**Gestor de Reserva De Salas**

**Sprint 1**

**Objetivo:**

* Crear Estructura de Base de Datos para todos los registros de la aplicación.
* Entregar aplicación funcional para poder empezar registrando los empleados y las salas con todas sus opciones de CRUD, con todas sus implementaciones de métodos de negocio (altas, bajas modificaciones y listas).
* Realizar test de verificación del correcto funcionamiento de lo que se entregara.
* Primera versión del menú de opciones de la aplicación.

**Planificación:**

**Historia de Usuario 1: Gestión Mediante menú**

* **Como** administrador
* **Quiero** gestionar todo el proceso de empleados, salas y reservas mediante un menú de opciones.
* **Para** que poder gestionar todo de manera sencilla.

**Tareas:**

1. Crear la clase menú con todas las opciones: CRUD empleados, salas y reservas.
2. Crear base de datos y clases necesarias.

**Historia de Usuario 2: Gestión de Empleados**

* **Como** administrador
* **Quiero** gestionar la información de los empleados
* **Para** que puedan realizar reservas de salas.

**Tareas:**

1. Crear la clase Empleado con atributos: DNI, nombre, apellidos, email y departamento.
2. Implementar métodos CRUD para empleados en el servicio GestorEmpleados.
3. Realizar pruebas unitarias para la gestión de empleados.

**Historia de Usuario 3: Gestionar Sala**

* **Como** administrador
* **Quiero** poder gestionar salas de reuniones
* **Para** poder dar de alta, modificar, eliminar o ver las salas para reuniones.

**Tareas:**

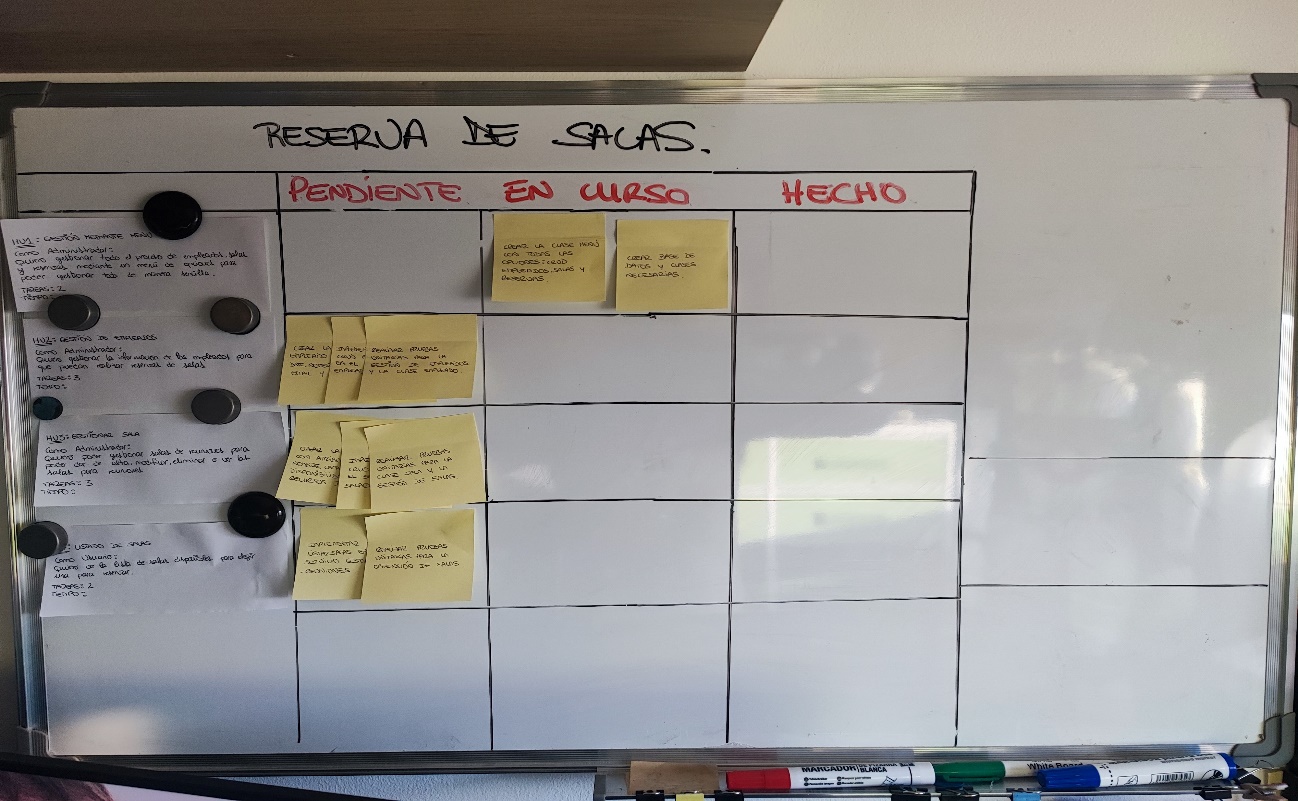
1. Crear la clase Sala con atributos: id, nombre, capacidad, disponibilidad y recursosDisponibles.
2. Implementar métodos CRUD para Salas en el servicio GestorSalaReuniones.
3. Realizar pruebas unitarias para la gestión de salas.

