



Propuesta de Proyecto: Gestor para Taller Mecánico "ReyBoxes"



20 DE MARZO DE 2025

PROYECTO PROPUESTO POR EL DEPARTAMENTO DE INFOMATICA (BRIANDA DE
MENDOZA)

Realizado por: D. Álvaro Gómez Tejada

Índice.

1.- Elección de la Idea	2
2.- Estudio de Viabilidad	2
3.- Objetivo del Proyecto	3
4.- Herramientas y Recursos a Utilizar	3

Propuesta de Proyecto: Gestor para Taller Mecánico "ReyBoxes"

1.- Elección de la Idea

El proyecto consiste en desarrollar un gestor integral para un taller mecánico. La iniciativa surge a petición del dueño del taller, quien, al conocer mi experiencia en programación, me solicitó una aplicación que centralice y simplifique la gestión del negocio. Con esta solución se busca eliminar la necesidad de utilizar entre 6 y 7 aplicaciones diferentes (cada una con su correspondiente licencia) y ofrecer una herramienta personalizada y adaptada a las necesidades del taller "ReyBoxes".

2.- Estudio de Viabilidad

Para llevar a cabo este proyecto se aplicarán los conocimientos de las siguientes asignaturas:

- **Programación:** Desarrollo del software que implementa la lógica del negocio.
- **Bases de Datos:** Gestión y almacenamiento de toda la información relevante del negocio.
- **Lenguaje de Marcas:** Desarrollo de una aplicación web que funcione en el navegador.
- **Entornos de Desarrollo:** Selección de IDEs y frameworks adecuados para optimizar el proceso de desarrollo.
- **Acceso a Datos:** Facilitar la manipulación y tratamiento de la información mediante técnicas especializadas.
- **Diseño de Interfaces:** Creación de una interfaz gráfica amigable, versátil y personalizada para el cliente.
- **Programación Multimedia y Dispositivos Móviles:** Desarrollo de una aplicación responsive que funcione correctamente en equipos de sobremesa, tablets y móviles.
- **Sistemas y Gestión de Empresas:** Aplicación de conceptos ERP para implementar una lógica de gestión personalizada y adaptada al negocio.
- **Programación de Servicios y Procesos:** Utilización de sockets para la comunicación con la base de datos alojada en un servicio web.

3.- Objetivo del Proyecto

El objetivo principal es cubrir la necesidad del taller mecánico "ReyBoxes" para gestionar el negocio de forma centralizada, eliminando la dependencia de múltiples aplicaciones y licencias. Se pretende ofrecer una solución personalizada que optimice la gestión de los procesos internos del taller, mejorando la eficiencia y facilitando el control de la información.

4.- Herramientas y Recursos a Utilizar

- **Equipos:** Un ordenador de sobremesa y un portátil, utilizando Visual Studio Code con sus respectivos plugins.
- **Backend (Servidor):**
 - **Node.js:** Plataforma de ejecución de JavaScript en el servidor.
 - **Express.js:** Framework para crear el servidor web y manejar rutas.
 - **Sequelize:** ORM para interactuar con la base de datos PostgreSQL usando JavaScript.
 - **PostgreSQL:** Base de datos para almacenar los datos del taller (usuarios, reparaciones, etc.).
 - **JWT (Json Web Token):** Sistema de autenticación basado en tokens para gestionar sesiones de usuarios.
 - **bcryptjs:** Librería para encriptar y comparar contraseñas de los usuarios.
 - **dotenv:** Gestión de variables de entorno (por ejemplo, credenciales de la base de datos).
 - **Nodemailer:** Librería para enviar correos electrónicos, como en el caso de la recuperación de contraseña.
- **Frontend (Interfaz de Usuario):**
 - **React.js:** Biblioteca para construir la interfaz de usuario de forma dinámica.
 - **Vite:** Herramienta de desarrollo rápida para aplicaciones React.
 - **React Router:** Gestión de rutas y navegación en la aplicación web.
 - **Bootstrap:** Framework CSS para diseñar una interfaz atractiva y responsive.
 - **Axios:** Cliente HTTP para realizar peticiones al backend.

- **LocalStorage:** Almacenamiento en el navegador para mantener la sesión activa.
- **Infraestructura y Despliegue:**
 - **Supabase:** Servicio en la nube que proporciona una base de datos PostgreSQL con API de autenticación y almacenamiento.
 - **Cors:** Middleware para permitir la comunicación entre el frontend y el backend.
 - **Nodemon:** Herramienta para recargar automáticamente el servidor cuando se realicen cambios en el código.

Integración de las Tecnologías

- **Frontend (React):** Presenta la interfaz gráfica y permite la interacción del usuario, enviando datos al backend mediante Axios y almacenando información de sesión en LocalStorage.
- **Backend (Node.js + Express + Sequelize):** Recibe y procesa las solicitudes provenientes del frontend, se conecta a la base de datos en Supabase y gestiona la autenticación de usuarios mediante JWT y bcryptjs, además de enviar correos a través de Nodemailer cuando sea necesario.
- **Base de Datos (PostgreSQL en Supabase):** Almacena toda la información crítica del sistema, como usuarios, reparaciones y facturas, permitiendo un acceso estructurado y eficiente a través de Sequelize.