



C.P.R. Liceo “La Paz”

Proyecto Fin de Ciclo

Desarrollo de Aplicaciones Web

Autor: Pablo Rivero Criado

Tutor: Jesús Ángel Pérez-Roca Fernández

Resumen

El proyecto TUT-RallyShop es una innovadora aplicación web de comercio electrónico centrada en coches de rally. La idea clave era crear una plataforma moderna y profesional donde los usuarios pudieran explorar y adquirir vehículos de manera sencilla y efectiva, replicando la experiencia de una tienda en línea real.

Desde el comienzo, se pensó en un desarrollo integral que abarcase tanto el aspecto visual como el funcional, prestando especial atención a la seguridad, la experiencia del usuario y la integración con servicios externos. La aplicación ha sido construida utilizando React y Vite para el frontend, mientras que Spring Boot se encarga del backend. Además, se hace uso de una base de datos MySQL y un sistema de autenticación seguro basado en JWT.

También se han implementado microservicios como la API de **Cloudinary**, que permite enviar, recibir y almacenar las imágenes de los usuarios y de sus proyectos, optimizando el rendimiento y la gestión multimedia de la plataforma.

En resumen, este proyecto busca demostrar una integración total entre el frontend, el backend y servicios externos, aplicando todo lo aprendido a lo largo del ciclo. Cumple con cada uno de los requisitos establecidos por el TFC: autenticación segura, gestión desde la web, pasarela de pago, carga de imágenes, control de stock, envío de emails, paginación, filtros visuales, uso de HTML semántico, favicon y títulos personalizados, entre otros. Todo el código está meticulosamente organizado y listo para ser presentado de manera profesional.

Abstract

TUT-RallyShop is an innovative e-commerce web application focused on rally cars. The main idea was to create a modern and professional platform where users can easily and effectively explore and purchase vehicles, replicating the experience of a real online store.

From the beginning, the project was designed as a comprehensive solution, covering both the visual and functional aspects, with special attention to security, user experience, and integration with external services. The application is built using React and Vite for the frontend, while the backend is powered by Spring Boot. It also relies on a MySQL data-base and a secure JWT-based authentication system.

Additionally, microservices such as the Cloudinary API have been integrated to handle user and product image uploads, storage, and delivery, enhancing multimedia management and overall platform performance.

To sum up, this project aims to demonstrate full integration between the frontend, backend, and external services, applying all the knowledge acquired throughout the course. It meets all the official requirements for the Final Project: secure authentication, web-based management, payment gateway, image upload, stock control, email notifications, pagination, visual filters, semantic HTML usage, favicon, and custom page titles, among others.

All the code is thoroughly organized and ready to be presented in a professional manner.

Palabras Clave

React: es una popular librería de JavaScript que se utiliza para construir interfaces de usuario. Es ideal para desarrollar el frontend de aplicaciones, ya que permite la creación de componentes reutilizables, lo que facilita la elaboración de interfaces que son tanto interactivas como responsivas.

Vite: es una herramienta moderna que se encarga del desarrollo frontend. Esta mejora notablemente la velocidad de carga y la compilación del proyecto. En TUT-RallyShop, se utiliza Vite para optimizar el rendimiento durante el desarrollo en React.

Spring Boot: es un framework para Java que hace que la creación de aplicaciones web escalables sea mucho más sencilla. Este framework gestiona la lógica de negocio, la seguridad y la conexión con la base de datos.

MySQL: es el sistema de gestión relacional que utilizamos para almacenar de manera estructurada la información de usuarios, coches, pedidos y favoritos dentro de la plataforma.

JWT (JSON Web Token): es un método que permite identificar a los usuarios de forma confiable y proteger el acceso a las funcionalidades del sistema.

Cloudinary: es un servicio en la nube que se encarga de almacenar, optimizar y entregar imágenes. En este proyecto, se utiliza para gestionar y visualizar las imágenes de los coches.

Stripe: es la pasarela de pago que hemos integrado para procesar pagos con tarjeta de manera segura. Esto permite a los usuarios comprar coches de rally directamente desde el carrito de compras.

API REST: es un estilo arquitectónico que facilita la comunicación entre el frontend y el backend a través de peticiones HTTP. Esto es clave para gestionar operaciones CRUD y transferir datos en formato JSON.

UX/UI: Conjunto de principios aplicados al diseño de la interfaz (UI) y la experiencia de usuario (UX). En TUT-RallyShop se ha priorizado una navegación fluida, filtros intuitivos y un diseño visual atractivo para mejorar la interacción del usuario.



Proyecto Desarrollo de Aplicaciones Web

Gracias a mi familia, mis amigos y a los profesores que me han impartido clase.

Proyecto Desarrollo de Aplicaciones Web

Resumen	3
Abstract.....	4
Palabras Clave	5
Introducción/motivación.....	11
Objetivos.....	12
Estado del arte.....	13
Caso de estudio.....	15
Arquitectura de la pagina.....	17
Diagramas	20
Wireframes y Mocks	24
Guía de estilos.....	28
Desarrollo del proyecto	30
Manual Administrador	32
Manual Usuario.....	35
Viabilidad tecno-económica.....	42
Trabajo futuro.....	45
Conclusiones.....	47
Biblioteca de recursos web y referencias.....	¡Error! Marcador no definido.
Anexos.....	48

Introducción/motivación.

Desde el comienzo del ciclo formativo en Desarrollo de Aplicaciones Web, tuve muy claro que quería aprovechar el proyecto final para poner en práctica todo lo aprendido. Mi meta no era crear una plataforma grande y complicada, sino desarrollar una pagina sencilla, pero a la vez completa, funcional y coherente, que realmente refleje el esfuerzo de estos años.

Así es como nació TUT-RallyShop, una tienda online enfocada en coches de rally. Este tema, que me resulta visualmente atractivo, me ofreció la oportunidad de construir un proyecto con una buena lógica, un diseño atractivo y funcionalidades reales.

A lo largo del desarrollo, siempre he priorizado la claridad, la estabilidad del sistema y una estructura bien organizada. He puesto atención tanto en el aspecto visual como en el funcionamiento interno, integrando características como el sistema de autenticación con JWT, la subida de imágenes a través de Cloudinary, la pasarela de pago con Stripe, el control de stock, filtros de búsqueda, favoritos, y un panel de administración que funciona correctamente.

Todas estas decisiones están alineadas con los requisitos establecidos para el TFC, pero también buscan preparar una aplicación que tenga sentido en un contexto real. Mi principal motivación ha sido explorar hasta dónde podía llegar con mis propios recursos. No buscaba destacar por hacer algo grandioso, sino por hacerlo bien.

He decidido afrontar todos los desafíos que implica un desarrollo web completo, desde el diseño hasta la lógica de negocio, utilizando herramientas actuales y buenas prácticas. Este proyecto no es solo una exhibición de innovación, sino constancia, esfuerzo y aprendizaje.

Objetivos.

Uno de los principales objetivos al iniciar este proyecto fue demostrarle a mí mismo que podía crear una aplicación completa desde cero. Más allá de cumplir con los requisitos académicos del TFC, deseaba construir algo que realmente funcionara, que tuviera sentido y que reflejara el esfuerzo y las horas invertidas a lo largo del ciclo.

Además, me propuse aprender a integrar adecuadamente todos los elementos de un desarrollo web moderno: desde el diseño visual hasta la lógica del backend, sin olvidar la seguridad, el control de datos y la experiencia del usuario. Quería que cada parte estuviera conectada de manera coherente, y que la aplicación pudiera usarse de verdad, como si se tratara de una tienda real.

Otro objetivo personal fue superar mis propias limitaciones técnicas. Durante el proyecto me enfrenté a muchos desafíos: desde el sistema de autenticación con JWT, el manejo de imágenes mediante Cloudinary, hasta la integración de pagos con Stripe o la creación de un panel de administración. Cada uno de estos aspectos representó una meta en sí misma, y lograr cada uno de ellos me ha hecho sentir que he crecido significativamente como desarrollador.

También quise cuidar el diseño visual, dentro de mis posibilidades, y crear una web que fuera sencilla pero agradable de usar, que cargara rápidamente, que funcionara en distintos dispositivos y que tuviera un mínimo de estilo profesional.

Por último, uno de mis objetivos fue entregar un proyecto del que me sintiera orgulloso. Aunque sea sencillo y sin grandes pretensiones, he intentado hacerlo bien, organizarlo de forma adecuada, documentarlo como es debido y preparar todo para que se pueda probar fácilmente y presentar sin errores.

Estado del arte.

Hoy en día, hay una gran cantidad de plataformas que se dedican a la compra y venta de vehículos. Muchas de estas están enfocadas en coches deportivos, clásicos o de competición. También existen opciones más generales de comercio electrónico que permiten vender una amplia variedad de productos, incluidos automóviles. A continuación, se presenta un análisis de algunas plataformas destacadas y su comparación con TUT-RallyShop.

senrasports.com es una tienda que se especializa en la venta de coches de rally, para competición como para uso diario. Su catálogo es bastante completo, dando información técnica detallada y un contacto directo con el vendedor.

