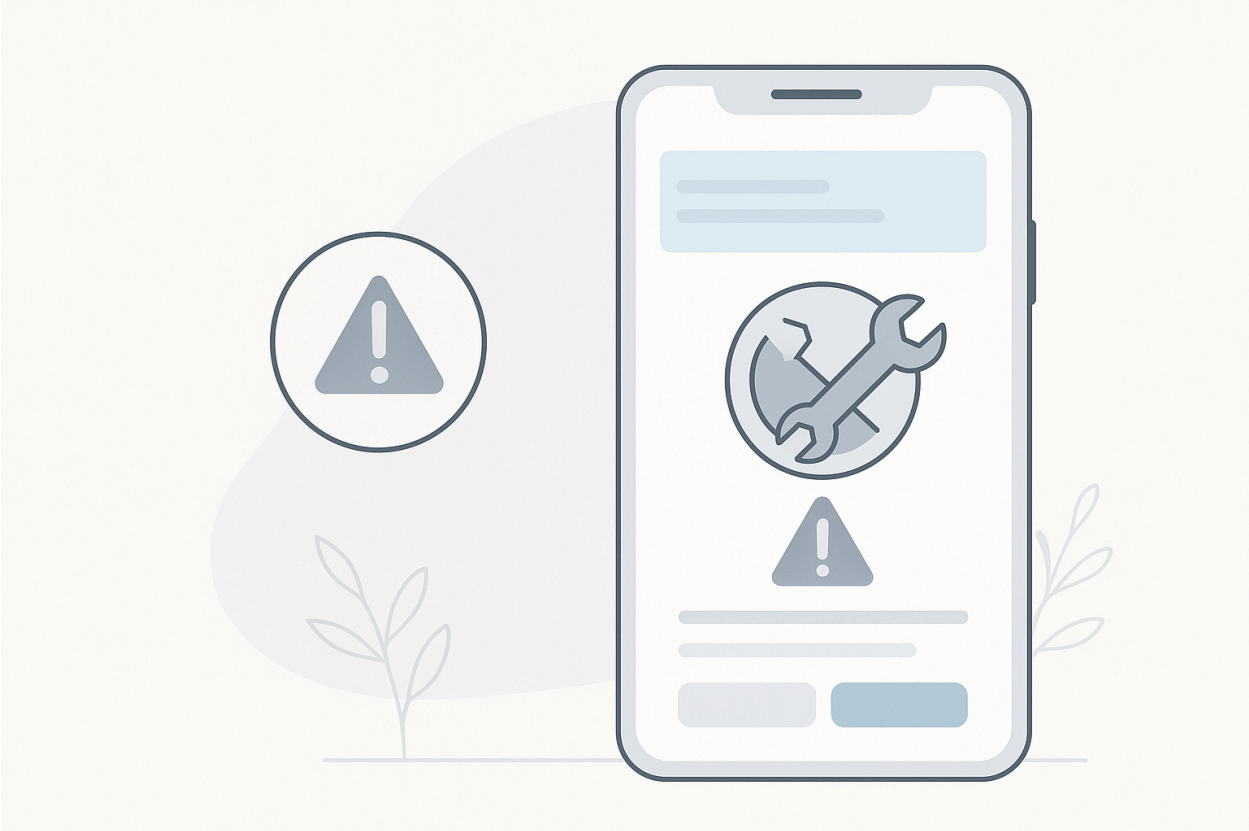
I.E.S Torre de los guzmanes

Proyecto Final

Gestión de desperfectos



Autor: Óscar Coronado Domínguez

2º DAM

Fecha : 29 de octubre de 2025

Índice

Introducción

En el proyecto final realizaremos la aplicación web **“Gestión de desperfectos por aulas”** para el registro, seguimiento y resolución de desperfectos detectados en las instalaciones del centro educativo (aulas, talleres, pasillos, etc.).

Este proyecto tendrá como **finalidad** llevarnos a realizar una aplicación en un **entorno real**, en donde los centros educativos se producen desperfectos en las aulas (puertas, ventanas, mobiliario, equipos informáticos, etc.).

El objetivo del proyecto es ofrecer **una herramienta web** para registrar, gestionar y resolver dichos desperfectos de forma ágil.

Tecnologías utilizadas

Backend

En el **backend** se utilizará la tecnología **Spring Boot** junto con **Java 21**.
Esta parte del proyecto será la encargada de implementar la **API REST**, los **servicios** y la **capa de acceso a datos**, permitiendo la comunicación entre el sistema y la base de datos MySQL la cual estará alojada en docker.

El entorno de desarrollo del **backend** se ha configurado utilizando el IDE **Eclipse**.

El estado **inicial** del backend ha sido generarlo a través de la herramienta **Spring Initializr**, dándonos la estructura inicial y la configuración básica de dependencias necesarias. Por lo que ahora mismo está en un **estado base**.

