|  |
| --- |
| **Práctica Final de Curso:**  Desarrollo de una Aplicación  Java con Scrum y MySQL |

**Joel Eguren Guijarro**

|  |
| --- |
| **Contenido** |

[1. Introducción 2](#__RefHeading___Toc2605_260722825_Copia_1)

[2. Historias de usuario 3](#__RefHeading___Toc2605_260722825_Copia_2)

[3. Sprint 1 3](#__RefHeading___Toc2605_260722825_Copia_1)

[1.1. Configurar IP estàtica 3](#__RefHeading___Toc19390_2060922764_Copia)

[4. Sprint 2 4](#__RefHeading___Toc2605_260722825_Copia_3)

[2.1. Instal·lar isc-dhcp-server 4](#__RefHeading___Toc19390_2060922764_Copi1)

|  |
| --- |
| 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.

|  |
| --- |
| 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.

|  |  |
| --- | --- |
| **HU1 – Alta sala** |  |
|  | |

|  |  |
| --- | --- |
| **HU2 – Baja sala** |  |
|  | |

|  |  |
| --- | --- |
| **HU3 – Modificación sala** |  |
|  | |

|  |  |
| --- | --- |
| **HU4 – Listar salas** |  |
|  | |

|  |  |
| --- | --- |
| **HU5 – Alta empleado** |  |
|  | |

|  |  |
| --- | --- |
| **HU6 – Baja empleado** |  |
|  | |

|  |  |
| --- | --- |
| **HU7 – Modificación empleado** |  |
|  | |

|  |  |
| --- | --- |
| **HU8 – Listar empleados** |  |
|  | |

|  |  |
| --- | --- |
| **HU9 – Reservar una sala** |  |
|  | |

|  |  |
| --- | --- |
| **HU10 – Cancelar reserva** |  |
|  | |

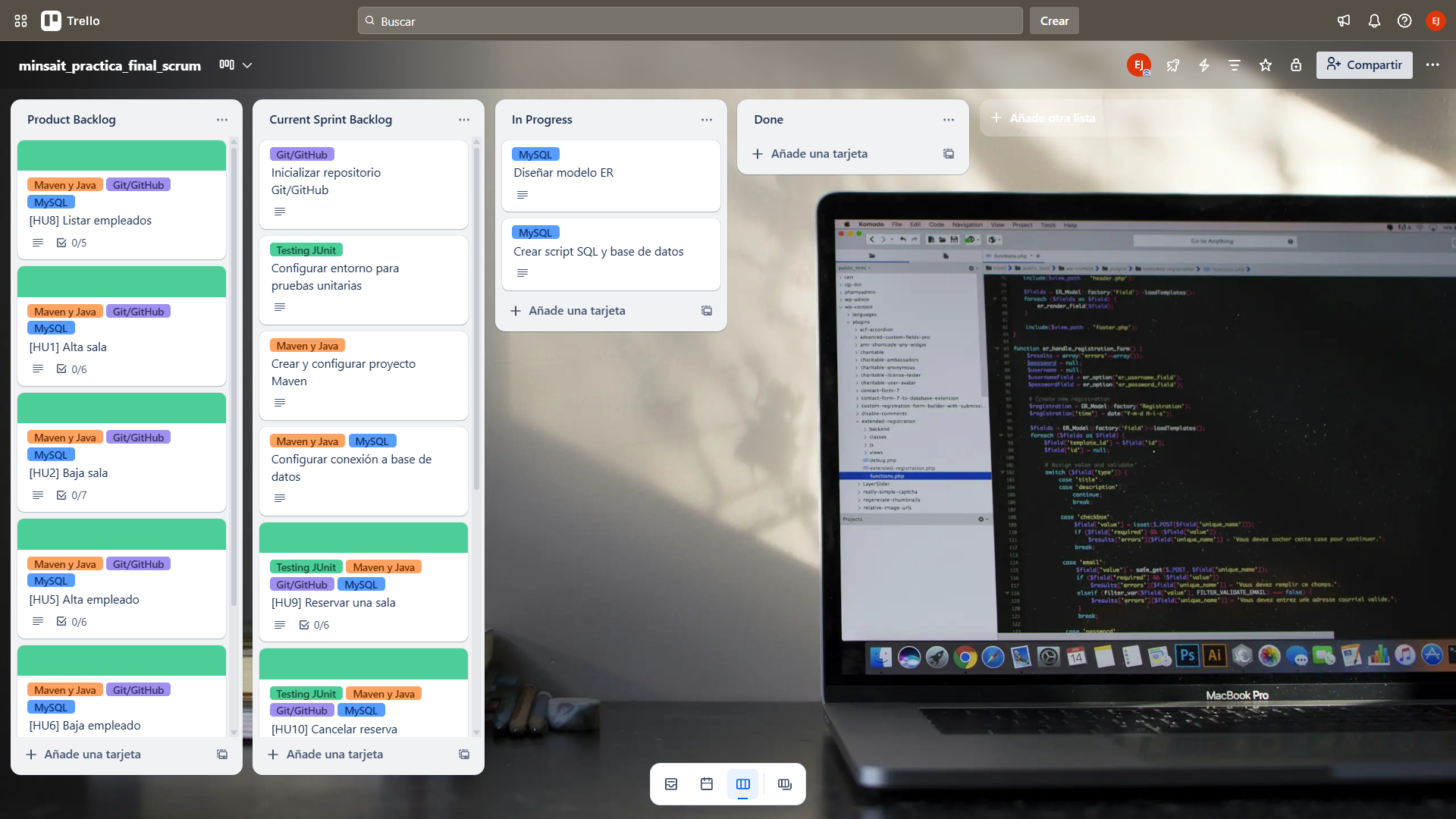
|  |
| --- |
| 3. Sprint 1 |

|  |
| --- |
| 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](#_toc378)
* [HU10 - Cancelar reserva](#_toc395)
* [HU3 - Modificación sala](#_toc248)
* [HU7 - Modificación empleado](#_toc332)
* [HU4 - Listar salas](#_toc270)

****

***Figura 1.*** *Situación actual en Trello*

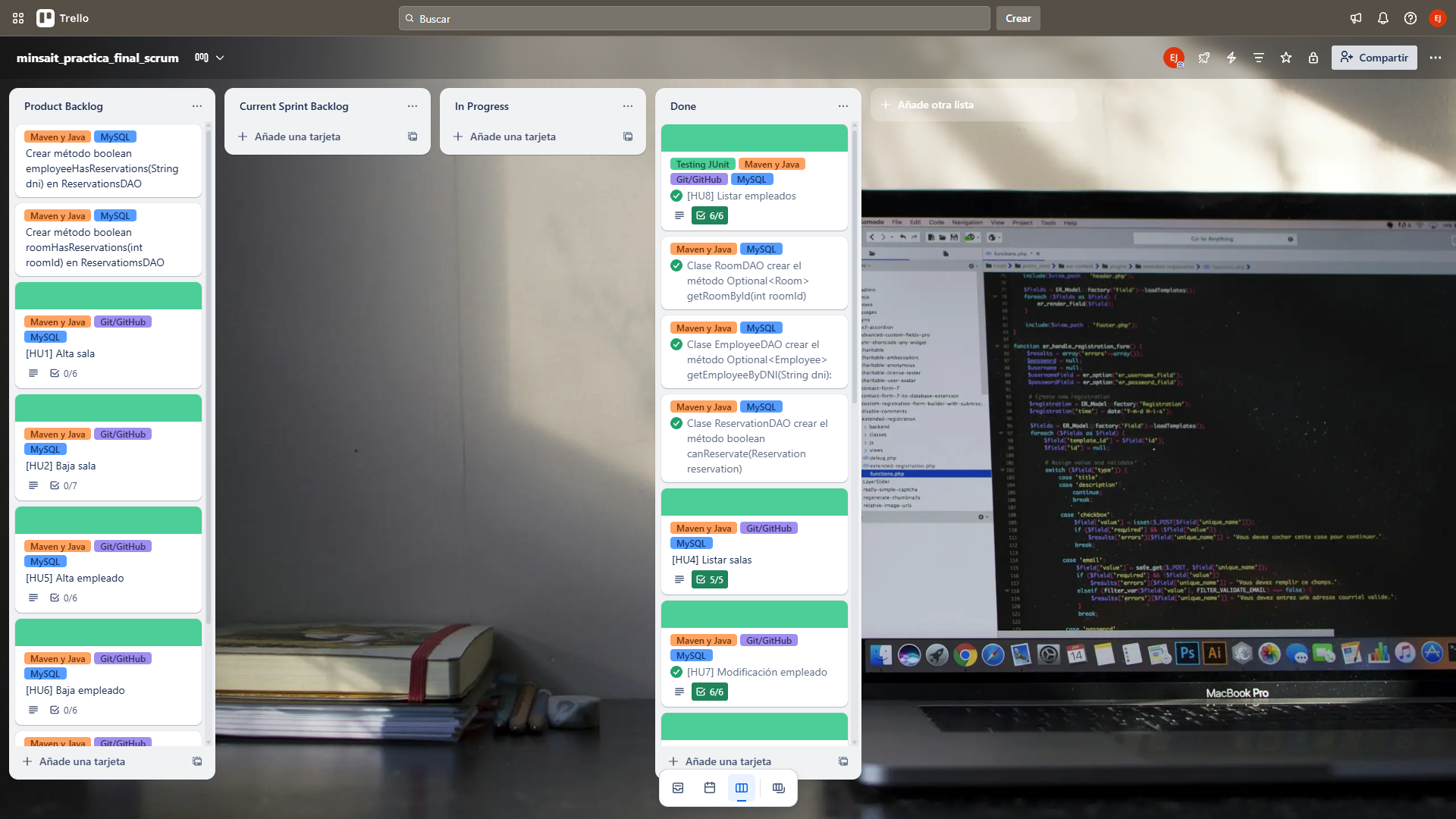
|  |
| --- |
| 3.2. Reflexión |

En este Sprint se han logrado realizar todas las tarjetas planificadas, excepto algún objetivo de la tarjeta *HU9 – Reservar una sala*, porqué 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*](#_toc355).