- **Puntos fuertes:** Su enfoque especializado en coches de rally, un catálogo profesional, y una marca bien establecida.
- **Puntos débiles:** No es una plataforma interactiva; carece de registro de usuarios y no permite funciones como agregar favoritos o realizar compras online. La navegación es bastante básica y no cuenta con filtros dinámicos.

Coches.net es un portal muy popular para la compraventa de vehículos usados, con una amplia variedad de modelos, incluidos deportivos y de rally.

- **Puntos fuertes:** Gran volumen de vehículos, un sistema de filtros detallados, una app móvil y una comunidad activa.
- **Puntos débiles:** Su enfoque está más en los anuncios que en las compras directas. No ofrece funciones como favoritos ni un método de pago integrado en la plataforma.

Wallapop es una aplicación muy conocida para la compraventa entre particulares, donde también es posible encontrar vehículos. Funciona como un marketplace, pero no gestiona los pagos internamente.

- **Puntos fuertes:** Es fácil de usar para publicar anuncios y contactar con otros usuarios, y es muy popular entre los jóvenes.
- **Puntos débiles:** No tiene una estructura de tienda ni una pasarela de pago. Además, no permite funciones específicas como la gestión de stock o filtros personalizados para coches.

Por su parte, **TUT-RallyShop** surge como una plataforma más modesta, pero con metas claras: ofrecer una experiencia auténtica de compra online, centrada en coches de rally, que funcione de manera integral desde el registro del usuario hasta el pago final. A diferencia de muchas plataformas que solo actúan como intermediarios de anuncios, esta aplicación incluye herramientas que

mejoran la experiencia del usuario: gestión de favoritos, pasarela de pago con Stripe, control de stock, filtros visuales....

Proyecto Desarrollo de Aplicaciones Web

Además, todo ha sido desarrollado desde cero, sin el uso de sistemas de gestión de contenido o plantillas, conectando tecnologías reales del sector como React, Vite, Spring Boot, JWT, Stripe, MySQL y Cloudinary. Se ha prestado especial atención a la estructura del código, la seguridad y la facilidad de uso desde diferentes dispositivos. Aunque no compite en volumen ni en complejidad con plataformas comerciales establecidas, TUT-RallyShop se destaca por ofrecer una experiencia de comercio electrónico real, completa y funcional, desarrollada por un único estudiante.

Caso de estudio.

El proyecto TUT-RallyShop nace con la meta de demostrar que, dentro de un ciclo formativo, es posible crear una tienda online que sea funcional, segura y profesional. Antes de iniciar el desarrollo, se identificaron varias necesidades y desafíos reales que cualquier comercio electrónico moderno debe enfrentar.

Este caso de estudio detalla cómo se idearon soluciones viables desde el principio, con una perspectiva integral.

1. Autenticación segura y gestión de usuarios

- Posible problema: Muchos proyectos educativos suelen simular el proceso de inicio de sesión sin implementar medidas de seguridad efectivas, lo que puede poner en riesgo tanto la lógica de roles como la protección de datos.
- Solución planteada: Se buscaría implementar un sistema de autenticación real con JWT, encriptación de contraseñas y control de roles diferenciados (usuario y administrador), garantizando así el acceso seguro a distintas partes de la aplicación..

2. Interfaz moderna, accesible y responsive

- Posible problema: En muchas iniciativas académicas, la interfaz es básica y carece de un diseño responsive, lo que limita tanto la usabilidad como la profesionalidad del proyecto.
- Solución planteada: Se diseñaría una interfaz clara, visual y adaptativa utilizando React y Vite, incorporando filtros visuales, tarjetas (cards) dinámicas y una navegación sencilla tanto en escritorio como en dispositivos móviles.

3. Integración real con pasarela de pago

- Posible problema: Simular un proceso de compra sin conexión con servicios reales deja incompleto el flujo de e-commerce.
- Solución planteada: Se consideraría la integración de Stripe como pasarela de pago, con un carrito persistente, gestión del stock, y redirecciones a páginas de éxito o cancelación tras el proceso de pago.

4. Gestión multimedia mediante almacenamiento en la nube

- Posible problema: Cargar imágenes de forma local trae consigo problemas de escalabilidad y portabilidad del proyecto.

- Solución planteada: Se optaría por integrar Cloudinary para permitir la subida de imágenes desde el panel de administración, almacenándolas externamente y mostrando las URLs optimizadas en el frontend.

5. Panel de administración web completo

- Posible problema: Muchos proyectos no permiten modificar datos directamente desde la web, obligando a acceder a la base de datos.
- Solución planteada: Se desarrollaría un panel exclusivo para administradores donde se pudieran crear, editar o eliminar coches, gestionar el stock y controlar el contenido visible en la tienda.

6. Experiencia de usuario completa e integrada

- Posible problema: Tener funcionalidades aisladas o mal conectadas genera una experiencia confusa para el usuario.
- Solución planteada: Se planearía una experiencia integrada que incluya registro, inicio de sesión, sistema de favoritos, carrito de la compra, filtros por marca, modelo y precio, paginación, favicon, título personalizado y una estructura visual coherente..

Este enfoque previo al desarrollo ha permitido plantear soluciones coherentes, funcionales y realistas. Aunque TUT-RallyShop no tiene la complejidad de una plataforma comercial, está diseñada con una mentalidad profesional y construida paso a paso, tal como lo haría un desarrollador en un entorno laboral real.

Arquitectura de la pagina.

Desde el principio tuve claro que no quería complicarme con tecnologías que no dominara del todo, pero sí montar una estructura real, bien organizada y funcional. La arquitectura que seguí separa claramente el frontend, el backend, la base de datos y los servicios externos como Cloudinary. Así cada parte hace su función y se comunica con el resto de forma ordenada.

No utilicé ningún CMS ni nada parecido, todo lo desarrollé y lo ejecuté en mi entorno de desarrollo. Para la base de datos y su gestión usé **MySQL** junto con **phpMyAdmin**,

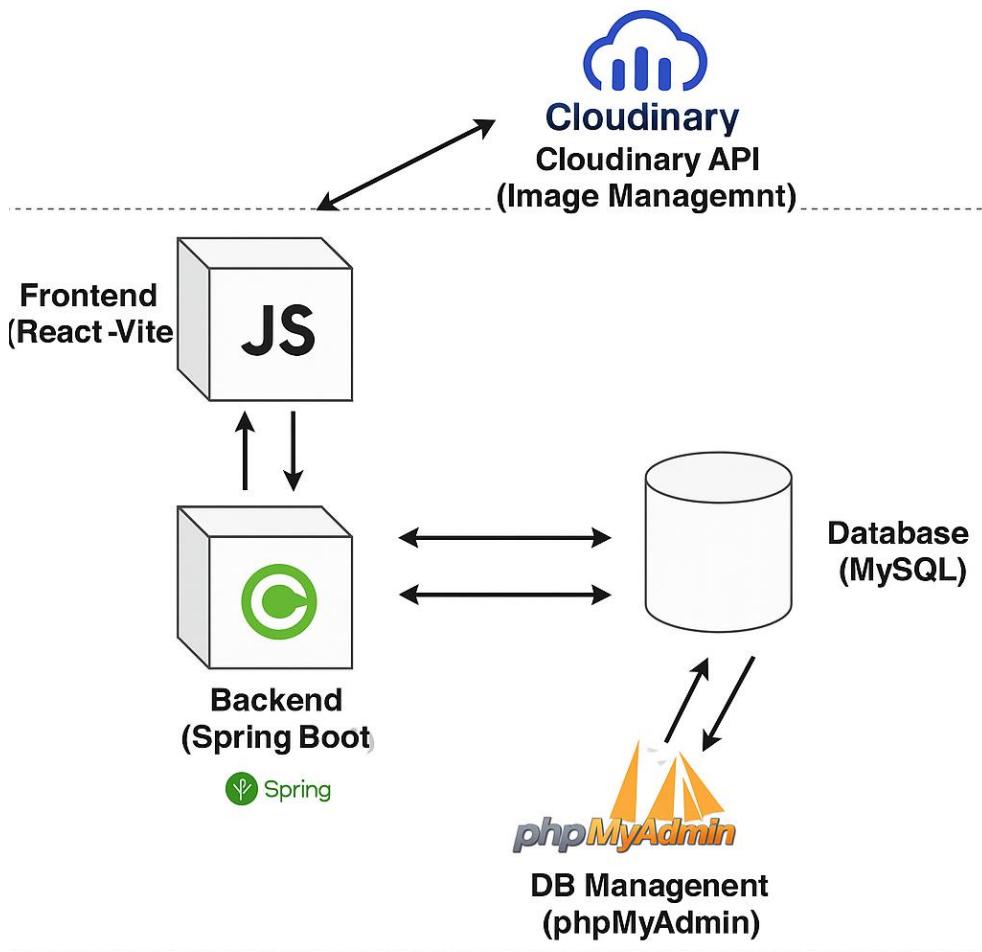


Figura 1: Arquitectura de la aplicación. Elaboración propia.