La **estructura general** del proyecto es la siguiente:

```
gestion-desperfectos-backend/
 ├─ src/main/java/
 ├─ src/main/resources/
 │   ├─ application.properties
     ├─ documentacion/
 ├─ target/
 ├─ pom.xml
 ├─ README.md
 ├─ .gitignore
```

Frontend

Para la parte del **frontend** he decidido utilizar el framework **Angular** en lugar de **Thymeleaf** como ya te comente , ya que es una forma mucho más profesional y es lo que ahora mismo están utilizando las empresas, así podre obtener más conocimientos de este framework envés de utilizar tymleaft que lo veo mas obsoleto

El entorno de desarrollo del **frontend** se ha configurado utilizando el IDE **Visual Studio Code.**

El **estado inicial** del frontend ha sido obtener la estructura base para empezar a desarrollar, mediante la generación de **CLI Angular** por defecto con la siguiente estructura. Por lo cual está en un **estado base**.

```
gestión-desperfectos-frontend/
 ├─ src/app/
 ├─ angular.json
 └─ package.json
```

Base de datos

Para la persistencia de los datos he decidido utilizar **MySQL**, una base de datos muy utilizada en proyectos desarrollados con **Spring Boot**, alojándola en un **contenedor Docker** para facilitar la portabilidad y evitar conflictos de versiones o configuraciones.

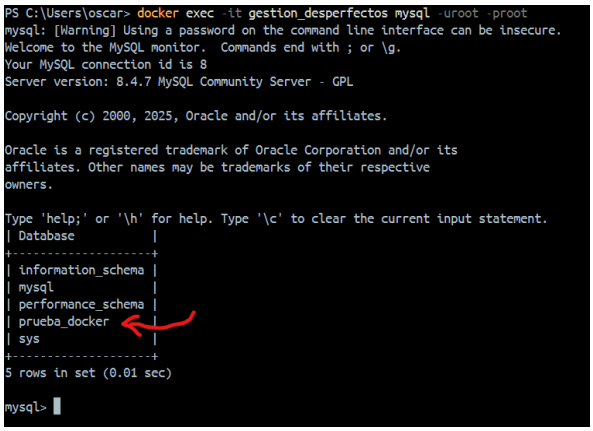
Además, al estar desplegada en **Docker**, es posible acceder a la base de datos desde cualquier equipo o entorno, como si se tratara de un servidor remoto, lo que permite simular un entorno real de despliegue y colaboración.

Para la gestión de la base de datos utilizo el gestor **MySQL Workbench**, que permite realizar consultas y añadir datos cuando sea necesario.

De esta forma, se obtiene una combinación eficaz —**MySQL + Docker + Workbench.**

De momento el estado inicial es la conexión de **MySQL + Docker + Workbench** y la realización de una base de datos mediante **Workbench Workbench** para poder visualizarla en docker y así verificar el estado de la conexión.

Aquí dejo una imagen.



Control de versiones

Para gestionar el control de versiones, como no estamos utilizando **Git** y alojando el proyecto en **Github**.

La forma de de alojar el proyecto que me propusiste fue de alojarlo todo el frontend y el backend en **el mismo repositorio**, pero yo he optado por hacer **repositorios separados**, uno para el frontend y otro para para el backend, de esta manera es más sencillo que yo o usted podamos portarlos más fácilmente a cada ide y no tener que hacer otros procesos de por medio.

Así que he decidido que la parte de documentación irá solo en la parte del backend en el **ressources** en la carpeta **documents**, como habrás visto en la estructura del backend.

De todas formas si esta forma no te parece la más adecuada házmelo saber y lo corregiré.

Aunque te he puesto de colaborador en mis repositorios, aquí tienes el enlace si es mas facil para ti:

Frontend:

https://github.com/oscarsheins/proyecto-gestion-desperfectos-frontend.git

Backend:

https://github.com/oscarsheins/proyecto-gestion-desperfectos-backend.git

Testing / Calidad

Para la realización de un código claro, mantenible y de calidad, se están utilizando **pruebas unitarias con JUnit**, junto con el **plugin JaCoCo** para el análisis de la cobertura del código.

Además, se emplea **SonarLint** integrado en el entorno de desarrollo, con el fin de detectar errores, malas prácticas y mejorar la legibilidad del código durante el proceso de programación.

El estado inicial de estas herramientas ha sido en su **instalación y configuración** mediante el archivo *pom.xml*.

Aunque todavía no se ha implementado ninguna prueba unitaria, el entorno ya está preparado para comenzar con su desarrollo en los próximos sprints.

Cronograma

Aquí tienes un cronograma sencillo y básico de los conceptos de todos los sprints.

Semana	Actividades principales	Objetivos
1	- Reunión inicial del equipo - Definir alcance y requisitos del proyecto - Configurar entorno de desarrollo (Java, Node, MySQL, Docker) - Crear repositorio Git	Tener el entorno preparado y los objetivos definidos
2	- Diseñar base de datos en MySQL - Crear entidades y repositorios en Spring Boot - Probar conexión con Docker	Backend base funcionando
3	- Crear componentes iniciales en Angular - Conectar frontend con backend - Diseñar interfaz básica (login, dashboard)	Aplicación funcional mínima (MVP)
4	- Agregar validaciones y autenticación JWT - Mejorar UI/UX - Documentar código y endpoints con Swagger	Aplicación segura y documentada
5	- Pruebas finales (unitarias e integración) - Corrección de errores - Despliegue en servidor o Docker Compose	Proyecto terminado y desplegado

Conclusión del sprint 1

Sé que en la documentación del proyecto se pedía elaborar un documento breve **(entre una y dos páginas)**, pero considero que esta es la mejor forma de asegurar que **no me dejo nada importante del trabajo**.

Con esta primera fase, se deja preparado el entorno de trabajo y la estructura base del proyecto sobre la que se desarrollarán las funcionalidades en los siguientes sprints. Además de los repositorios donde se irán actualizando el frontend y el backend.