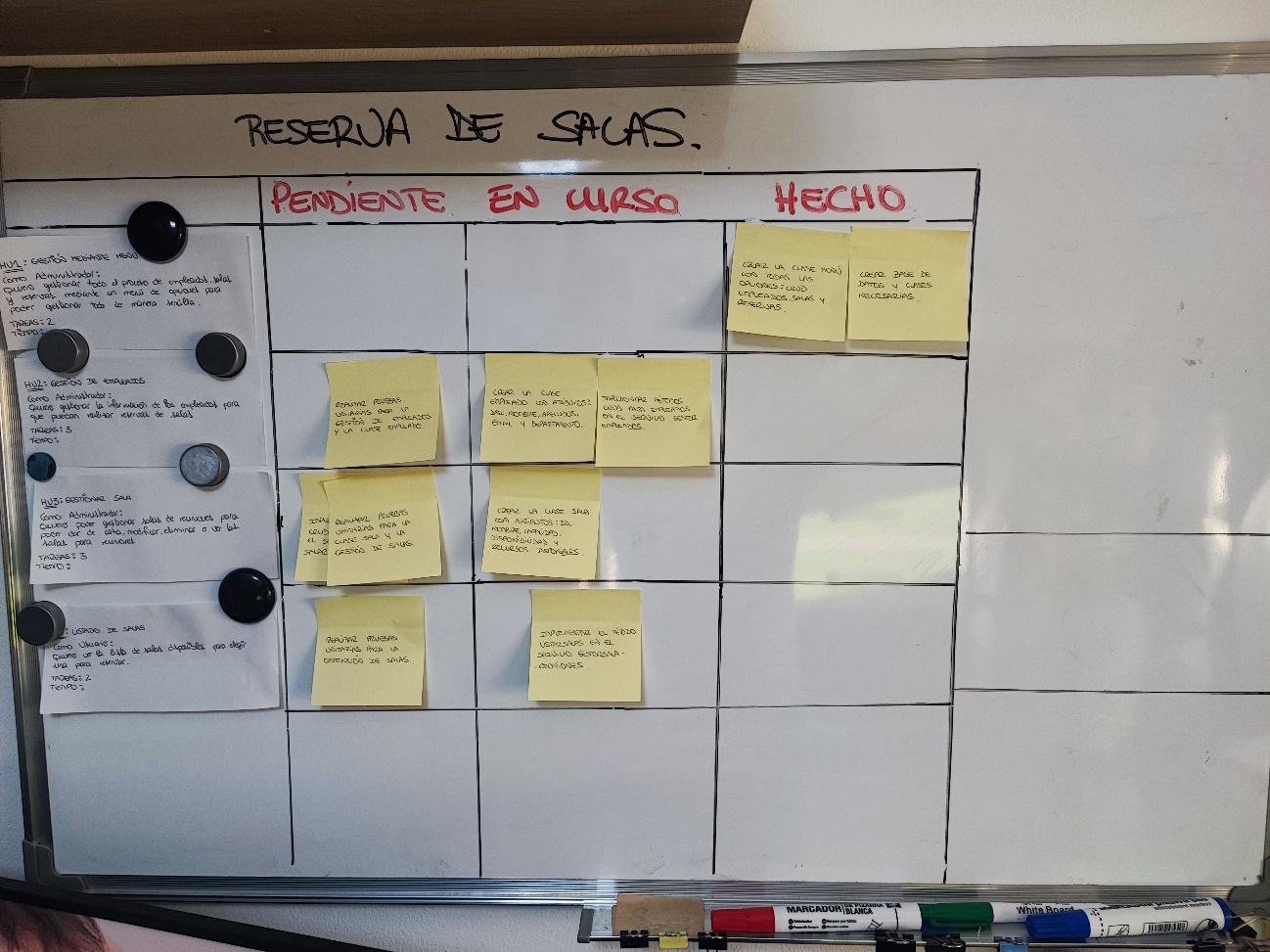
**Historia de Usuario 4: Listado de Salas**

