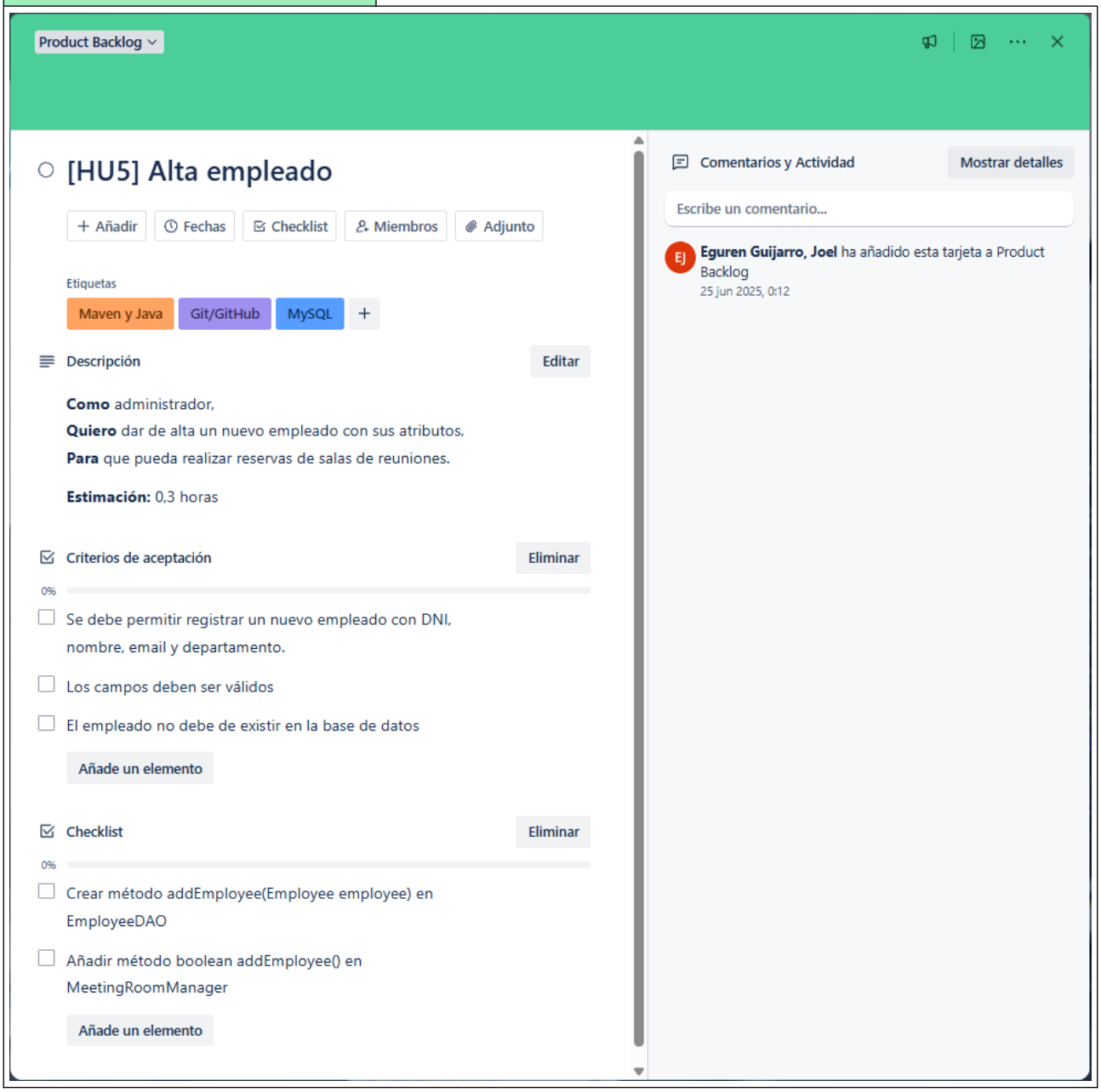
2.3. HU3 – Modificación sala



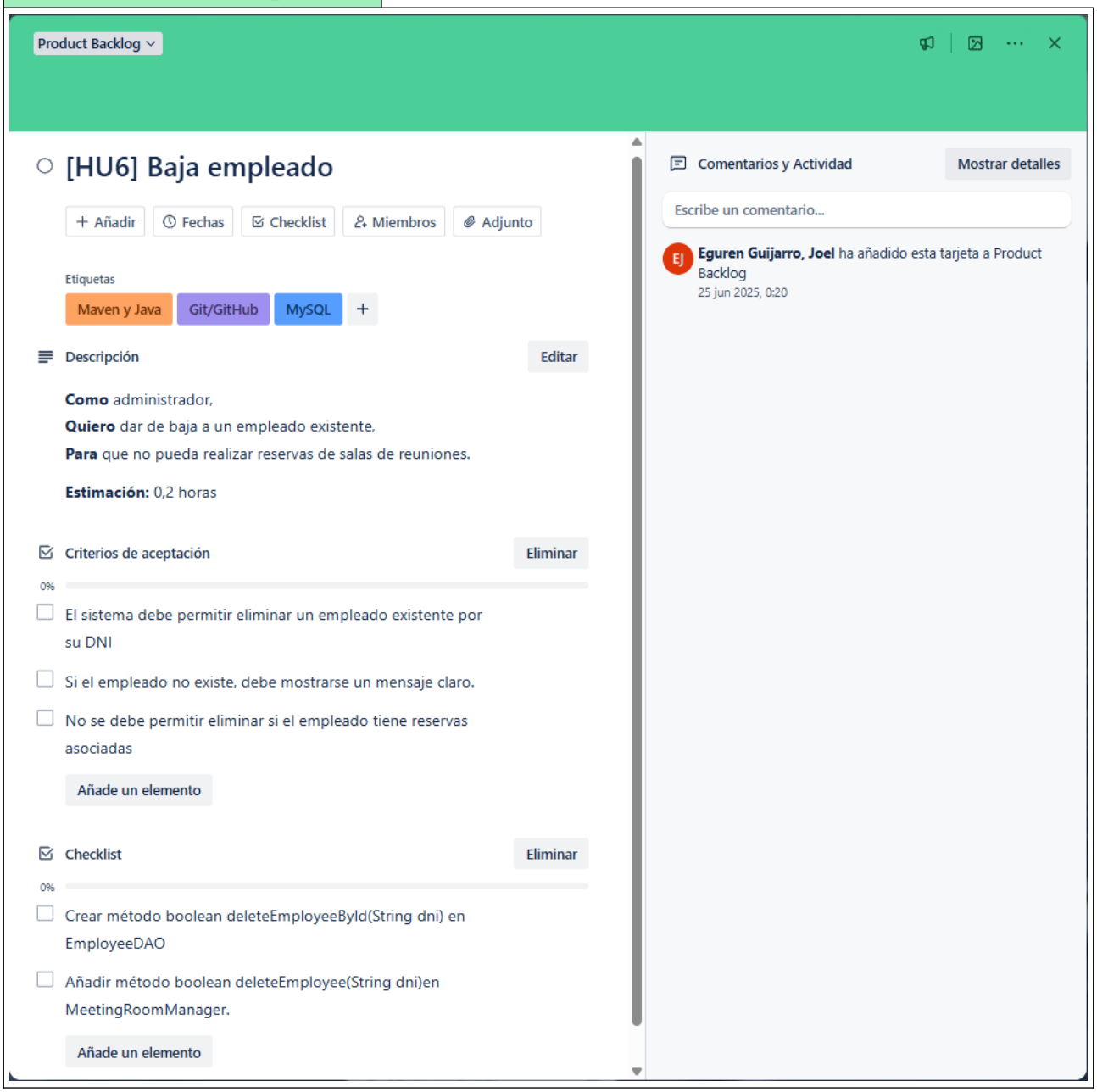
2.4. HU4 – Listar salas



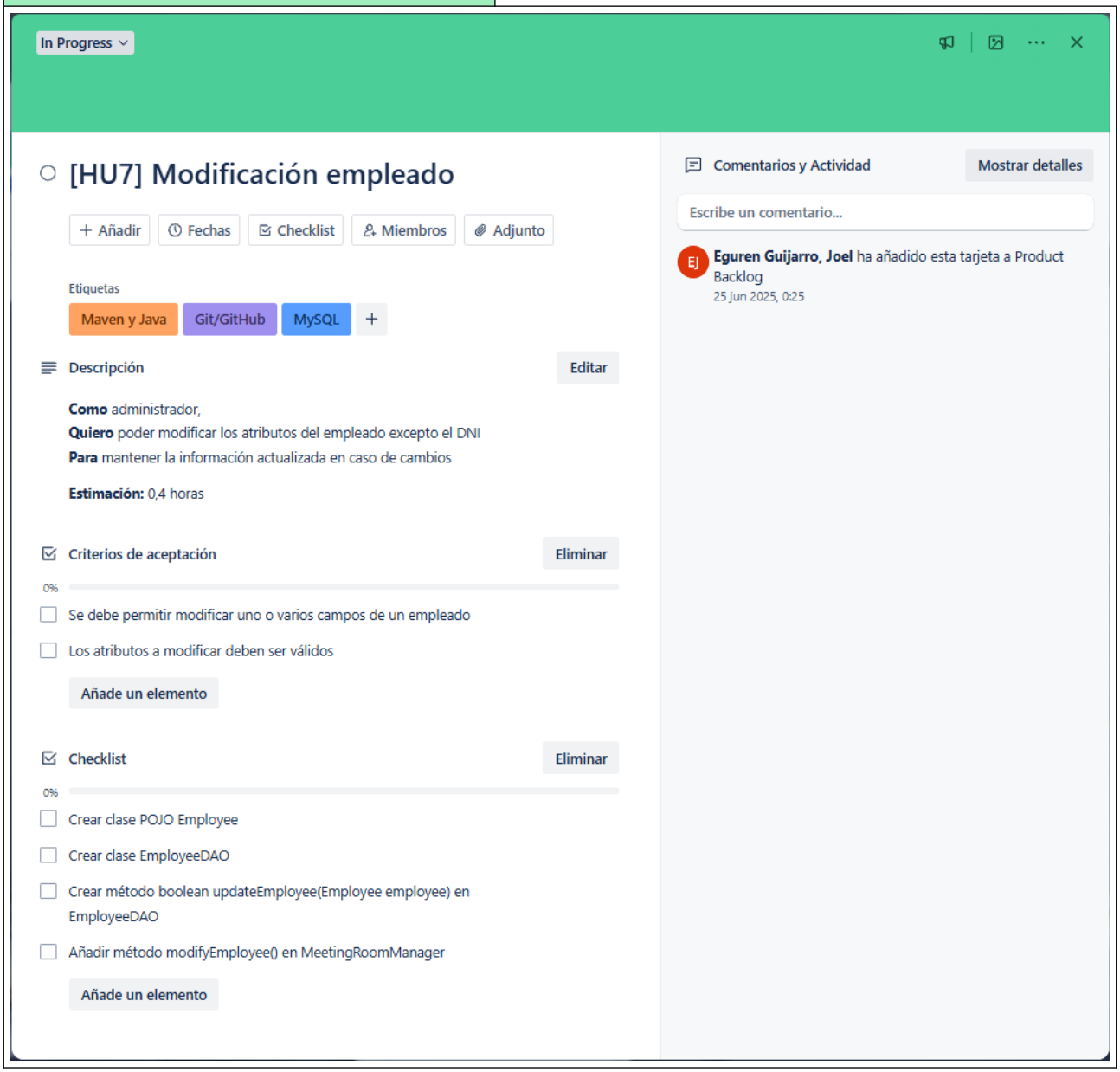