Frontend (React + Vite)

La parte visual está hecha con **React** y el sistema de desarrollo es **Vite**, que va más rápido que Create React App. Todo está separado en componentes: filtros, cards, detalles del coche, login, carrito...

Uso **Axios** para hablar con el backend y todo está preparado para que se vea bien en cualquier dispositivo (responsive). También añadí favicon y título personalizado por cada vista.

Backend (Spring boot)

Para el backend usé **Spring Boot**, que es muy potente y ya lo dominábamos en el ciclo. Aquí está toda la lógica: seguridad, acceso a datos, favoritos, compras, etc. Usé **JWT** para que al hacer login se genere un token y así se controle qué puede ver o hacer el usuario. También encripto las contraseñas y tengo control de roles (USER y ADMIN).

Organización del backend:

- Controladores → reciben peticiones.
- Servicios → lógica de negocio.
- Repositorios → JPA para conectar con MySQL.
- Entidades → lo que se guarda en la base de datos.

Base de datos MySQL

Toda la información está guardada en una base de datos relacional hecha con **MySQL**. Las tablas están bien relacionadas con claves foráneas. Durante el desarrollo usé **phpMyAdmin** para ver los datos, probar cosas o hacer consultas si algo no cuadraba.

Tablas principales:

- usuarios
- coches
- favoritos
- compras

Hay mas tablas, pero son para en un futuro poder ir mejorando la página.

Claudinary – Gestión de imágenes

Para no complicarme guardando imágenes en local, usé **Cloudinary**, que es un servicio en la nube. Al crear o editar un coche desde el panel de admin, se puede subir una imagen y automáticamente se guarda en Cloudinary. Luego, desde el frontend, simplemente uso la URL y se ve perfectamente.

Proyecto Desarrollo de Aplicaciones Web

phpMyAdmin – Visualización de la base de datos

Durante el desarrollo, para comprobar lo que tenía guardado en MySQL, utilicé **phpMyAdmin**. Me ayudó mucho a ver si se estaban guardando bien los favoritos, los coches, los usuarios, etc.

Como se comunica todo

- El **frontend** le pide cosas al **backend** (coches, login, favoritos, etc.).
 - El **backend** habla con la base de datos usando JPA.
 - Las imágenes se suben a **Cloudinary** desde el **backend**.
 - phpMyAdmin sirve para ver los datos desde el navegador.
-

Aunque es un proyecto de un alumno, la estructura está pensada como si fuera una aplicación real, separada por capas, con autenticación, pagos, administración y buenas prácticas. Todo esto lo hice paso a paso, conectando piezas hasta que funcionara como una tienda online completa.

Diagrams.

Diagrama E-R (Entidad-Relación)

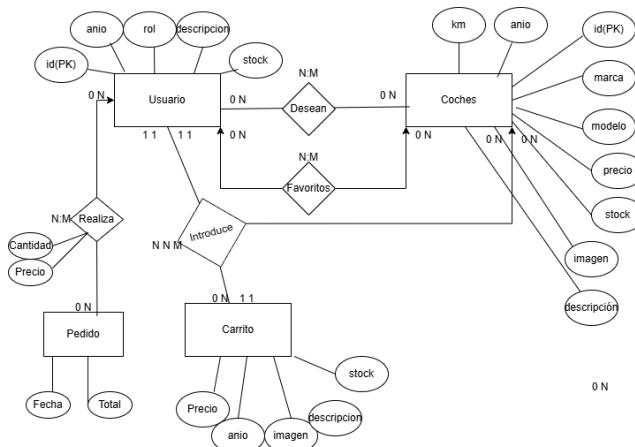


Figura 2: Diagrama E-R. Elaboración propia.

Usuario: Representa a los usuarios registrados en la plataforma. Tiene atributos como id, rol, descripción, stock y año (aunque algunos no se usan en producción). Cada usuario puede:

- Introducir coches en su carrito (relación **Introduce**).
- Realizar pedidos (relación **Realiza**).
- Añadir coches como favoritos (relación **Favoritos**).
- Desear coches (relación **Desean**, no implementado pero si creado en código)

Coche: Representa los coches de rally disponibles en la tienda. Cada coche tiene atributos como id, marca, modelo, precio, stock, imagen, descripción, año y km. Un coche puede:

- Ser añadido por muchos usuarios al carrito.
- Ser deseado por muchos usuarios.
- Estar en múltiples favoritos.
- Estar asociado a muchos pedidos.

Carrito: Es la entidad que guarda temporalmente los coches que un usuario quiere comprar. Tiene atributos como precio, stock, descripción, imagen y año. Está en relación N:M entre Usuario y Coche, mediante la relación **ternaria**.

- **¿Por qué relación ternaria?** Cada vez que un usuario añade un coche al carrito necesitamos saber a la vez quién es el usuario, cuál es su carrito y qué coche es. Con esa ternaria (“Usuario–Carrito–Coche”) evitamos ambigüedades y garantizamos que el carrito siempre pertenezca al usuario que realiza la acción.

Favoritos: Es una relación N:M entre Usuario y Coche que indica qué coches ha marcado como favoritos cada usuario.

Proyecto Desarrollo de Aplicaciones Web

Desean: Relación N:M similar a Favoritos, representa los coches que los usuarios han marcado como deseados (no activa en el sistema actual).

Pedido: Representa un pedido finalizado. Tiene atributos como fecha y total, y se relaciona con el usuario que lo realizó y los coches incluidos, mediante la relación **Realiza**. Contiene también la cantidad y el precio en cada línea del pedido.

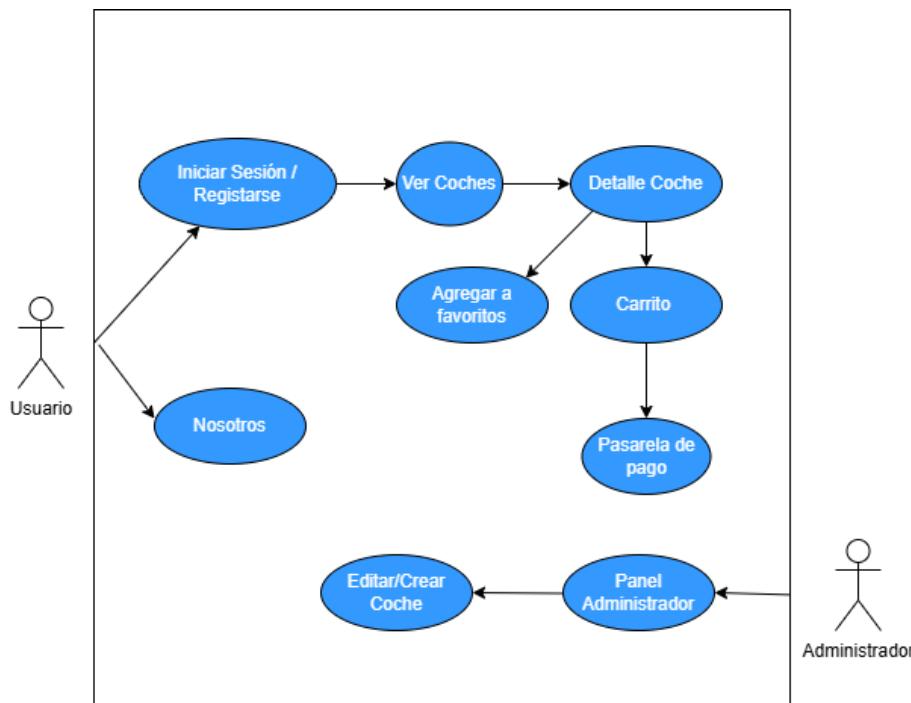


Figura 3: Diagrama caso de uso. Elaboración propia.

Diagrama de Casos de Uso

Este diagrama de casos de uso representa las funcionalidades principales que puede realizar un usuario dentro de la pagina. Está dividido entre las acciones del usuario general y del administrador, detallando las rutas más importantes de interacción con el sistema.

Usuario:

- **Iniciar sesión / Registrarse:** Accede al sistema mediante sus credenciales o crea una cuenta nueva. Es el paso inicial para poder guardar favoritos o comprar.
- **Ver Coches:** Accede al catálogo principal con paginación y filtros, sin necesidad de estar logueado.
- **Detalle Coche:** Consulta la ficha completa de un coche (marca, modelo, precio, stock, descripción...).
- **Agregar a Favoritos:** Si está logueado, puede añadir coches a su lista personal de favoritos.
- **Carrito:** Añade coches al carrito para realizar una compra más adelante. El carrito es persistente mientras dure la sesión.

- **Pasarela de Pago:** Inicia el proceso de compra mediante Stripe. Solo accesible con sesión iniciada y coches en el carrito.
- **Nosotros:** Página estática accesible desde cualquier punto, con información sobre el proyecto, contacto y localización.