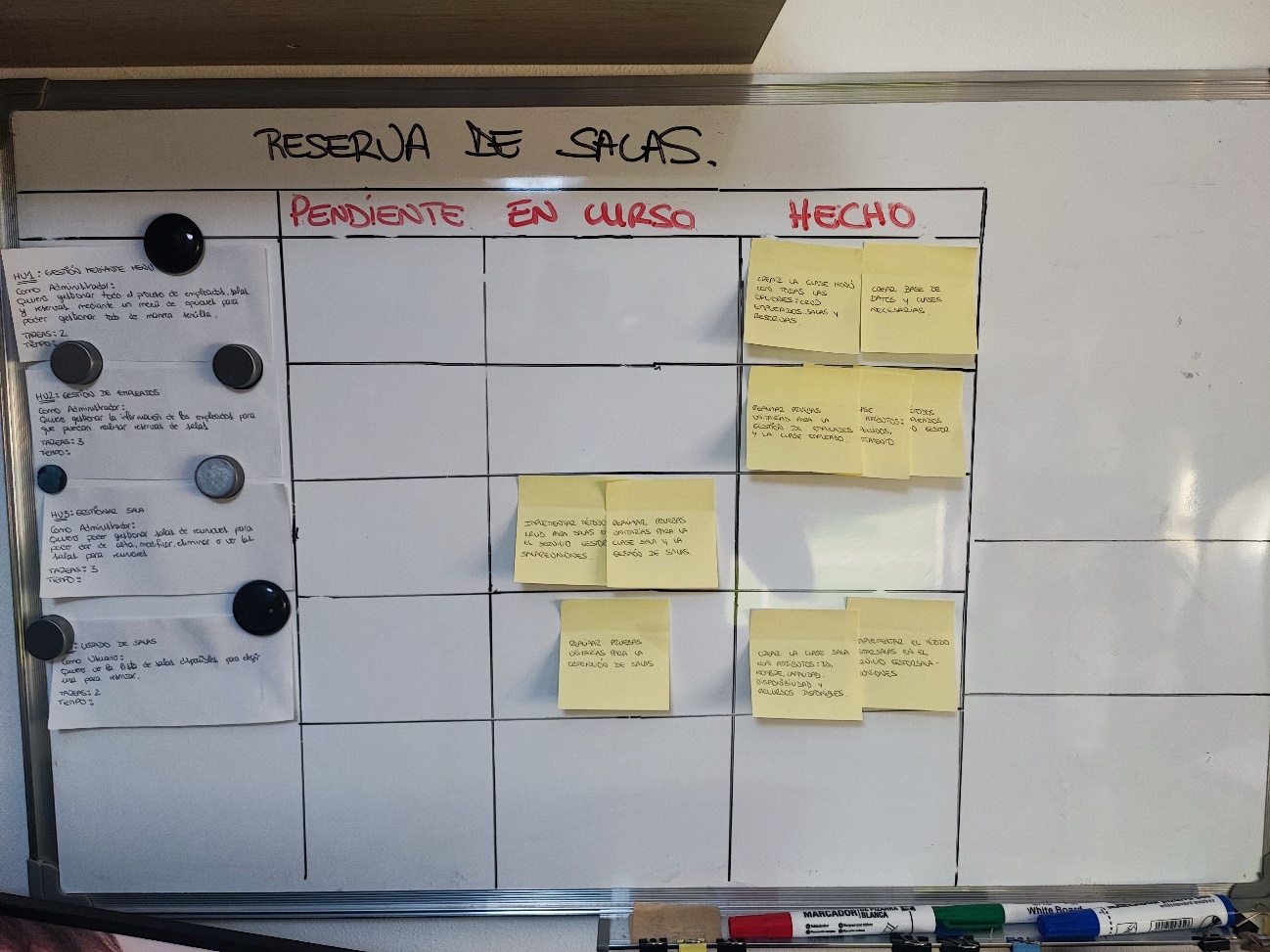
* **Como** usuario
* **Quiero** ver la lista de salas disponibles
* **Para** elegir una para reservar.

