



PROPUESTA PROYECTO DAW

Alumno: Jorge Alegre Maestre

Fecha: 7 de Octubre de 2025

Contenido

1.	Título y descripción general del proyecto	2
2.	Identificación del proyecto	2
3.	Justificación y objetivos	2
4.	Contenidos y aspectos principales.....	3
4.1	Requisitos funcionales.....	3
4.2	Requisitos no funcionales	4
4.3	Roles y permisos.....	4
5.	Medios que se utilizarán	5
5.1	Sistema operativo.....	5
5.2	Lenguajes de programación usados	5
5.3	Frameworks y librerías	6
5.4	Entorno de desarrollo	6
6.	Áreas del ciclo formativo	7



1. Título y descripción general del proyecto

Título:

Aplicación web interna para la gestión de la trazabilidad en hostelería

Descripción general:

El proyecto consiste en el desarrollo de una aplicación web interna que facilite la gestión de la trazabilidad de productos alimentarios dentro de una cocina profesional.

La herramienta permitirá al cocinero y al jefe de cocina registrar los productos recibidos, almacenar su información (nombre, fechas de entrega y caducidad, zona del almacén, trazabilidad, imagen del producto), y planificar menús semanales basados en la disponibilidad y caducidad de los ingredientes.

El sistema se centrará en la **organización eficiente del almacén** y en la **digitalización del control de productos** para mejorar la seguridad alimentaria y la planificación diaria del trabajo en cocina.

2. Identificación del proyecto

- **Participante:** Jorge Alegre Maestre
- **Ciclo formativo:** Desarrollo de Aplicaciones Web (2º DAW)
- **Centro educativo:** IES Albarregas, Mérida

3. Justificación y objetivos

La trazabilidad es un aspecto clave en la gestión alimentaria y en el cumplimiento de las normativas sanitarias. En muchos establecimientos, los datos de recepción, caducidad y procedencia de los productos se registran de forma manual, lo que puede generar errores y pérdida de información.

Este proyecto busca ofrecer una **solución digital moderna** que permita registrar, consultar y gestionar toda la información de trazabilidad de manera centralizada, además de facilitar la **planificación de menús** en base al inventario disponible.



Objetivos principales:

- ❖ Implementar un sistema de **roles y autenticación** (jefe de cocina y cocinero).
- ❖ Desarrollar un **CRUD completo de productos** (crear, leer, actualizar y eliminar).
- ❖ Permitir la **subida de imágenes** asociadas a cada producto (etiquetas, envases, etc.).
- ❖ Diseñar una **interfaz web responsive e intuitiva** con Bootstrap y Astro.
- ❖ Conectar la aplicación a **Supabase** para gestionar la base de datos, almacenamiento e imágenes.
- ❖ Integrar **Spring Boot** como backend para procesar las peticiones y asegurar la comunicación entre frontend y base de datos.

4. Contenidos y aspectos principales

4.1 Requisitos funcionales

- ❖ **RF1.** Registro y autenticación de usuarios (cocinero / jefe de cocina).
- ❖ **RF2.** CRUD de productos (alta, modificación, eliminación y consulta).
- ❖ **RF3.** Campos de producto: nombre, descripción, zona del almacén, fecha de entrega, fecha de caducidad, lote, proveedor y trazabilidad.
- ❖ **RF4.** Subida de imágenes asociadas a los productos (fotografía del producto o etiqueta).
- ❖ **RF5.** Gestión y visualización de zonas del almacén.
- ❖ **RF6.** Planificación de menús semanales según los productos disponibles y su caducidad.
- ❖ **RF7.** Control de acceso por roles (el jefe de cocina tiene permisos de gestión global; el cocinero puede registrar y consultar productos).
- ❖ **RF8.** Consulta del histórico de productos recibidos y caducados.



- ❖ **RF9.** Búsqueda y filtrado por nombre, proveedor o fecha.
- ❖ **RF10.** Exportación básica de informes (listado de productos y fechas críticas).