Administrador:

- **Panel Administrador:** Zona protegida con acceso solo para usuarios con rol admin. Desde aquí se gestionan los coches del sistema.
- **Editar / Crear Coche:** Desde el panel se pueden crear nuevos coches o editar los existentes. Incluye subida de imágenes, gestión de stock y edición de descripciones.

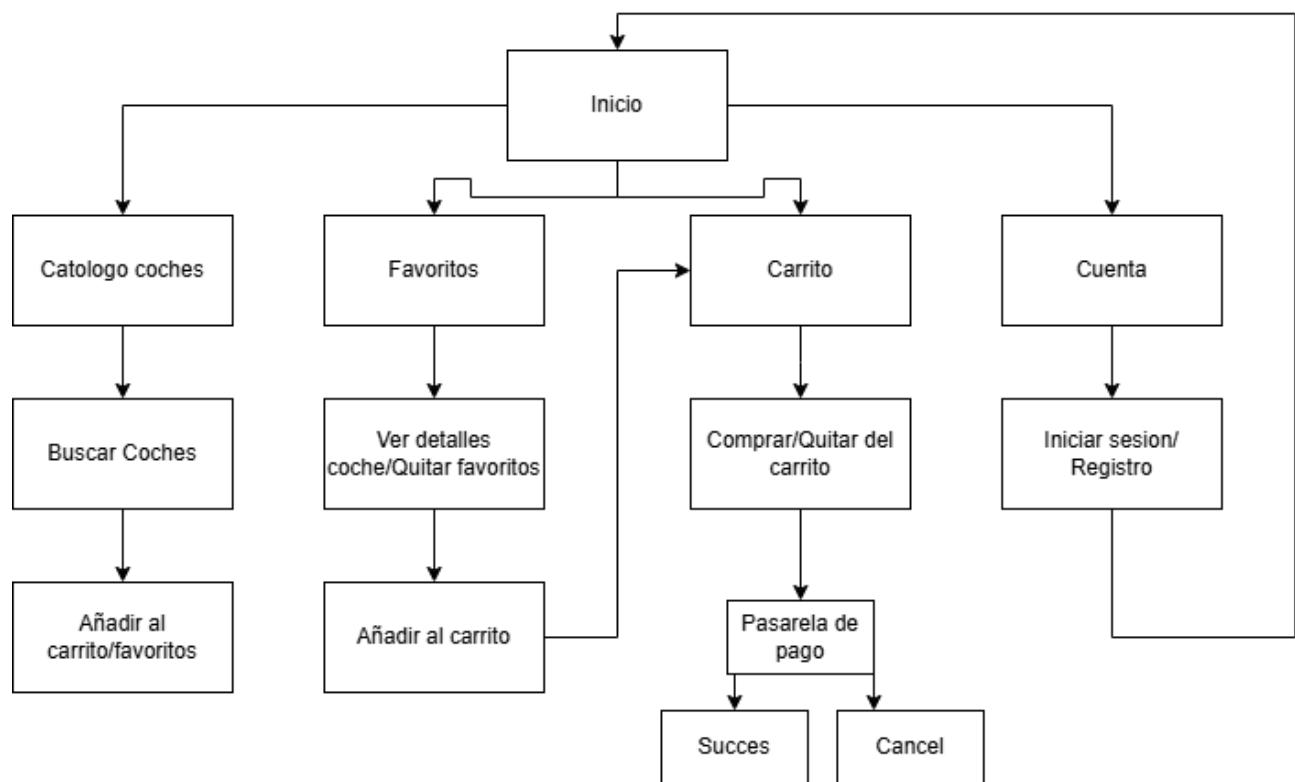


Figura 4: Mapa de navegación. Elaboración propia.

Mapa de Navegación

El mapa de navegación representa gráficamente las rutas y conexiones principales de la aplicación web TUT-RallyShop. Permite visualizar cómo interactúa un usuario con las distintas funcionalidades, desde el acceso hasta la compra o la gestión del contenido.

Explicación:

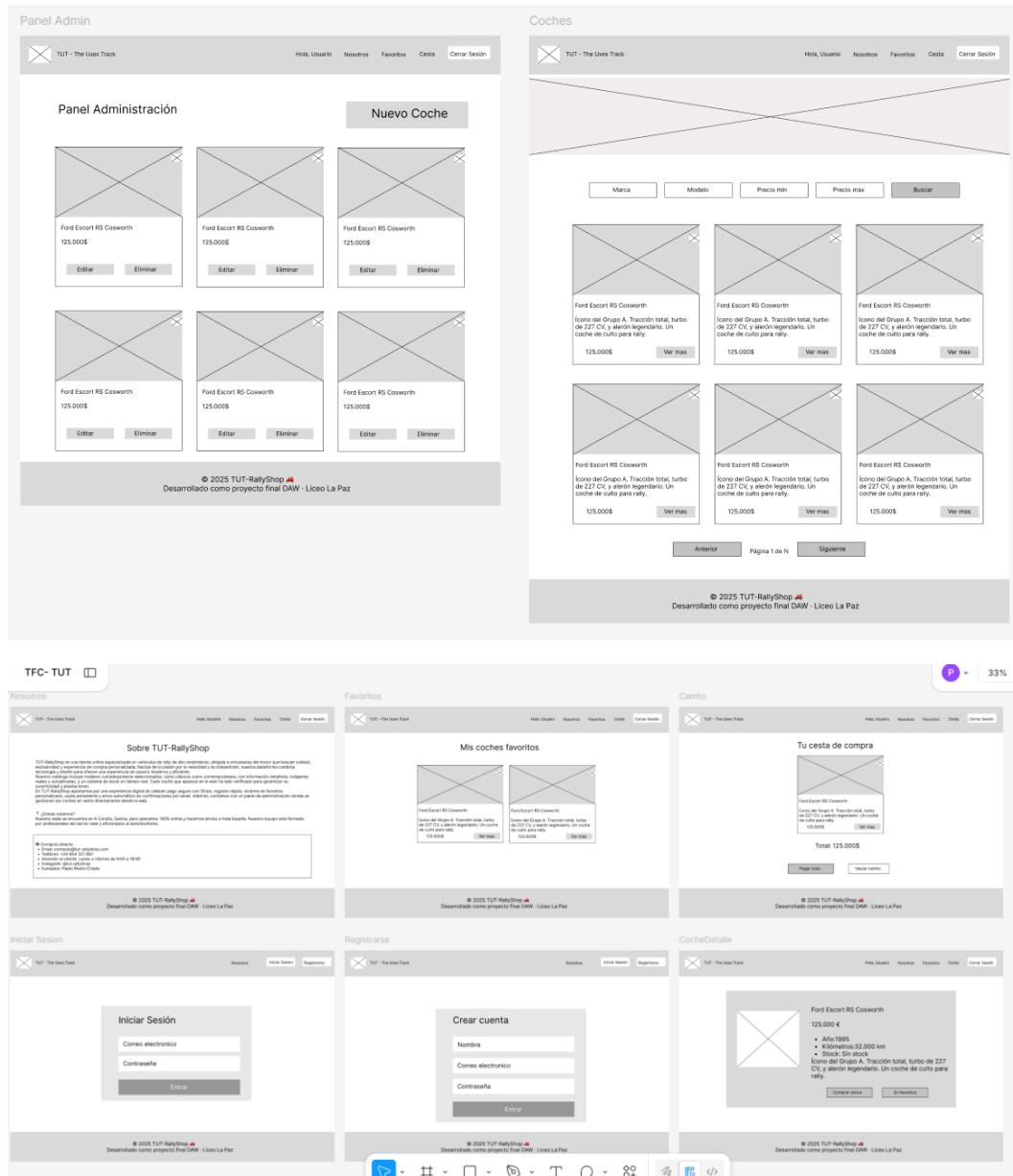
- **Iniciar sesión / Registrarse:** Entrada inicial para acceder a funcionalidades como favoritos, carrito y compra. El usuario introduce su correo y contraseña o se crea una nueva cuenta.

Proyecto Desarrollo de Aplicaciones Web

- **Ver Coches:** Página principal del catálogo. Aquí se muestran los coches en formato de cards con paginación y filtros. No requiere estar logueado.
- **Detalle Coche:** Vista ampliada de un coche seleccionado, con más información y botones para añadir a favoritos o al carrito. Si no estás logueado, esos botones aparecen desactivados.
- **Agregar a Favoritos:** Si el usuario ha iniciado sesión, puede guardar coches en su lista de favoritos desde el catálogo o desde el detalle.
- **Carrito:** Acceso al carrito persistente donde se visualizan los coches añadidos. Desde aquí se puede quitar elementos o iniciar el proceso de compra.
- **Pasarela de Pago:** Redirección a Stripe para completar la compra. Solo accesible si el usuario ha iniciado sesión y hay coches en el carrito.
- **Nosotros:** Página informativa con detalles del proyecto, contacto y ubicación. Accesible siempre, sin necesidad de iniciar sesión.
- **Panel Administrador:** Accesible únicamente si el usuario tiene rol admin. Desde aquí se pueden ver todos los coches y gestionar el inventario.
- **Editar / Crear Coche:** Dentro del panel de administrador, permite modificar o crear nuevos coches, incluyendo imagen, precio, stock, etc.