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*

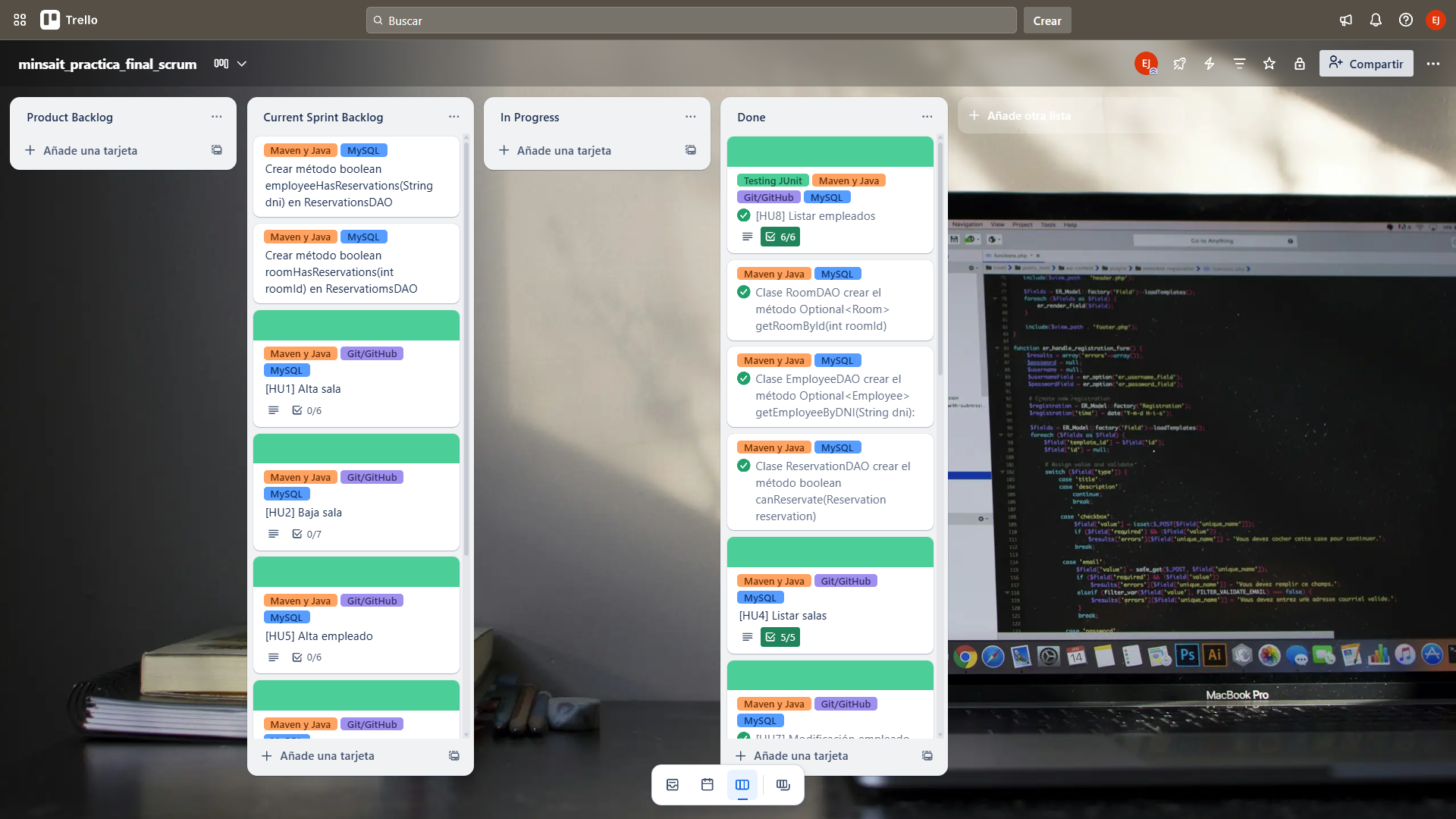
|  |
| --- |
| 4. Sprint 2 |

|  |
| --- |
| 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](#_toc215)
* [[HU2] - Baja sala](#_toc228)
* [[HU5] - Alta empleado](#_toc292)
* [[HU6] - Baja empleado](#_toc314)
* Añadir tests faltantes de verificación
* Crear programa principal Main
* Crear README.md

****

***Figura 1.*** *Situación actual en Trello*

|  |
| --- |
| 4.2. Reflexión |