2.5. HU5 – Alta empleado



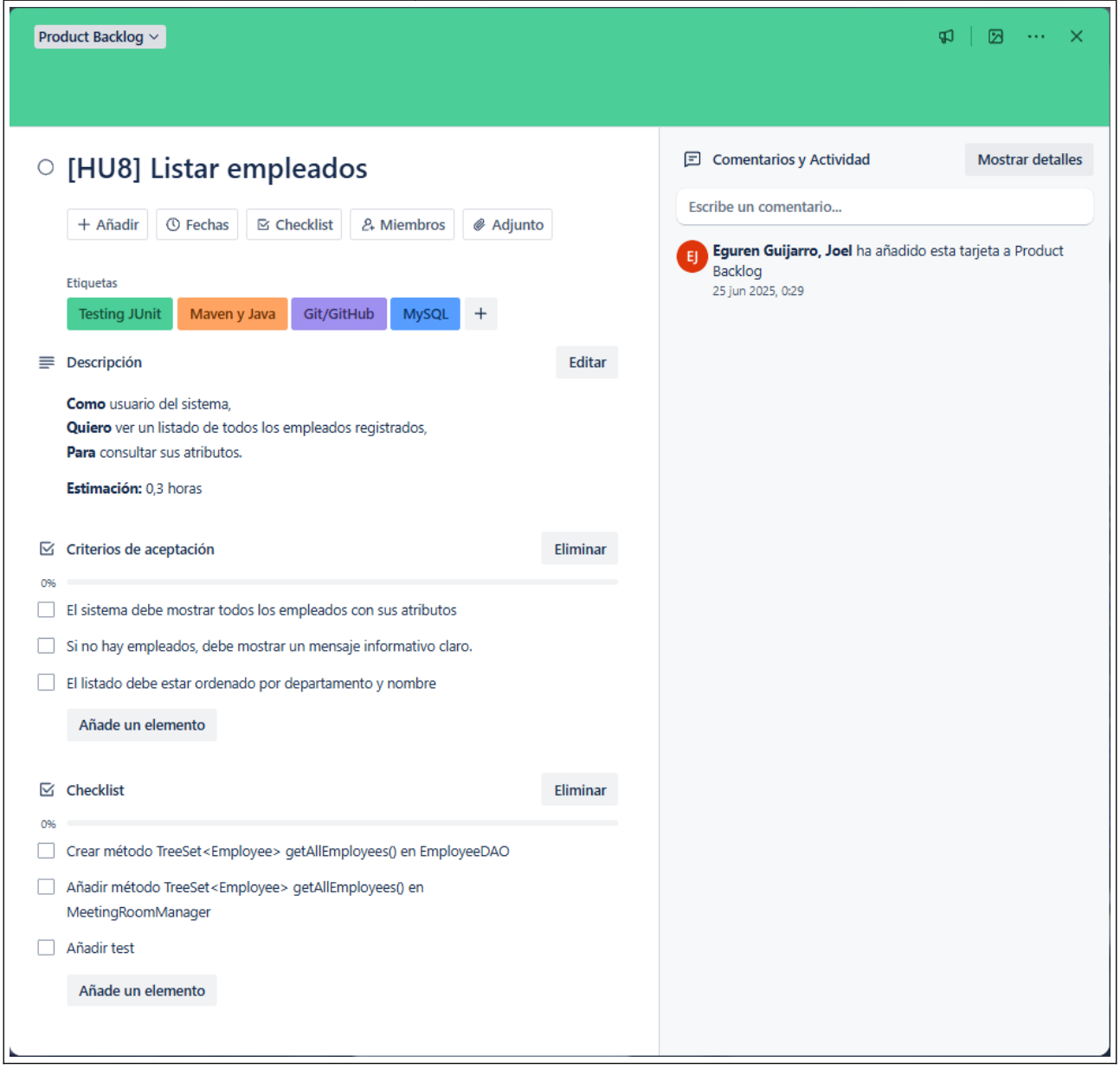
2.6. HU6 – Baja empleado



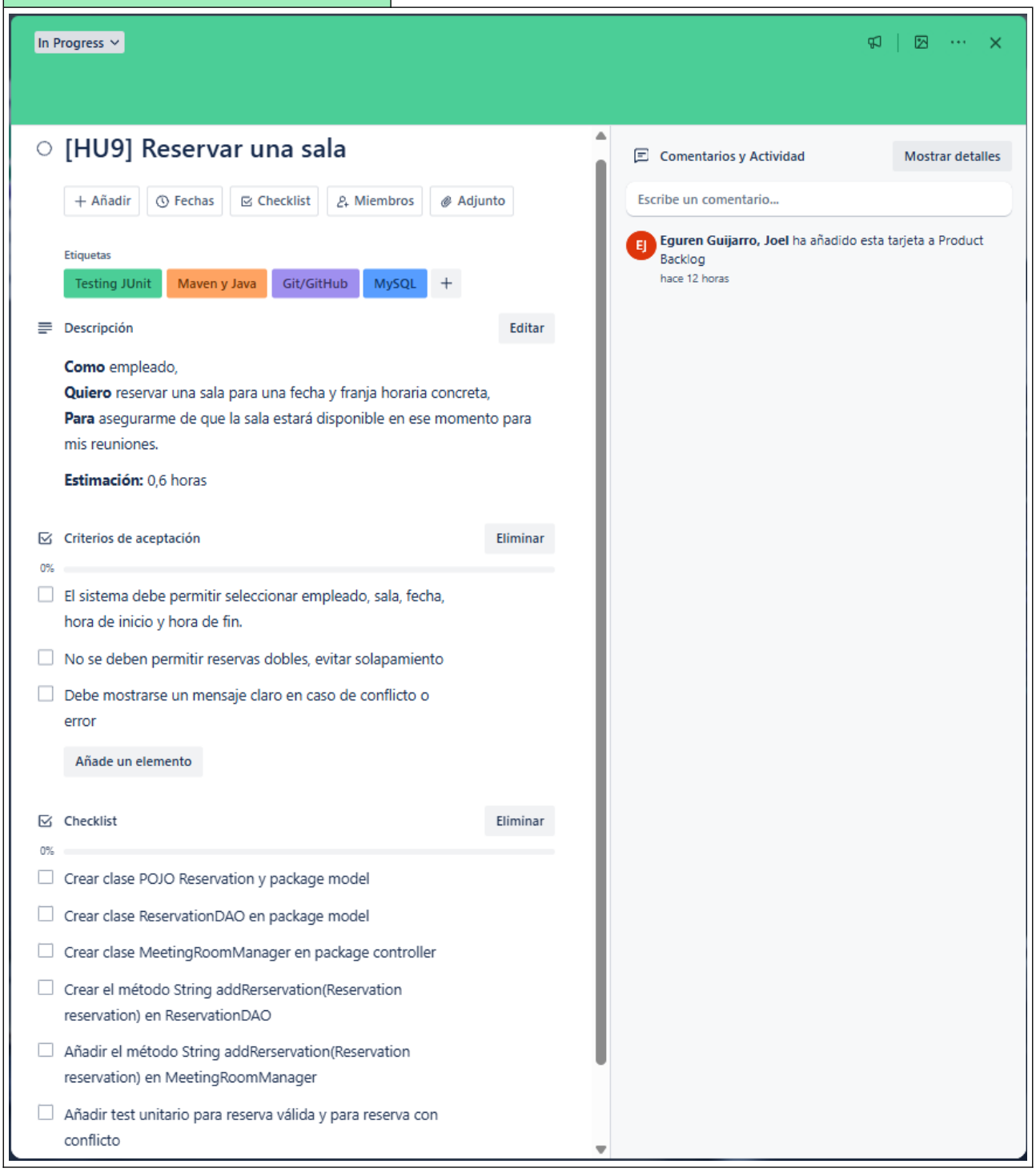