Wireframes y Mocks

Wireframes:



The image displays a series of wireframes and mockups for the TUT-RallyShop application, illustrating its development from a basic administration panel to a full e-commerce platform. The wireframes are arranged in three main sections:

- Panel Admin:** Shows the initial administration interface with a grid of vehicle cards (Ford Escort RS Cosworth) and buttons for 'Editar' and 'Eliminar'.
- Coches:** Shows the evolution of the car listing page, including a search bar, filters (Marca, Modelo, Precio min, Precio max), and a detailed view of a car card (Ford Escort RS Cosworth).
- TFC- TUT:** Shows the final e-commerce interface with sections for 'Mis coches favoritos' (Favorites), 'Tu cesta de compra' (Cart), and user authentication (Iniciar Sesión, Registrarse).

Each section includes a copyright notice: "© 2025 TUT-RallyShop" and "Desarrollado como proyecto final DAW - Liceo La Paz".

Figuras 5. Wireframes Elaboración propia.

Desde la primera versión, el diseño de TUT-RallyShop ha evolucionado de forma significativa, pero siempre manteniendo una idea clara: ofrecer una interfaz moderna, funcional y fácil de usar para cualquier tipo de usuario, incluso sin experiencia en plataformas e-commerce.

Al inicio, se optó por una estructura clásica de tienda online, con elementos bien definidos: un encabezado en la parte superior, filtros fácilmente accesibles, tarjetas de productos y llamadas a la

Proyecto Desarrollo de Aplicaciones Web

acción claras como 'Ver más', 'Favoritos' o 'Comprar ahora'. A medida que avanzaba el desarrollo, se realizaron numerosos ajustes para optimizar la experiencia del usuario, mejorar el diseño visual y asegurar la adaptabilidad a diferentes dispositivos.

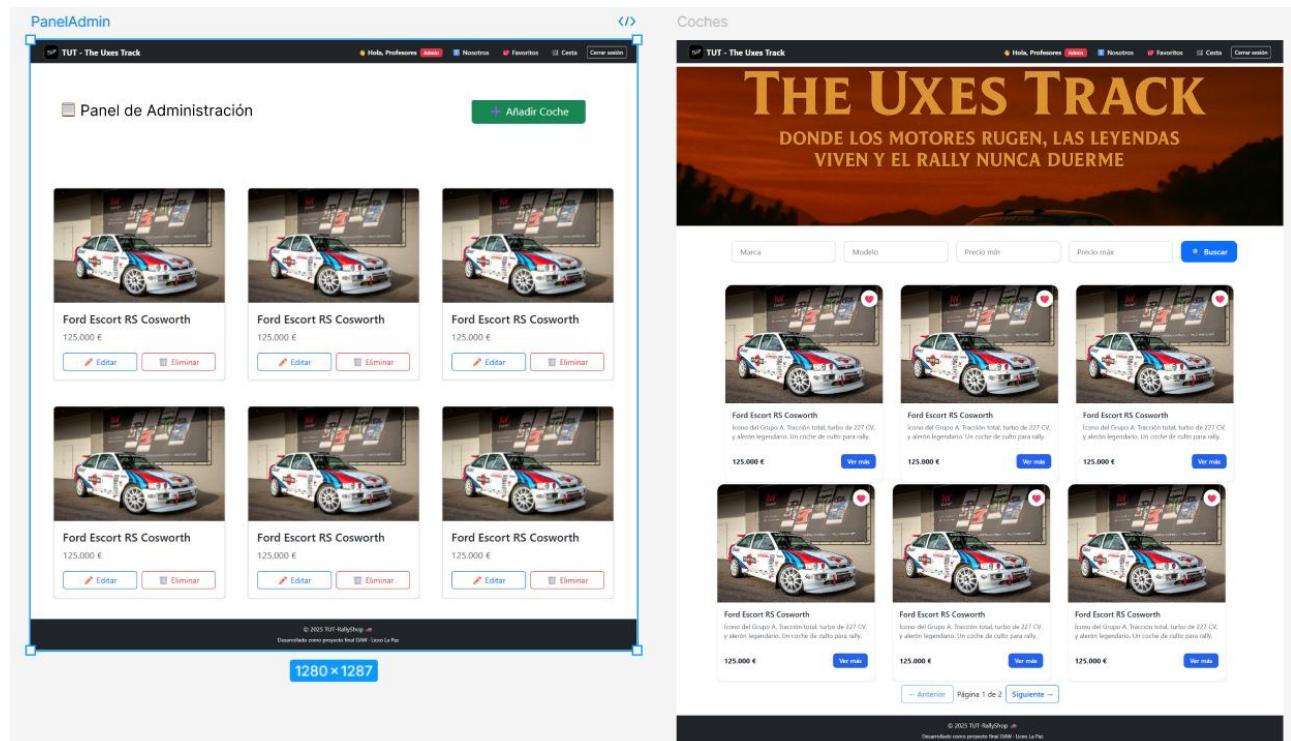
Los wireframes incluyen todas las vistas clave de la aplicación, como la página principal con filtros dinámicos, las tarjetas de productos en formato de rejilla, la vista detallada de cada coche, el sistema de favoritos, el carrito de compras con Stripe, así como el inicio de sesión, registro y el panel de administración.

Un cambio significativo fue la simplificación del encabezado, que ahora organiza de manera ordenada enlaces como 'Nosotros', 'Favoritos', 'Cesta' y el acceso al panel de administración (visible solo para quienes tienen rol de admin). Esto facilita una navegación más clara y lógica para el usuario final.

Respecto a los formularios (inicio de sesión, registro, creación de coches...), se eligió un diseño minimalista y centrado, con campos claros y botones llamativos para mejorar su usabilidad. Se ha dado prioridad al contraste, la legibilidad y el uso adecuado del color para resaltar acciones importantes como confirmar, eliminar o añadir.

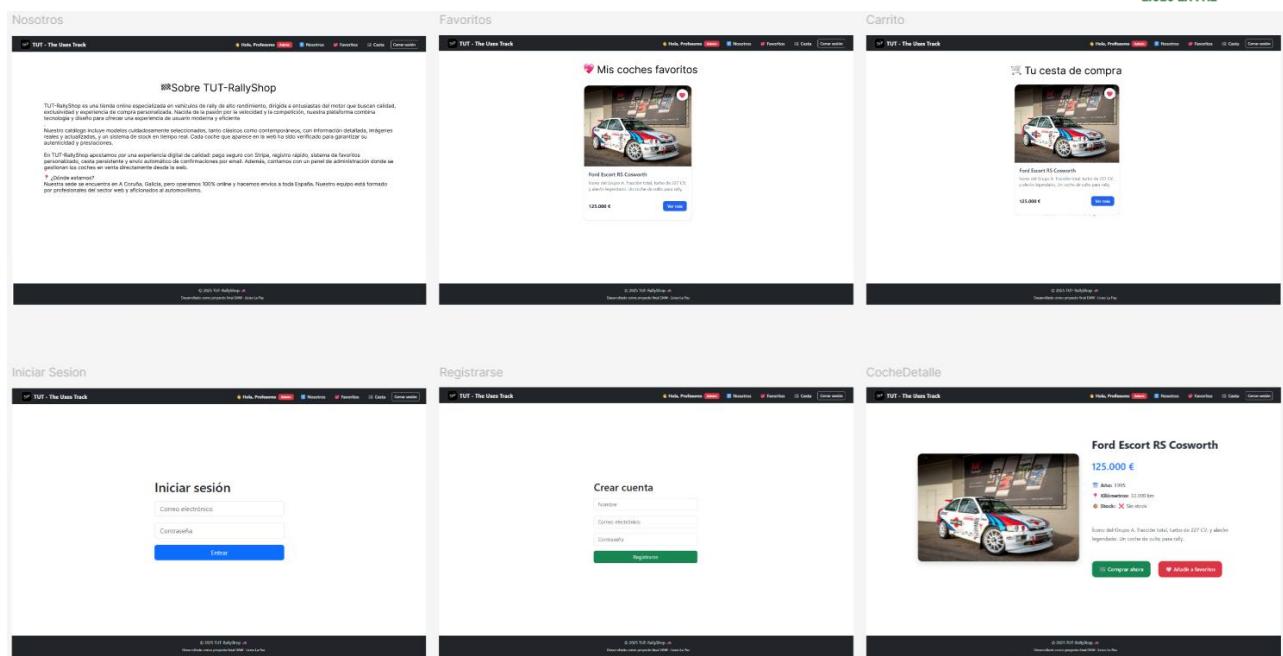
Gracias a los wireframes, se logró planificar de manera ordenada toda la interfaz, validar decisiones antes de desarrollar el código y asegurar que cada elemento cumpla con el objetivo del proyecto: ofrecer una tienda visualmente atractiva, funcional y adaptada a un público real.