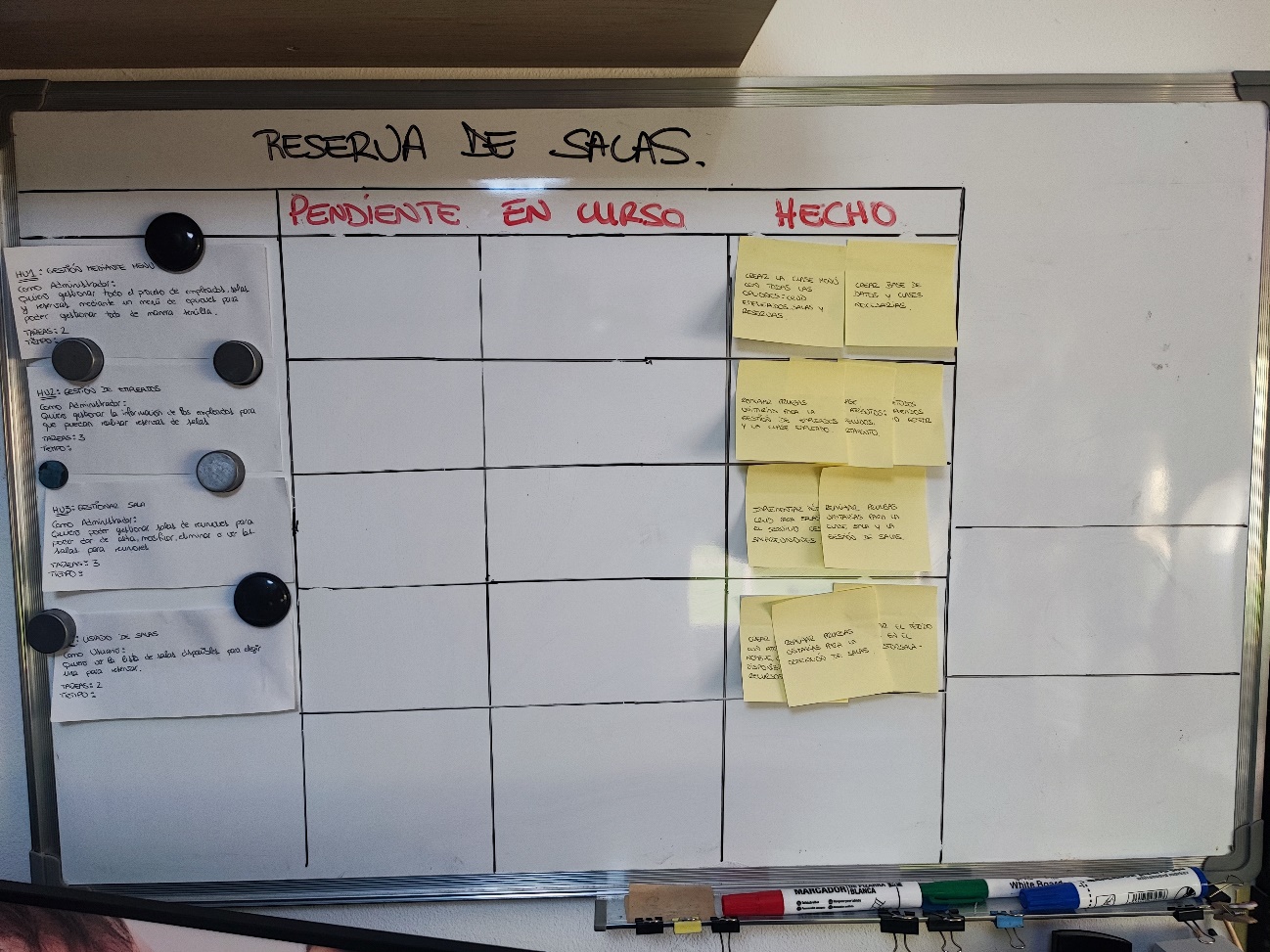
**Tareas:**

1. Implementar el método listarSalas en el servicio GestorSalaReuniones.
2. Realizar pruebas unitarias para la obtención de salas.

**Progresión Tablero en este Sprint:**

****

****

****

**Reflexión:**

* Se completo lo esperado con tiempo suficiente.
* Hace falta más comunicación con el cliente después de cada Historia de Usuario para poder verificar si la aplicación de las mismas es correcta.
* Realización de pruebas reales con la base de datos.