4.2 Requisitos no funcionales

- ❖ **RNF1.** El sistema deberá ser accesible desde navegadores modernos (Chrome, Edge, Firefox).
- ❖ **RNF2.** La interfaz será **responsive** y usable en dispositivos móviles y tablets de cocina.
- ❖ **RNF3.** El tiempo de respuesta medio será inferior a **2 segundos**.
- ❖ **RNF4.** Los datos estarán protegidos mediante **autenticación y roles** en Supabase y control de acceso en el backend.
- ❖ **RNF5.** El código seguirá una estructura modular y documentada (MVC en backend, componentes en frontend).
- ❖ **RNF6.** El almacenamiento de imágenes se realizará en **Supabase Storage** con rutas seguras.
- ❖ **RNF7.** Control de versiones con **Git y GitHub**, con ramas de desarrollo y commits documentados.
- ❖ **RNF8.** Pruebas funcionales y de integración antes del despliegue final.
- ❖ **RNF9.** Despliegue en entorno web interno o en servicios como Netlify (frontend) y Render (backend).
- ❖ **RNF10.** Cumplimiento de buenas prácticas de seguridad y protección de datos.

4.3 Roles y permisos

Acción / Permiso	Invitado	Cocinero	Jefe de Cocina
Ver landing o login	✓	✓	✓
Registro / Inicio de sesión	✓	✓	✓
Consultar productos		✓	✓



Registrar producto (CRUD)		✓	✓
Subir imagen del producto		✓	✓
Planificar menús semanales		✓	✓
Gestionar zonas del almacén			✓
Modificar/eliminar productos			✓
Gestionar usuarios			✓

5. Medios que se utilizarán

5.1 Sistema operativo

Se utilizará **Windows 11** como sistema principal de desarrollo, por ser el entorno instalado en el equipo personal del alumno.

5.2 Lenguajes de programación usados

- ❖ **JavaScript:** lenguaje principal del frontend (Astro).
- ❖ **HTML5:** estructura y contenido de las vistas web.
- ❖ **CSS3:** estilos personalizados y diseño responsive con Bootstrap.
- ❖ **Java:** lenguaje del backend con Spring Boot.
- ❖ **SQL (MySQL):** consultas, relaciones y estructuras de datos en Supabase.
- ❖ **JSON:** formato de intercambio de datos entre frontend y backend.



5.3 Frameworks y librerías

- ❖ **Astro:** framework moderno para construir el frontend con componentes estáticos y rápidos.
- ❖ **Bootstrap 5:** framework CSS para diseño adaptable y profesional.
- ❖ **Spring Boot:** framework backend en Java para la lógica de negocio y API REST.
- ❖ **Supabase:** backend-as-a-service con base de datos PostgreSQL, autenticación y almacenamiento de imágenes.
- ❖ **supabase-js:** librería oficial para conexión entre Astro y Supabase.
- ❖ **Spring Web:** módulos de Spring para construir y persistir los datos en el backend.
- ❖ **Maven:** gestión de dependencias en el proyecto Java.

5.4 Entorno de desarrollo

- ❖ **Visual Studio Code** para el desarrollo del frontend.
- ❖ **IntelliJ IDEA / Spring Tools Suite** para el desarrollo backend con Java.
- ❖ **Node.js y npm** como entorno de ejecución y gestión de dependencias en el frontend.
- ❖ **Git y GitHub** para control de versiones y documentación del proyecto.
- ❖ **Navegadores modernos (Chrome, Edge, Firefox)** para pruebas y depuración.
- ❖ **Supabase Dashboard** para gestión de tablas, roles e imágenes.



6. Áreas del ciclo formativo

El proyecto integra aprendizajes de los siguientes módulos del ciclo de **Desarrollo de Aplicaciones Web (DAW)**:

- ❖ **Desarrollo Web en Entorno Cliente:** implementación de la interfaz con Astro, JavaScript y Bootstrap.
- ❖ **Desarrollo Web en Entorno Servidor:** creación del backend con Spring Boot.
- ❖ **Despliegue de Aplicaciones Web:** configuración y publicación de frontend y backend.
- ❖ **Diseño de Interfaces Web:** elaboración de una interfaz responsive e intuitiva.
- ❖ **Bases de Datos:** modelado e integración de la base de datos en Supabase.
- ❖ **Entornos de Desarrollo:** uso de Git, IDEs y herramientas colaborativas.
- ❖ **Proyecto de Desarrollo de Aplicaciones Web:** integración práctica de todos los módulos anteriores en una aplicación completa y funcional.