Mocks



Figuras 6: Admin y Home(mocks). Elaboración propia.

Proyecto Desarrollo de Aplicaciones Web



Figuras 7:Mocks. Elaboración propia.

A medida que avanzamos en el desarrollo de TUT-RallyShop, el diseño de la interfaz fue transformándose, logrando una experiencia que es tanto limpia como clara y funcional. Todo esto sin perder de vista la estética moderna que una tienda online profesional debe proyectar.

Desde los primeros bocetos y pruebas, la estructura de la plataforma se orientó a maximizar la usabilidad y el atractivo visual, adaptándose a los requerimientos del TFC y al perfil del usuario final. Los mockups fueron creados en Figma, representando cada vista clave de la aplicación: inicio, favoritos, carrito, login, registro, detalles del coche, y panel de administración, entre otros. Estas vistas han servido como una guía esencial para la implementación final, asegurando coherencia visual y estructural en todas las páginas.

A continuación, se resumen los aspectos más relevantes:

Página de Inicio: Esta sección incluye un banner que atrae visualmente, seguido por un sistema de filtros moderno y tarjetas que muestran los coches con imagen, descripción y acciones rápidas. La disposición en grilla y la paginación garantizan una navegación suave.

Barra de Navegación Superior: Fija en la parte superior, permite un acceso directo a secciones como “Favoritos”, “Cesta”, “Nosotros” o el panel de administrador, siempre que el usuario tenga los permisos necesarios. Además, se adapta al estado de login, mostrando los botones adecuados en cada caso.

Login y Registro: Ambas páginas son minimalistas, con formularios sencillos y centrados en el contenido, que cuentan con inputs claros y botones destacados. En la página de registro, se incluye una confirmación visual y gestión de errores.

Vista de Detalle: Al seleccionar un coche, el usuario accede a una ficha completa con todos los detalles técnicos del vehículo: imagen destacada, precio, stock, año, kilómetros y botones para añadir a la cesta o a favoritos.

Proyecto Desarrollo de Aplicaciones Web

Carrito de Compra: Resume todos los coches seleccionados, incluyendo un botón para pagar a través de Stripe, opciones para vaciar el carrito o eliminar coches de manera individual.

Favoritos: Vista sencilla que muestra los coches que el usuario ha marcado como favoritos, con la opción de ver más detalles o eliminarlos de la lista.

Página “Nosotros”: Esta sección es informativa y corporativa, diseñada para reforzar la confianza del usuario al explicar el propósito de la web, con datos de contacto reales y un diseño cuidado.

Panel de Administración: Exclusivo para administradores, este panel permite añadir, editar y eliminar coches. El diseño sigue la misma línea visual del resto del sitio, con botones de acción destacados y tarjetas que incluyen la información principal.

Todo este diseño fue ideado para facilitar la interacción del usuario, transmitir profesionalismo y adaptarse tanto a dispositivos grandes como móviles. El resultado final se alinea con los objetivos del proyecto y las expectativas del usuario final, creando una experiencia de compra completa, visual y efectiva.

Guía de estilos

Colores

	#0d6efd → Azul principal (botones, enlaces activos)
	#198754 → Verde éxito (botón confirmar)
	#dc3545 → Rojo error (botón eliminar)
	#f8f9fa → Fondo claro
	ffffff → Blanco puro (cards, fondo navbar)
	#212529 → Texto principal

Iconos

(❤)	– Favoritos
(🛒)	– Carrito
(🔒)	– Iniciar sesión / Registrarse
(👤)	– Usuario
(🛡)	– Administrador
(📝)	– Editar coche
(🗑)	– Eliminar coche
(➕)	– Añadir coche
(✖)	– Quitar del carrito
(✓)	– Confirmar compra
(👤)	– Página "Nosotros"
(⬅)	– Botón página anterior
(➡)	– Botón página siguiente

Tipografía:

- Fuente principal: Inter o Sans-serif (puedes buscar Inter en Figma)
- Estilos: Regular, Medium, Bold
- Tamaños recomendados:
 - Títulos: 24–32 px
 - Subtítulos: 18–20 px
 - Texto: 14–16 px
 - Detalles / botones: 12–14 px

Botones:

→ Estilo: redondeados border-radius: 12px

→ Botón primario (azul):

- Texto blanco, fondo #0d6efd

[Ver más](#)

→ Botón comprar (borde):

- Texto #0d6efd, borde #0d6efd, fondo blanco

[Comprar ahora](#)

→ Botón eliminar (rojo):

- Texto blanco, fondo #dc3545 →

[Eliminar](#)



Figura 8:Guía de estilos. Elaboración propia.

La guía de estilos de **TUT-RallyShop** define los colores, tipografías e iconos usados en toda la aplicación para mantener una apariencia profesional y coherente.

Colores:

Se usa azul (#0d6efd) como color principal para botones y enlaces, verde (#198754) para confirmar compras y rojo (#dc3545) para eliminar. El fondo es blanco o gris claro, con texto en negro (#212529) para facilitar la lectura.

Tipografía:

La fuente principal es inter, de tipo sans-serif, usada en tamaños adaptados a títulos, textos y botones. Es una fuente moderna y fácil de leer.

Botones:

Los botones tienen bordes redondeados (12px). El botón azul es el principal, el verde se

Proyecto Desarrollo de Aplicaciones Web

usa para comprar y el rojo para eliminar. Todos tienen buen contraste para ser fácilmente reconocibles.

Iconos:

Se usan iconos en los botones y enlaces para representar funciones como favoritos, carrito, editar, eliminar o iniciar sesión, ayudando a que la navegación sea más visual y clara.

Esta guía de estilos se ha implementado desde el principio tanto en los wireframes como en los mockups elaborados en Figma. Gracias a esto, el proyecto ha mantenido una estética clara, intuitiva y bien estructurada en todos sus componentes, alineándose con los objetivos iniciales del diseño y mejorando la experiencia del usuario en cada rincón de la aplicación.

Desarrollo del proyecto

Tecnologías utilizadas

- **Frontend:** React + Vite
- **Backend:** Spring Boot
- **Base de datos:** MySQL
- **Autenticación:** JWT
- **Pasarela de pago:** Stripe
- **Subida de imágenes:** Cloudinary
- **Emails:** Spring Mail (Gmail SMTP)

Estructura del proyecto

Frontend

- **Componentes** organizados en carpetas (Navbar, CocheCard, etc.).
- **Páginas** por vistas (Coches, Carrito, Favoritos, Admin, etc.).
- **Context API** para el carrito de compra.
- **React Router** para navegación de rutas.
- Estilos CSS propios y adaptados a cada componente.

Backend

- **Controladores REST** para manejar peticiones y respuestas.
- **Servicios** con lógica de negocio sin usar interfaces o impl.
- **Repositorios** extendiendo JpaRepository.
- Seguridad configurada con **Spring Security y JWT**, sin sesiones.

Decisiones importantes de desarrollo

Desde el inicio se buscó crear un e-commerce realista, funcional y visualmente atractivo. Se optó por tecnologías modernas que permitieran buena experiencia y escalabilidad.

Se decidió usar **Cloudinary** en vez de almacenamiento local por problemas al subir imágenes desde el panel de administración, y **Stripe** como pasarela por su facilidad de integración. El envío de correos se implementó con Gmail y autenticación por aplicación.

Todo el diseño visual fue ajustado a mano para que fuese **responsive** y atractivo en todas las vistas, con un enfoque en la experiencia del usuario (UX).

Proceso de desarrollo

1. Backend:

- Se empezó por la creación de las entidades (Coche, Usuario, Favorito, Lista deseos, etc.) y relaciones.
- Se implementó la autenticación JWT con protección de rutas.
- Después se añadieron Stripe, Cloudinary y el envío de correos.
- Finalmente se hizo el panel de administración con creación/edición/eliminación de coches.

2. Frontend:

- Se estructuró el proyecto con Vite y se conectó al backend con un axiosConfig con interceptores.
- Se desarrollaron los componentes visuales, cards de coches, favoritos, carrito, login y filtros.
- Se añadió la paginación por URL y botones.
- Se implementó el panel admin y la subida de imágenes.

Pruebas realizadas

- Se probó toda la API desde el frontend (también se usó Postman).
- Se testearon todos los flujos principales: login, registro, añadir a favoritos, añadir al carrito, pagar, gestionar coches desde el panel admin, etc.
- Se forzaron errores (stock 0, sin login, token caducado, etc.) y se controlaron con toasts visuales.

Pros y contras de decisiones

Ventajas:

- JWT permite escalabilidad y mejor integración con frontend moderno.
- Stripe y Cloudinary dan profesionalidad al proyecto.
- Toda la lógica está bien organizada.

Inconvenientes:

- Requiere configuración previa para funcionar (claves, base de datos, etc.).
- El correo por Gmail puede fallar si no se habilitan permisos adecuados.

