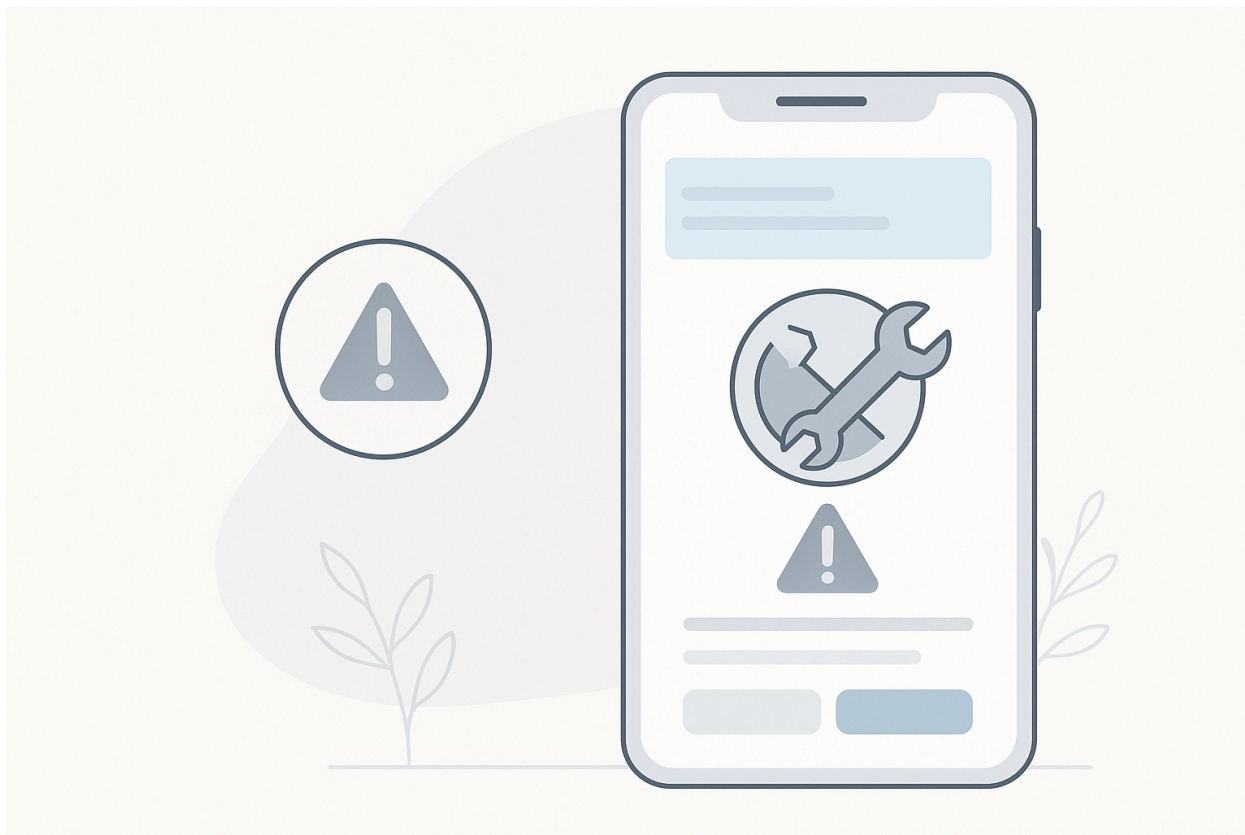


Sprint 1



Autor: Óscar Coronado Domínguez

2º DAM

Fecha : 28 nov 2025

Índice

Introducción.....	3
Objetivos y alcance del sistema.....	3
Objetivo general.....	3
Alcance funcional del sistema.....	4
Fuera de alcance.....	4
Definición de requisitos.....	5
Requisitos Funcionales (RF).....	5
Requisitos No Funcionales (RNF).....	6
Análisis de roles y actores.....	7
Diagrama de casos de uso (UML).....	8
Bocetos de Interfaz.....	9
Plan de Pruebas Inicial.....	10
Tipos de pruebas previstas.....	10
Pruebas unitarias (JUnit).....	10
Pruebas de integración (Postman).....	10
Pruebas de aceptación (funcionales).....	11
Objetivos del plan de pruebas.....	11
Casos de prueba iniciales.....	12

Introducción

Este documento trata sobre el Sprint 1 del proyecto Gestión de desperfectos por aulas.