2.7. HU7 – Modificación empleado



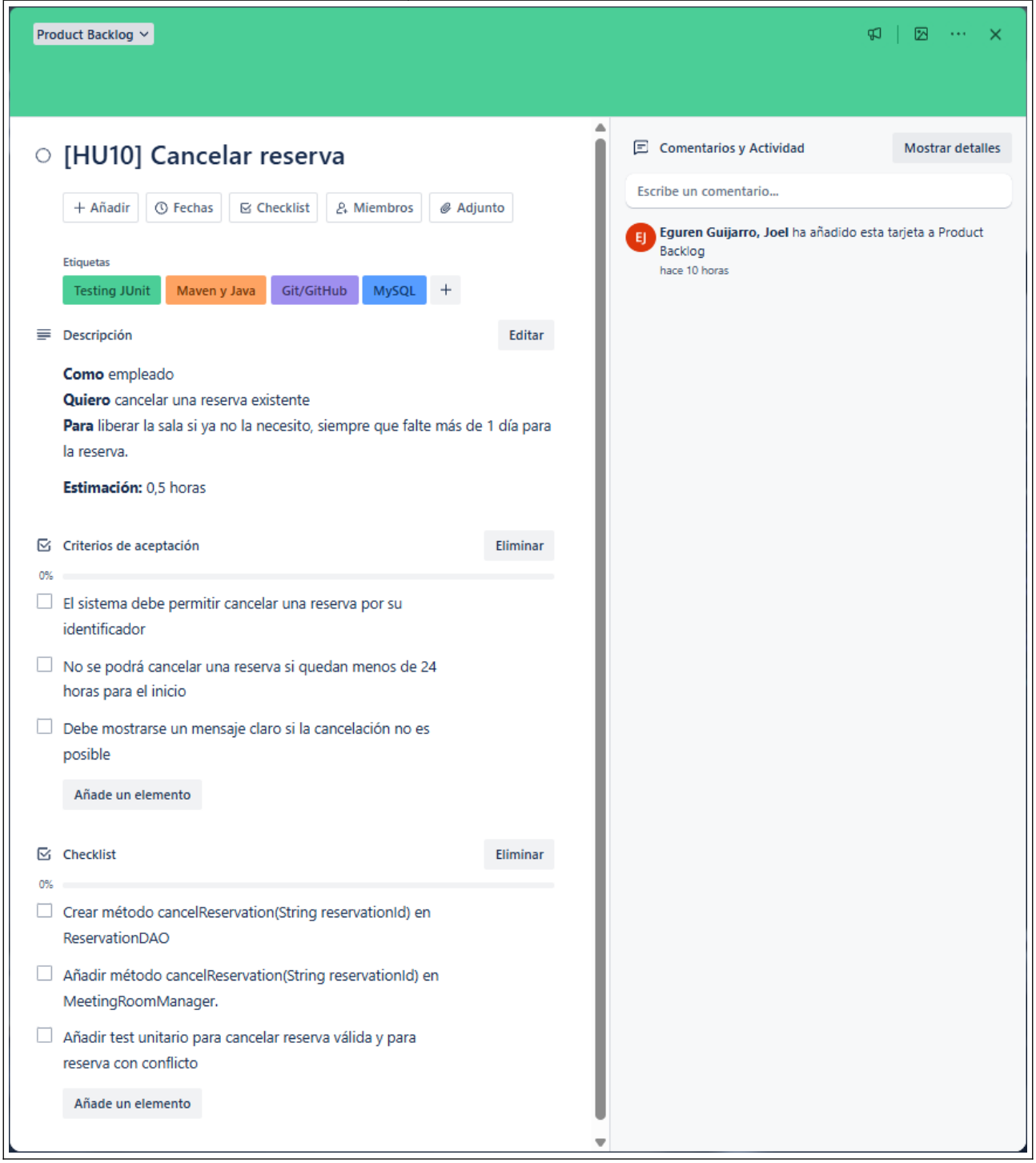
2.8. HU8 – Listar empleados



2.9. HU9 – Reservar una sala



1.10. HU10 – Cancelar reserva



3. Sprint 1

3.1. Objetivos y planificación

El Sprint 1 comprenderá el día dos y tres. Para este Sprint se han elegido las tarjetas e historias de usuario con mayor estimación, para dar un alto valor al proyecto teniendo en cuenta las dependencias técnicas. *(La estimación está en la descripción de cada tarjeta).*