Manual Administrador

Requisitos previos

Antes de instalar y ejecutar la aplicación **TUT-RallyShop**, es necesario contar con los siguientes elementos instalados en el sistema:

- **Node.js** (v16 o superior)
- **npm** (v6 o superior)
- **Java Development Kit (JDK)** versión 17
- **Maven** (v3.6 o superior)
- **MySQL Server**
- **phpMyAdmin** (opcional, para visualizar la base de datos)

Instalacion del proyecto

1. Clonar repositorio

```
git clone https://github.com/8pabloes/tut-ecommerce.git
```

Backend (Spring boot)

Configurar la base de datos

Accede al archivo application.properties dentro del proyecto backend (ruta: src/main/resources/) y configura tus credenciales de acceso a MySQL:

```
properties
spring.datasource.url=jdbc:mysql://localhost:3306/rallyshop
spring.datasource.username=tu_usuario
spring.datasource.password=tu_contraseña

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

También deberás crear un archivo adicional llamado application-secrets.properties con las claves privadas (Stripe y Gmail):

```
properties
# application-secrets.properties
stripe.secret.key=tu_clave_privada
spring.mail.username=profesores@gmail.com
spring.mail.password=tu_clave_gmail
```

Este archivo no se incluye por seguridad en el repositorio.

Proyecto Desarrollo de Aplicaciones Web

Lanzar backend

Desde la raíz del backend:

```
cd rallyshop
```

```
./mvnw spring-boot:run
```

El backend estará disponible en <http://localhost:8080>.

Frontend (React + Vite)

4. Instalar dependencias