**Sprint 2**

**Objetivo:**

* Implementar todos los métodos de negocio relacionados con la reserva (alta, cancelación, modificación y lista).
* Controlar los posibles conflictos de reserva entre las salas horarios y fechas para evitar duplicados de reserva en la misma sala.
* Realización de test para comprobar el correcto funcionamiento de las reservas.
* Entregar aplicación final funcional para poder realizar lo acordado con el cliente.

**Planificación:**

**Historia de Usuario 5: Reserva de Sala**

* **Como** empleado
* **Quiero** poder reservar una sala para una fecha y hora específicas
* **Para** asegurarme de que la sala esté disponible.

**Tareas:**

1. Crear la clase Reserva con atributos: idReserva, empleado, sala, fecha, horaInicio, horaFin.
2. Implementar el método altaReservaen el servicio de GestorReserva.
3. Realizar pruebas unitarias para la creación de reservas.

**Historia de Usuario 6: Control de Conflictos de Horario**

* **Como** sistema
* **Quiero** verificar conflictos de horario
* **Para** evitar reservas dobles en la misma sala.

**Tareas:**

1. Implementar lógica en el método altaReserva para comprobar disponibilidad.
2. Crear una respuesta adecuada para conflictos de horario.
3. Realizar pruebas unitarias para la validación de conflictos.

**Historia de Usuario 7: Listar Reservas**

* **Como** administrador
* **Quiero** ver todas las reservas realizadas
* **Para** gestionar el uso de las salas.

**Tareas:**

1. Implementar el método listarReservas en el servicio GestorReserva.
2. Realizar pruebas unitarias para la obtención de reservas.

**Historia de Usuario 8: Actualizar Reserva**

* **Como** empleado
* **Quiero** poder actualizar una reserva
* **Para** cambiar la fecha o sala si es necesario.

**Tareas:**

1. Implementar el método modificarReservaen el servicio GestorReserva.
2. Realizar pruebas unitarias para la actualización de reservas.

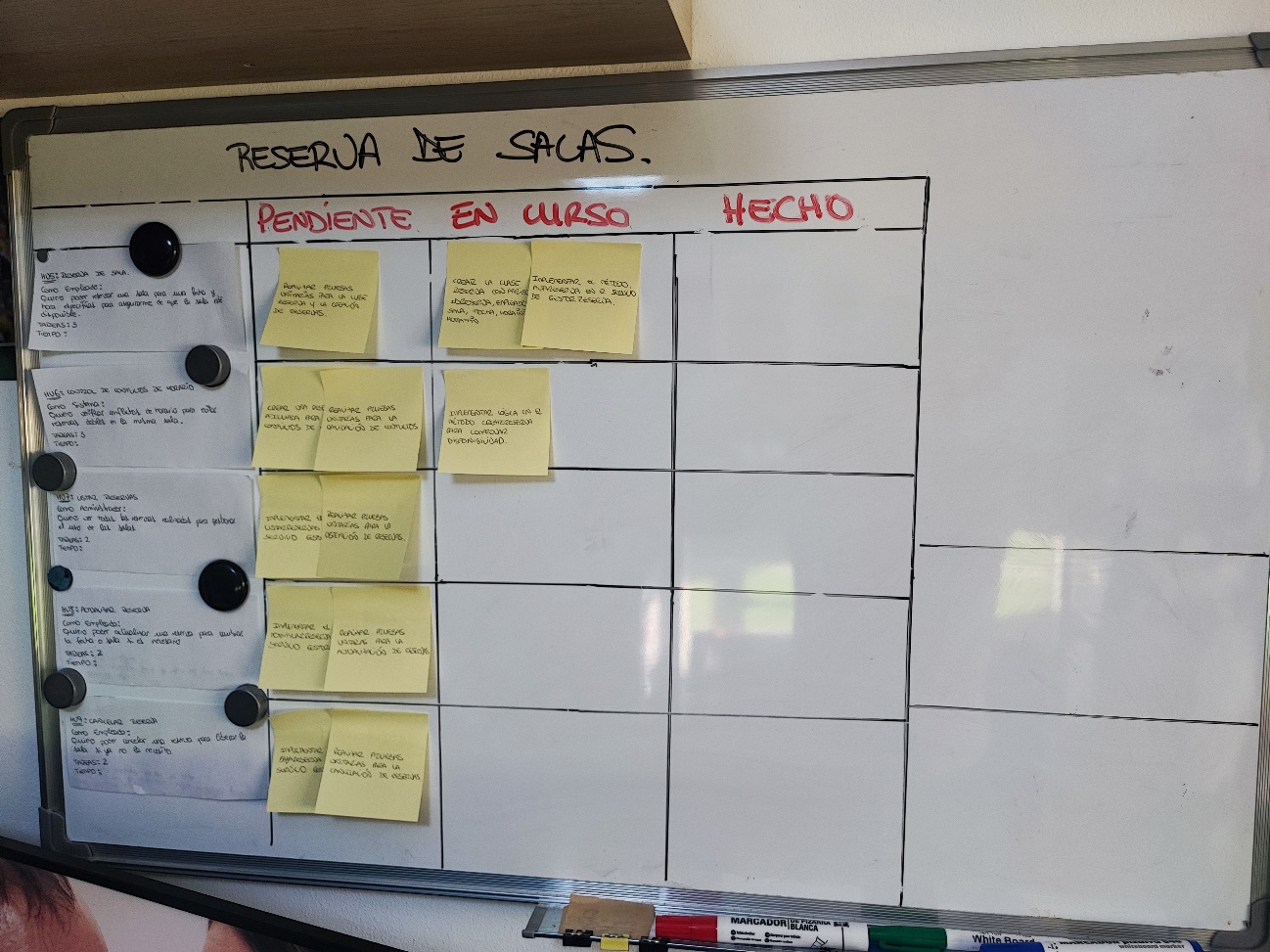
**Historia de Usuario 9: Cancelar Reserva**