Ordenados, los objetivos de este Sprint son:

- Diseñar modelo ER
- Crear script SQL y base de datos
- Inicializar repositorio Git/GitHub
- Configurar entorno para pruebas unitarias
- Crear y configurar proyecto Maven
- Configurar conexión a base de datos
- HU9 - Reservar una sala
- HU10 - Cancelar reserva
- HU3 - Modificación sala
- HU7 - Modificación empleado
- HU4 - Listar salas

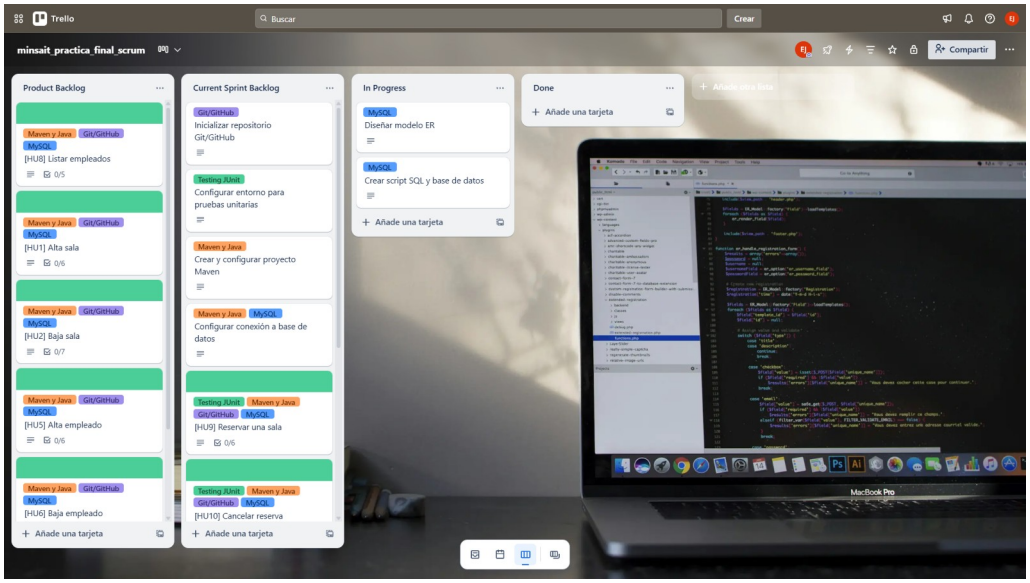


Figura 1. *Situación actual en Trello*

3.2. Reflexión

En este Sprint se han logrado realizar todas las tarjetas planificadas, excepto algún objetivo de la tarjeta *HU9 – Reservar una sala*, porque se ha necesitado crear tres tareas técnicas para verificar que una reserva es válida:

- En la clase **EmployeeDAO** crear el método Optional<*Employee*> *getEmployeeByDNI(String dni)*: para verificar que el empleado existe.
- En la clase **RoomDAO** crear el método *Optional<Room> getRoomById(int roomId)*: para verificar que la sala existe.
- En la clase **ReservationDAO** crear el método *boolean canReservate(Reservation reservation)*: para verificar que la fecha de la reserva es correcta y no solape con otra.

Se ha controlado correctamente el límite *Work In Progress* (WIP), lo cual ha facilitado centrarse en pocas tareas a la vez y mejorar la eficiencia en el desarrollo. Además, se pudo absorber y completar satisfactoriamente la *HU8 – Listar Empleados*.

Como mejora, se podrían optimizar la estimación de las tarjetas para evitar tiempos muertos entre sprints y controlar mejor posibles tareas técnicas que podrían surgir.