En esta primera fase se lleva a cabo el análisis inicial del sistema, definiendo sus objetivos, alcance funcional, requisitos y actores principales. Además, se establecen los primeros elementos de diseño mediante diagramas UML y bocetos de la interfaz, formando la base conceptual sobre la que se desarrollarán los sprints posteriores.

El propósito de este sprint es que el proyecto parte de una visión clara y ordenada, permitiendo entender qué problema se pretende resolver, cómo se comportará el sistema. Toda esta información quedará recogida en este documento, que actuará como guía técnica para la implementación del backend, frontend y las pruebas del sistema en las siguientes fases.

Objetivos y alcance del sistema

Objetivo general

El **sistema Gestión de desperfectos** por aulas tiene como finalidad ofrecer una herramienta interna para que el profesorado y el personal del centro educativo **puedan comunicar, registrar y hacer seguimiento de los desperfectos** detectados en las instalaciones (aulas, talleres, laboratorios, pasillos, etc.).

A su vez, permite que el equipo responsable de mantenimiento pueda revisar, validar o denegar estas incidencias de forma centralizada, rápida y organizada.

El objetivo principal es agilizar el proceso de notificación, mejorar la comunicación entre los usuarios y la administración del centro, y garantizar una gestión eficiente de las incidencias.

Alcance funcional del sistema

El sistema **incluirá** las **siguientes funcionalidades**:

- Inicio de sesión seguro mediante usuario y contraseña.
- Registro de nuevas incidencias por parte del personal del centro.
- Consulta del estado de las incidencias creadas por cada usuario.
- Revisión de incidencias por parte de administradores.
- Validación o denegación de incidencias por parte del administrador.
- Gestión del estado de las incidencias.
- Interfaz web accesible y sencilla para usuarios y administradores.

Fuera de alcance

Quedan **fuera del alcance** del proyecto las siguientes características:

- Registro de usuarios por parte del propio sistema (los usuarios se gestionan externamente).
- Envío de notificaciones por email, SMS o mensajería interna.
- Asignación de técnicos reales o sistema de mensajería técnico-usuario.
- Aplicación móvil nativa (solo versión web).
- Integración con sistemas externos del centro educativo.
- Historial avanzado de cambios por incidencia.

Definición de requisitos

Requisitos Funcionales (RF)

- RF01** El sistema permitirá iniciar sesión con credenciales válidas.
- RF02** El sistema gestionará intentos fallidos de inicio de sesión y aplicará un bloqueo de 10 minutos tras tres fallos consecutivos.
- RF03** El personal podrá registrar nuevas incidencias indicando aula, descripción y fecha.
- RF04** El personal podrá consultar sus incidencias y ver su estado actual.
- RF05** El administrador podrá consultar todas las incidencias registradas en el sistema.
- RF06** El administrador podrá revisar y validar incidencias.
- RF07** El administrador podrá denegar incidencias con un motivo.
- RF08** El sistema almacenará las incidencias y usuarios en una base de datos MySQL.
- RF09** El sistema gestionará la autenticación mediante JWT.
- RF10** El sistema dispondrá de una interfaz web basada en Angular.
- RF11** (opcional) El sistema podrá exportar listados de incidencias.

Requisitos No Funcionales (RNF)

RNF01 La aplicación estará desarrollada con Java 21 y Spring Boot 3.

RNF02 Se utilizará MySQL 8 como sistema gestor de base de datos, desplegado opcionalmente mediante Docker.

RNF03 La persistencia se gestionará mediante JPA/Hibernate.

RNF04 El frontend se desarrollará con Angular 17.

RNF05 La seguridad se implementará mediante JWT y roles (USER y ADMIN).

RNF06 Se incluirán pruebas unitarias con JUnit 5 y pruebas de integración con Postman.

RNF07 Se generará un informe de cobertura con JaCoCo (mínimo orientativo 50–70%).

RNF08 Se incluirá SonarLint como herramienta de análisis de calidad de código.

RNF09 El sistema se desplegará localmente en localhost:8080 (backend) y localhost:4200 (frontend).

RNF10 La interfaz será intuitiva, accesible y con navegación clara.

Análisis de roles y actores

El sistema tiene **dos actores** principales:

Actor 1: Personal del centro (Usuario estándar)

Descripción:

Corresponde al profesorado o personal del centro educativo que necesita reportar desperfectos detectados en las instalaciones.