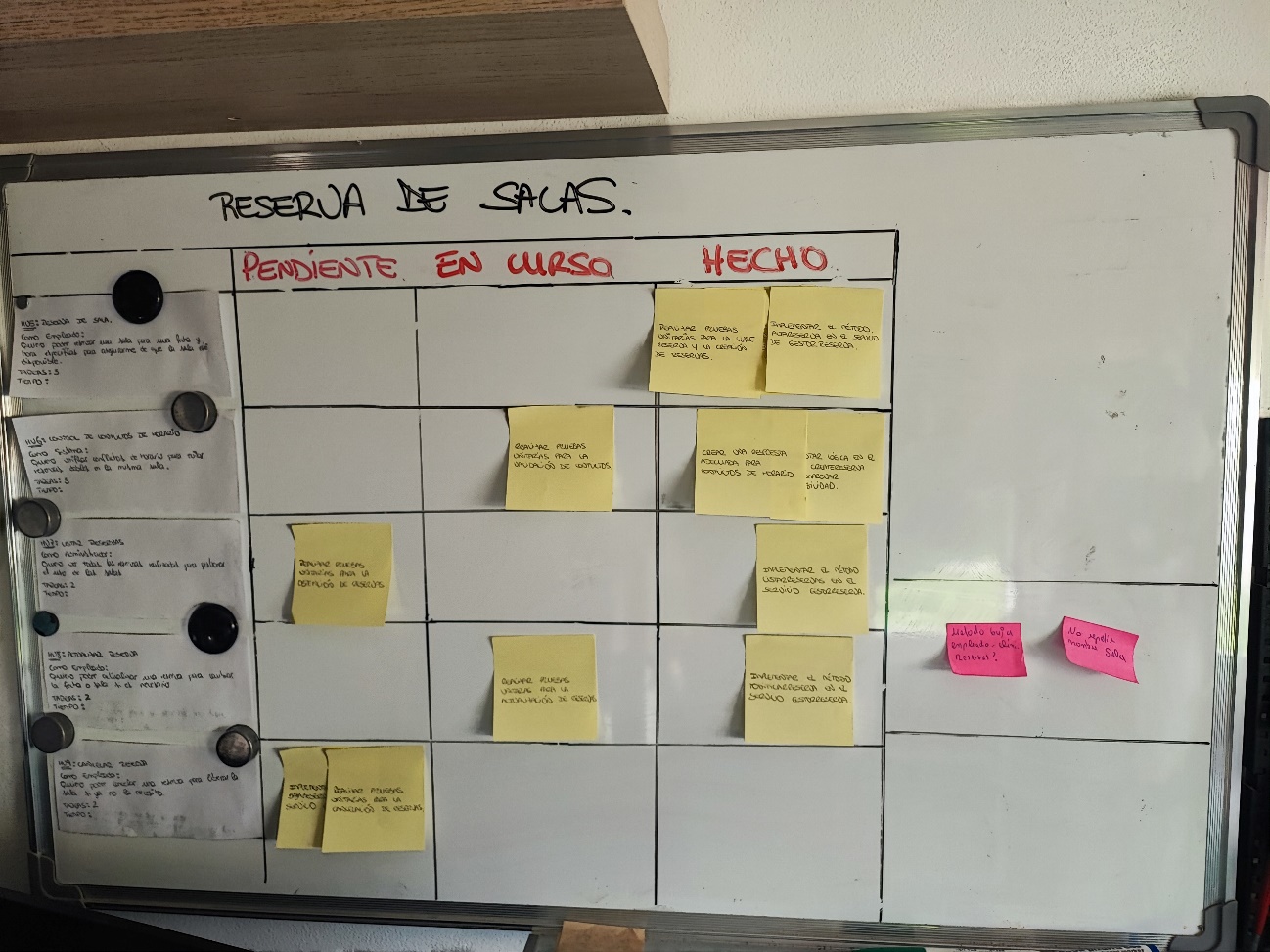
* **Como** empleado
* **Quiero** poder cancelar una reserva
* **Para** liberar la sala si ya no la necesito.