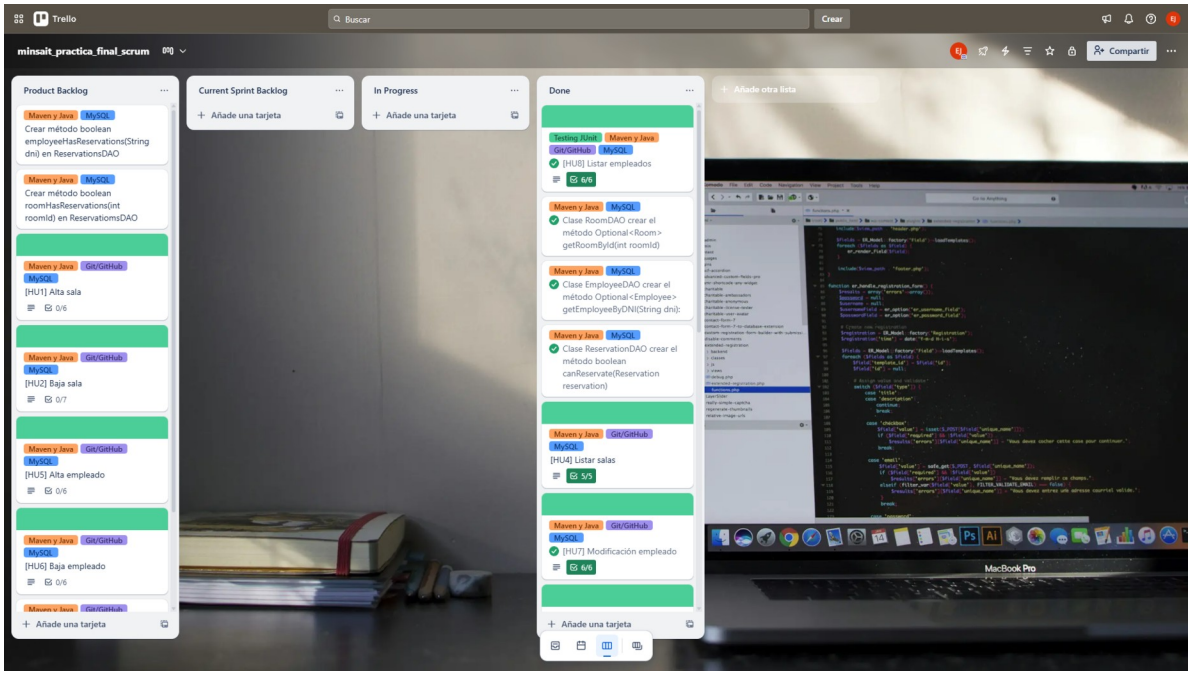


Figura 2. *Situación final Sprint en Trello*

4. Sprint 2

4.1. Objetivos y planificación

El Sprint 2 se ha planificado para el día cuatro del desarrollo. Para este Sprint se han elegido las tarjetas e historias restantes ordenadas de mayor a menor estimación y teniendo en cuenta las dependencias técnicas. *(La estimación está en la descripción de cada tarjeta).*

Ordenados, los objetivos de este Sprint son:

- Crear método *boolean employeeHasReservations(String dni)* en ReservationsDAO
- Crear método *boolean roomHasReservations(int roomId)* en ReservatiomsDAO
- [HU1] - Alta sala
- [HU2] - Baja sala
- [HU5] - Alta empleado
- [HU6] - Baja empleado
- Añadir tests faltantes de verificación
- Crear programa principal Main
- Crear README.md

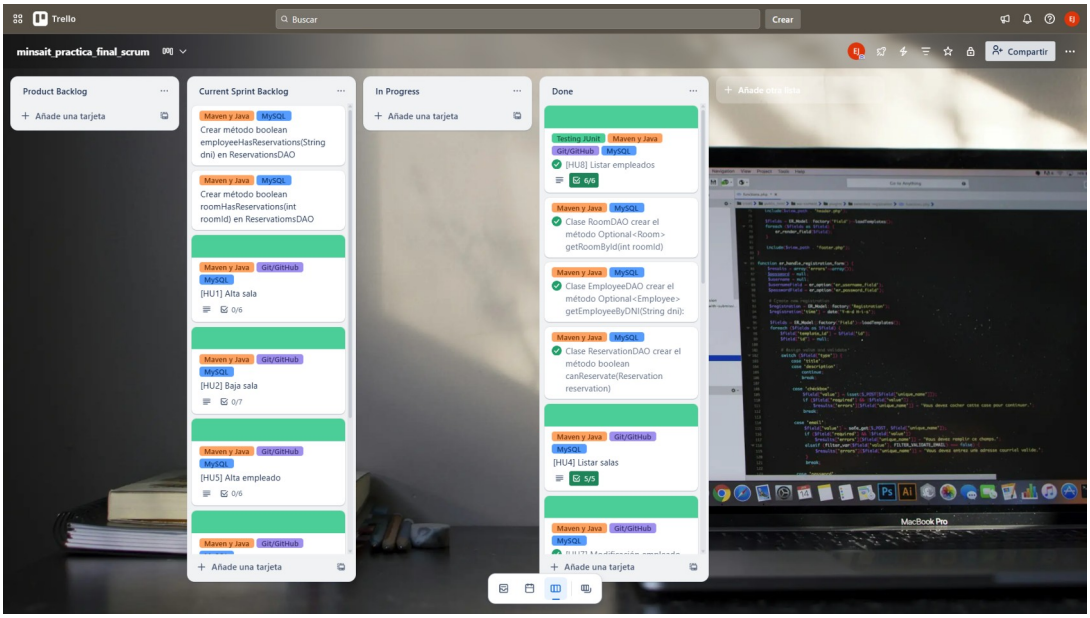


Figura 3. *Situación actual en Trello*

4.2. Reflexión

En este Sprint se han logrado realizar todas las tarjetas planificadas satisfactoriamente, pero se ha necesitado crear dos tareas técnicas para verificar si una sala es válida y una empleado es válido:

- En la clase **MeetingRoomManager** crear el método *boolean isValidRoom(Room room)*: para verificar que la sala apta para ser introducida en la BD.
- En la clase **MeetingRoomManager** crear el método *boolean isValidEmployee(Employee employee)*: para verificar que el empleado es apto para ser introducido en la BD.