Funciones principales:

- Iniciar sesión en el sistema utilizando sus credenciales.
- Registrar nuevas incidencias indicando los datos del desperfecto.
- Consultar el estado de las incidencias que él mismo ha creado.
- Revisar si sus incidencias han sido validadas o denegadas por el administrador.
- Interacción con el sistema:
- Accede únicamente a las funcionalidades relacionadas con la gestión de sus propias incidencias.

Actor 2: Administrador

Descripción:

Responsable de mantenimiento o personal encargado de gestionar todas las incidencias del centro.

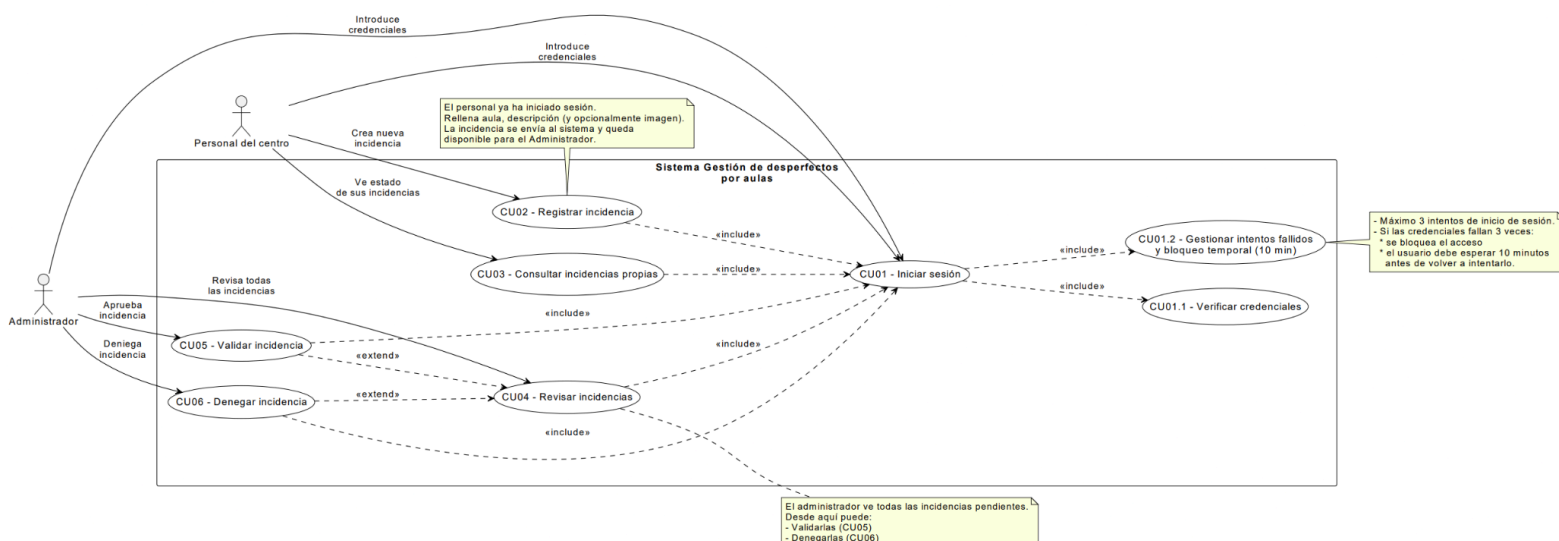
Funciones principales:

- Iniciar sesión igual que el personal del centro.
- Consultar el listado completo de incidencias registradas.
- Revisar la información de cada incidencia.
- Validar las incidencias aceptadas.
- Denegar aquellas que no sean correctas o no procedan.
- Interacción con el sistema:
- Accede a todas las incidencias y funcionalidades administrativas.

Diagrama de casos de uso (UML)

Aquí muestro un diagrama Uml del caso de uso, de cómo los actores interactúan con el sistema.

Estas son algunas de las **funcionalidades** que tendrán los actores.



Bocetos de Interfaz

Aquí he diseñado los wireframes de las pantallas principales del Frontend, la aplicación que he utilizado ha sido Figma en ella lo he diseñado tanto para versión móvil como para escritorio.

Estas son las pantallas que he diseñado:

1. Pantalla de login (básico).
2. Página de listado de incidencias.
3. Formulario de alta de incidencia.
4. Detalle de una incidencia.
5. Resolución o actualización de una incidencia (solo visible para administrador).

