# Bitácora del Proyecto: Los mejores autos

Fecha de inicio: 24/02/2025

Fecha de finalización: 16/03/2025

Equipo de desarrollo: Jose David Fernandez Barandica

## 1️ Definición del Proyecto

Objetivo:

Desarrollar una aplicación web de reservas de vehículos, permitiendo a los usuarios explorar, visualizar y reservar automóviles de manera sencilla e intuitiva. La aplicación también cuenta con un panel de administración para gestionar los productos disponibles.

Solución:

El sistema consiste en un Frontend en React y un Backend en Java Spring Boot con una base de datos H2, estructurados bajo principios S.O.L.I.D. para garantizar mantenibilidad y escalabilidad. Se implementó una arquitectura modular para separar la lógica de negocio, persistencia y presentación, asegurando así una solución escalable y eficiente.

## 2️ Funcionalidades Desarrolladas (Sprint Actual)

|  |  |  |
| --- | --- | --- |
| User Story # | Descripción | Estado |
| 1 | Colocar encabezado | Completado |
| 2 | Definir el cuerpo del sitio | Completado |
| 3 | Registrar producto |  |
| 4 | Visualizar productos aleatorios en la página de inicio. | Completado |
| 5 | Visualizar detalle de producto | Completado |
| 6 | Galería de imágenes en la página de detalle del producto. | Completado |
| 7 | Footer en toda la aplicación con información relevante. | Completado |
| 8 | Paginación en Home para visualizar productos en diferentes páginas. | Completado |
| 9 | Panel de Administración con acceso a herramientas de gestión. | Completado |
| 10 | Lista de Productos en el panel de administración. | Completado |
| 11 | Eliminar Producto con confirmación antes de eliminarlo. | Completado |

## 3 Tecnologías Utilizadas

Backend:

- Java Spring Boot → Framework principal del backend.

- H2 Database → Base de datos en memoria para pruebas.

- Spring Data JPA → Manejo de persistencia.

- Maven → Gestión de dependencias.

Frontend:

- React.js → Desarrollo del frontend con componentes reutilizables.

- React Router → Navegación entre páginas.

- Axios → Consumo de API.

- CSS y Media Queries → Diseño responsivo.

Herramientas de Desarrollo:

- Thunder client → Pruebas de API.

- GitHub → Control de versiones y colaboración.

- Maven & Node.js → Manejo de dependencias.

- VS Code → IDE utilizado.

## 4️ Retos y Soluciones

Problema: No se mostraban imágenes en la página principal.

Causa: El backend estaba almacenando imágenes con nombres genéricos.

Solución: Implementamos una convención de nombres (nombre\_producto\_1.jpg), asegurando que la primera imagen de cada producto fuera la principal.

Problema: El footer aparecía a la mitad en la página de detalle.

Causa: No estaba posicionado correctamente en la estructura de la página.

Solución: Ajustamos el `position` y `min-height` de `body` y `footer` para que siempre estuviera en la parte inferior.

Problema: No se podía eliminar un producto desde el panel de administración.

Causa: El botón de eliminar no tenía un evento `onClick`.

Solución: Implementamos `handleDelete` con un `window.confirm()`, y actualizamos el estado después de eliminar.

## 5️ Próximos Pasos

* Implementar autenticación de usuarios.
* Mejorar el sistema de subida de imágenes.
* Agregar más opciones de filtrado y búsqueda de productos.
* Optimización de rendimiento y carga diferida de imágenes.

## Conclusión

Este sprint permitió avanzar significativamente en la aplicación, logrando una navegación fluida, gestión de productos y experiencia de usuario mejorada. Se resolvieron diversos problemas técnicos y se dejó la base lista para nuevas mejoras.