También se ha controlado correctamente el límite *Work In Progress* (WIP), lo cual ha facilitado centrarse en pocas tareas a la vez y mejorar la eficiencia en el desarrollo.

Como mejora, se podría haber dividido la clase **MeetingRoomMain** en varias clases auxiliares, con el objetivo de repartir mejor las responsabilidades y favorecer una mayor modularidad y mantenibilidad del código.

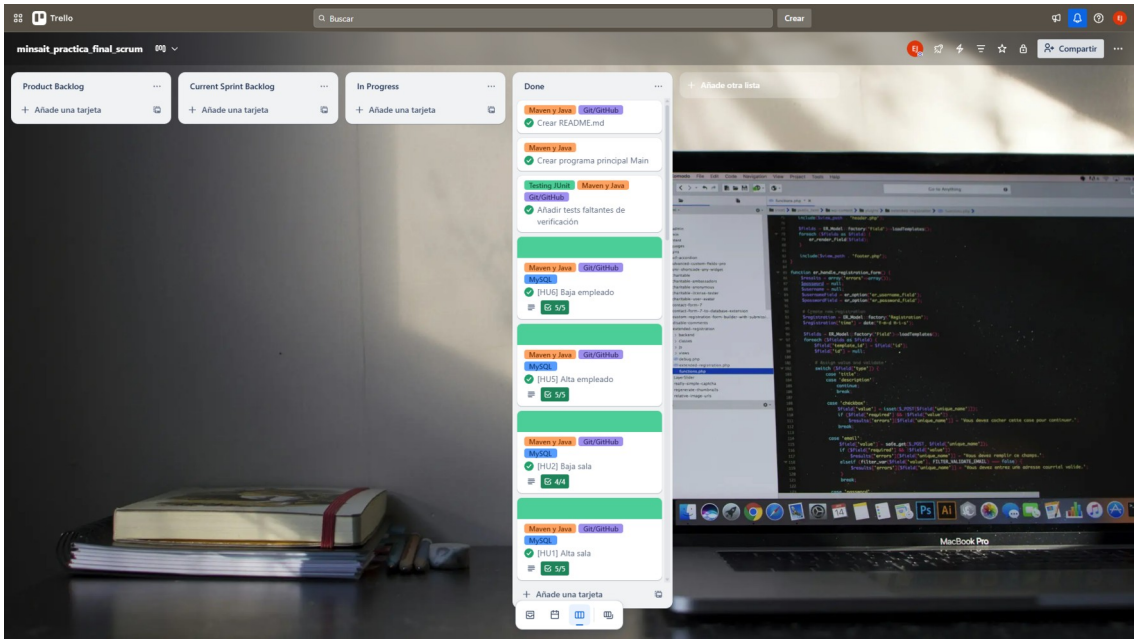


Figura 4. *Situación final Sprint en Trello*

5. Retrospectiva final

Los objetivos de este proyecto han sido completados como se esperaba. Se han logrado alcanzar todos los objetivos de los sprints, incluyendo las historias de usuario, las tareas técnicas que se planificaron y alguna tarea que ha surgido durante el desarrollo.

Durante el trascurso de este proyecto he reforzado habilidades relacionadas con la importancia de una buena planificación y organización, así como de mantener un flujo de trabajo constante. A nivel técnico he mejorado mi conocimiento en las tecnologías trabajadas como Java, MySQL, Maven y JUnit. He procurado aplicar principios de *clean code,* para mejorar la legibilidad y mantenibilidad del código y he documentado cada commit en Git de forma clara y coherente.

Como posibles dificultades durante el desarrollo, he enfrentado la gestión y creación de todas las tarjetas y historias de usuario, decidiendo su estimación y prioridad. Además, he diseñado toda la estructura de código aplicando MVC y cumpliendo con los requisitos del enunciado. Otra dificultad podría ser el corto plazo de entrega, pero finalmente se ha podido cumplir.

En resumen, este proyecto ha sido una experiencia muy enriquecedora que me ha permitido conocer y aplicar de manera práctica la metodología Scrum y mejorar en las tecnologías usadas.