El proyecto de Fima esta dividido en **tres secciones**, una la vista **móvil** la cual se encuentra tanto la de **usuario normal** como la de **administrador**, otra para la versión de **escritorio** también con la vista de **usuario** y la de **administrador**, y por último, una para el diseño de una **tabla** usada en mucha de las **vistas**.

Para poder **visualizarlas** aquí te dejo el enlace para acceder a mi proyecto de [Figma](#).

Plan de Pruebas Inicial

Tipos de pruebas previstas

Para la aplicación de Gestión de Desperfectos se contemplan los siguientes tipos de pruebas:

Pruebas unitarias (JUnit)

Se aplicarán sobre el backend desarrollado en Java/Spring Boot.

Validará el funcionamiento individual de los métodos de la lógica de negocio, controladores y servicios.

Se empleará JUnit y se medirá la cobertura con JaCoCo.

Pruebas de integración (Postman)

Validará la comunicación entre el frontend y el backend mediante peticiones REST.

Se comprobarán los códigos HTTP, el contenido de las respuestas JSON y el flujo de autenticación basado en JWT.

Se desarrollarán colecciones en Postman para automatizar las validaciones básicas.

Pruebas de aceptación (funcionales)

Se realizan desde la perspectiva del usuario final (personal y administrador).

Validará que la aplicación permite ejecutar correctamente los flujos principales: inicio de sesión, registro de incidencias, consulta y visualización del detalle.

Objetivos del plan de pruebas

Los objetivos de este plan son:

1. Garantizar que las funcionalidades principales implementadas en los primeros sprints funcionan correctamente.
2. Detectar errores de forma temprana para reducir el coste de corrección.
3. Validar las reglas de negocio (bloqueo de login, creación de incidencias, estados, etc.) se comportan según lo especificado.
4. Verificar la correcta integración entre las distintas capas del sistema: frontend, API REST y base de datos MySQL.
5. Asegurar una experiencia adecuada para el usuario final mediante pruebas funcionales y de aceptación.

Casos de prueba iniciales

A continuación tienes 10 casos de prueba representativos correspondientes a los tres tipos de pruebas descritos:

Tipo	Descripción	Datos de entrada	Resultado esperado	Criterio de aceptación
Unitaria	Crear incidencia válida	Incidencia con tipo, aula y descripción válidos.	La incidencia se guarda en la base de datos y devuelve un ID generado.	No se lanza excepción y el ID es distinto de null.
Unitaria	Rechazar incidencia sin descripción	Incidencia con descripción vacía.	Se lanza excepción de validación.	La excepción coincide con el mensaje esperado.
Unitaria	Bloqueo tras 3 intentos fallidos de login	El usuario intenta iniciar sesión 3 veces con contraseña incorrecta.	El usuario queda bloqueado por 10 minutos.	bloqueado = true y no permite un cuarto intento.
Unitaria	Listar incidencias de un usuario	Usuario con 2 incidencias registradas en la base de datos.	El servicio devuelve una lista con exactamente 2 incidencias.	La lista contiene 2 elementos y ambos pertenecen al usuario.
Unitaria	Cambiar estado a RESUELTA	Incidencia con estado EN_PROCESO.	El estado cambia a RESUELTA y registra la fecha de resolución.	Estado final RESUELTA y fecha de resolución no nula.
Integración	Login correcto	POST /auth/login con credenciales correctas.	HTTP 200 y se devuelve un token JWT válido.	Código 200 y respuesta JSON contiene campo token.
Integración	Login incorrecto	POST /auth/login con contraseña incorrecta.	HTTP 401/403 y mensaje de error.	Código ≠ 200 y mensaje coincide con lo esperado.
Integración	Crear incidencia vía API	POST /incidencias con JSON válido y token.	HTTP 201 y devuelve la incidencia creada.	Código 201 y estado devuelto = PENDIENTE.
Aceptación	Flujo: login + crear incidencia	Usuario inicia sesión y rellena formulario correctamente.	Se crea la incidencia y aparece en 'Mis incidencias'.	La incidencia es visible en la lista del usuario.
Aceptación	Ver detalle de incidencia	Usuario selecciona una incidencia en la tabla.	Se muestra detalle: tipo, ubicación, descripción, estado, fechas.	Los datos coinciden con los almacenados en base de datos.