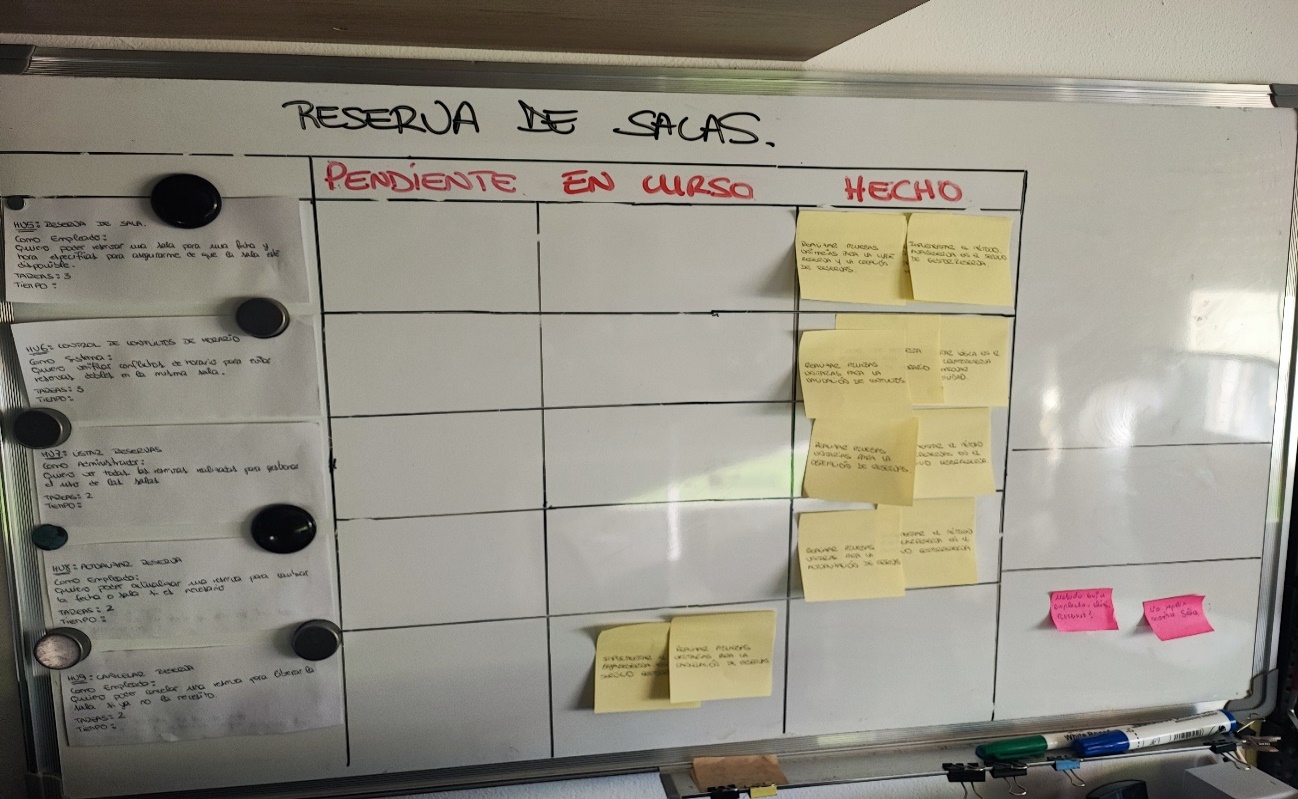
**Tareas:**

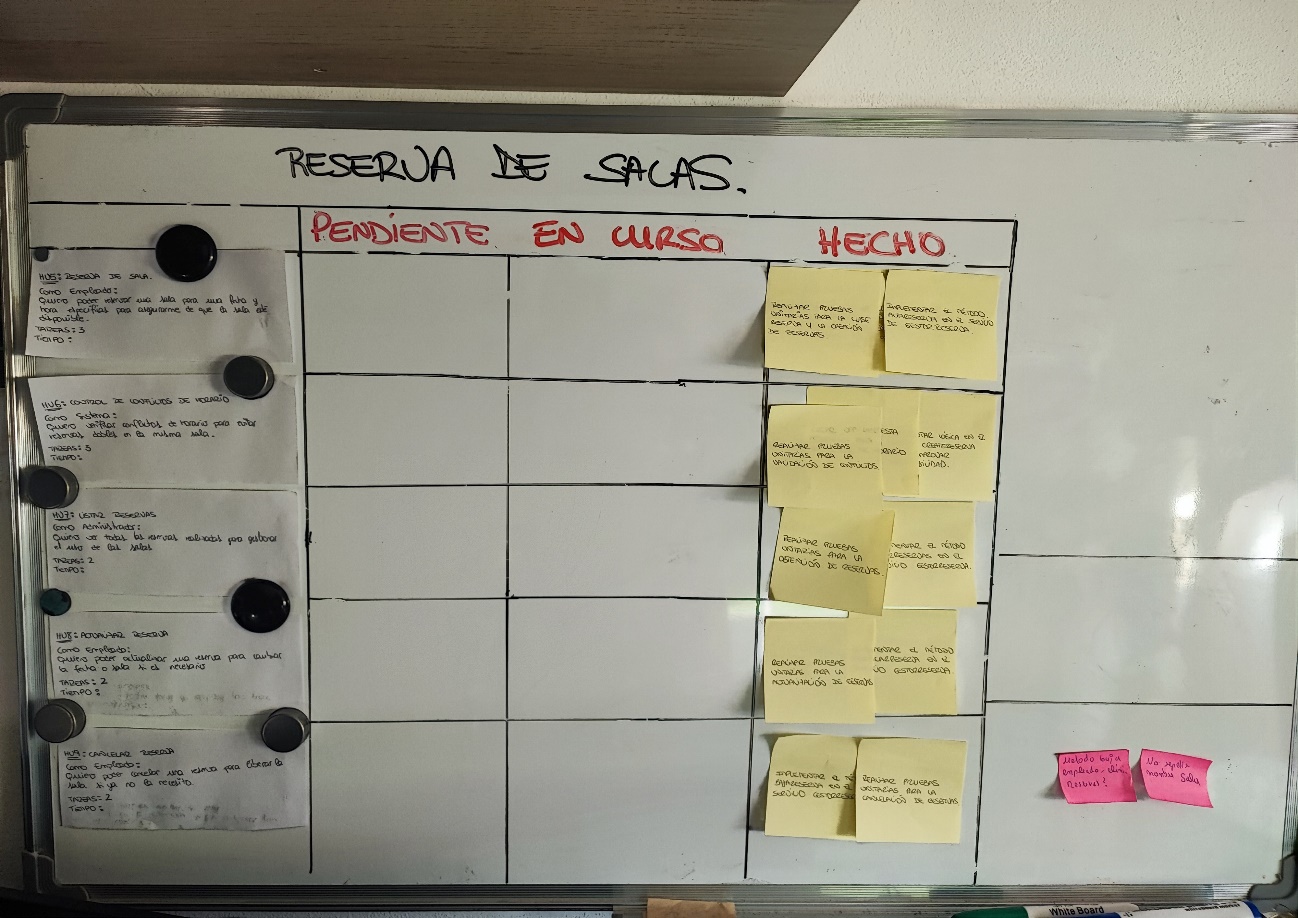
1. Implementar el método cancelacionReserva en el servicio GestorReserva.
2. Realizar pruebas unitarias para la cancelación de reservas.

**Progresión Tablero en este Sprint:**

****

****

****

****

**Reflexión:**

* En este Sprint aunque se termino lo esperado a tiempo se fue un poco a contrareloj, hubieramos necesitado mas tiempo para poder evitar posibles fallos.
* Revisión de la documentación creada.
* Realización de más test o pruebas unitarias.