```
cd frontend  
npm install
```

Iniciar servidor de desarrollo

```
npm run dev
```

El frontend se abrirá en <http://localhost:5173>.

Datos de prueba

Para facilitar la evaluación del proyecto, se incluye un archivo llamado `rallyshop.sql` con todos los datos de prueba cargados: coches, usuarios (incluyendo el administrador), favoritos, etc.

Este archivo se encuentra en la raíz del proyecto (junto al backend y frontend).

¿Cómo cargarlo?

PhpMyAdmin

1. Entra a <http://localhost/phpmyadmin>
2. Accede con tu usuario de MySQL.
3. Crea una base de datos llamada `rallyshop` (si no existe).
4. Seleccionala y pulsa en **Importar**.
5. Carga el archivo `rallyshop.sql`.
6. Pulsa **Continuar** y listo.

Con esto, al iniciar sesión como administrador

Correo electrónico: profesores@gmail.com

Contraseña: profesores

Proyecto Desarrollo de Aplicaciones Web

Una vez logueado, se mostrará un botón llamado **"Admin" (en rojo)** en la parte superior derecha de la página. Al pulsarlo, accederás al **panel de administración**, desde el cual podrás:

- Crear nuevos coches (con subida de imágenes a Cloudinary).
- Editar coches existentes.
- Eliminar coches de la base de datos.

Navegación del panel

El panel está organizado con una tabla que lista todos los coches disponibles. Junto a cada coche aparecen botones para editar o eliminarlo. En la parte superior también hay un botón para crear un nuevo coche.

Notas adicionales

- El puerto por defecto para el backend es 8080 y para el frontend es 5173.
- La base de datos debe estar corriendo en el puerto 3306.
- Asegúrate de que las variables de entorno de Stripe y Gmail están bien configuradas antes de lanzar el backend.
- Las imágenes subidas desde el panel se almacenan automáticamente en **Cloudinary** y se muestran en la tienda de forma optimizada.

Manual Usuario

Página de inicio

La página principal da la bienvenida al usuario y presenta visualmente el catálogo de vehículos disponibles.

Elementos:

- **Logo:** Situado en la parte izquierda del header, enlaza directamente a la página de inicio desde cualquier sección.
- **Menú de navegación:**
 - Si el usuario no ha iniciado sesión, se muestran enlaces visibles a "Nosotros", "Iniciar sesión" y "Registrarse".
 - Una vez logueado, se amplía el menú para incluir accesos a "Favoritos", "Cesta", y "Admin" si el rol es administrador.
- **Banner:** Imagen llamativa que se muestra solo en la página principal, dando identidad visual a la aplicación.
- **Filtros:**
 - Buscador visual para filtrar coches por marca, modelo, precio mínimo y máximo.
- **Cards:**
 - Incluyen un botón "Ver más" y un ícono de corazón en la parte superior derecha que permite añadir el coche a favoritos si el usuario está logueado.
 - Incluyen sistema de paginación real por URL para navegar por el catálogo.

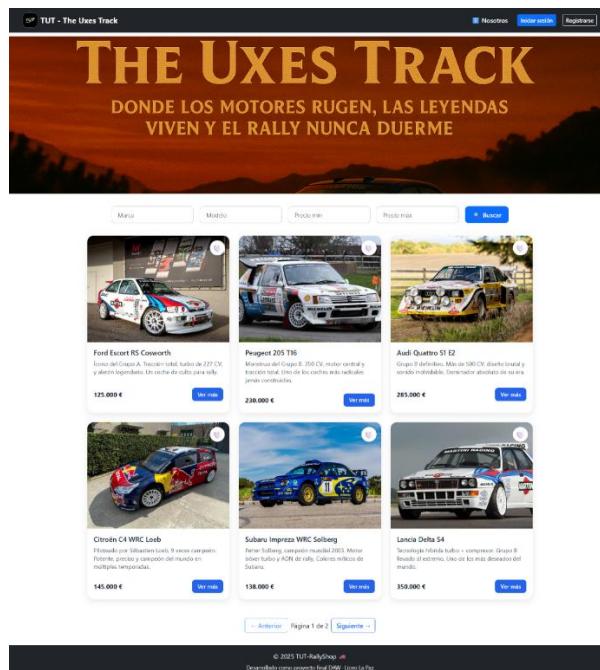


Figura 9: Pantalla de home.



Figura 10: Navbar iniciado sesión.

Detalle de la Publicación

La página de detalle del coche muestra toda la información completa de un vehículo específico. El diseño es limpio y centrado en ofrecer al usuario todos los datos necesarios para valorar una posible compra.

Elementos:

- **Título y descripción:** Se muestra el nombre completo del modelo (marca y modelo) y una descripción técnica con detalles.
- **Imagen principal:** Imagen destacada del coche con calidad alta, que se adapta al ancho de la tarjeta. La imagen es real y representativa del modelo publicado.
- **Información técnica:** Se incluyen todos los campos importantes del coche:
 - Año de fabricación
 - Kilómetros reales
 - Stock disponible
 - Precio actualizado en euros
- **Botón de acción:**
 - Si el usuario no ha iniciado sesión, aparece un mensaje informativo para que se identifique antes de poder comprar o guardar.
 - Si el usuario está logueado:
 - Puede añadir el coche a su carrito ( Comprar ahora).
 - Puede guardarlo en su lista de favoritos ( Añadir a favoritos), los cuales se gestionan desde la vista "Favoritos".

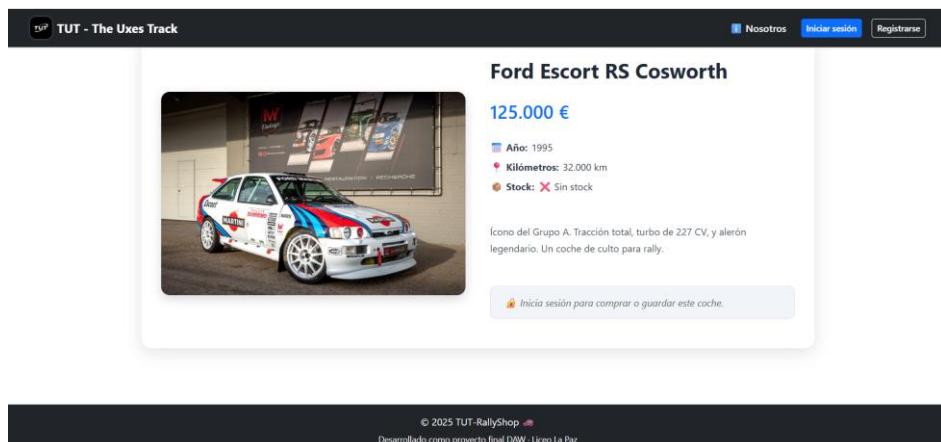


Figura 11: Detalle card.



Figura 12: Botones cuando estas logeado.

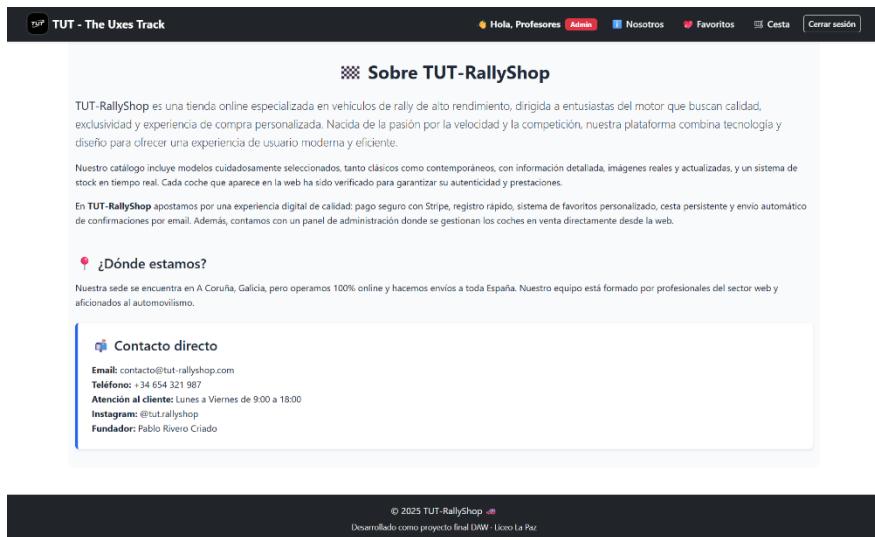
Proyecto Desarrollo de Aplicaciones Web

Página nosotros:

La página “Nosotros” explica de forma clara qué es TUT-RallyShop.

Contenido

- **Presentación:** Se explica que la web ha sido creada como un proyecto profesional, con diseño moderno y funcionamiento real. Se destacan aspectos como la experiencia de usuario, el catálogo visual y los procesos seguros.
- **Ubicación y forma de trabajar:** Aunque el proyecto está desarrollado en A Coruña, funciona 100% online y se simula una tienda que podría operar en toda España.
- **Contacto:** En la parte final de la página se muestran los datos principales: email, teléfono, horario de atención y autor del proyecto.



Sobre TUT-RallyShop

TUT-RallyShop es una tienda online especializada en vehículos de rally de alto rendimiento, dirigida a entusiastas del motor que buscan calidad, exclusividad y experiencia de compra personalizada. Nacida de la pasión por la velocidad y la competición, nuestra plataforma combina tecnología y diseño para ofrecer una experiencia de usuario moderna y eficiente.

Nuestro catálogo incluye modelos cuidadosamente seleccionados, tanto clásicos como contemporáneos, con información detallada, imágenes reales y actualizadas, y un sistema de stock en tiempo real. Cada coche que aparece en la web ha sido verificado para garantizar su autenticidad y prestaciones.

En TUT-RallyShop apostamos por una experiencia digital de calidad: pago seguro con Stripe, registro rápido, sistema de favoritos personalizado, cesta persistente y envío automático de confirmaciones por email. Además, contamos con un panel de administración donde se gestionan los coches en venta directamente desde la web.

>Contacto directo

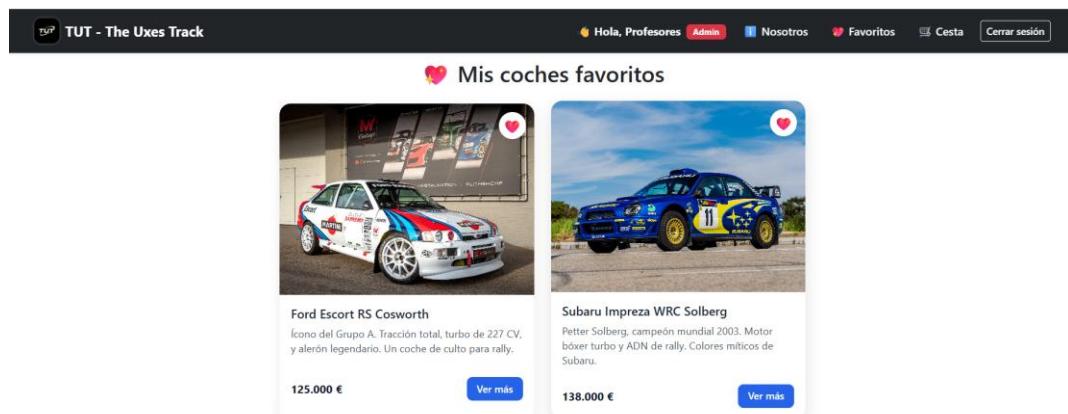
Email: contacto@tut-rallyshop.com
 Teléfono: +34 654 321 987
 Atención al cliente: Lunes a Viernes de 9:00 a 18:00
 Instagram: @tut_rallyshop
 Fundador: Pablo Rivero Criado

© 2023 TUT-RallyShop. Desarrollado como proyecto final DAW - Liceo La Paz.

Figura 13: Nosotros.

Página favoritos:

Esta vista muestra todos los coches que el usuario ha marcado como favoritos. Solo está disponible si has iniciado sesión. Cada tarjeta incluye imagen, nombre, descripción, precio y el botón para acceder al detalle o quitarlo de favoritos.



Mis coches favoritos

 <p>Ford Escort RS Cosworth Ícono del Grupo A. Tracción total, turbo de 227 CV, y alerón legendario. Un coche de culto para rally.</p> <p>125.000 €</p> <p>Ver más</p>	 <p>Subaru Impreza WRC Solberg Petter Solberg, campeón mundial 2003. Motor boxer turbo y ADN de rally. Colores miticos de Subaru.</p> <p>138.000 €</p> <p>Ver más</p>
---	--

Figura 14: Favoritos

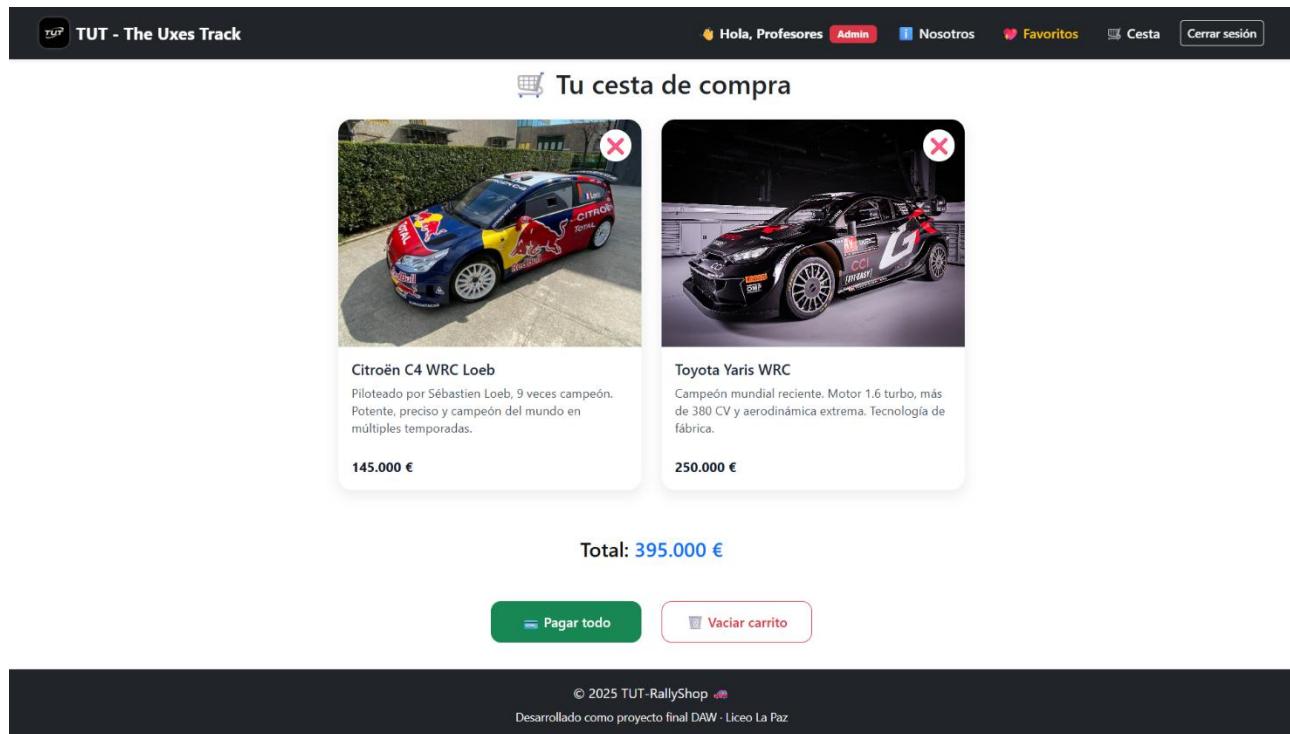
Proyecto Desarrollo de Aplicaciones Web

Página “Cesta”

En esta página el usuario visualiza los coches que ha añadido al carrito antes de realizar la compra.

Elementos:

- **Tarjetas de coches:** Imagen, nombre, descripción, precio y botón para eliminar.
- **Total:** Se muestra el precio total actualizado automáticamente.
- **Botones:** “Pagar todo” (redirige a Stripe) y “Vaciar carrito”.
- **Acceso:** Solo disponible para usuarios logueados.



Tu cesta de compra

Imagen	Nombre	Descripción	Precio
	Citroën C4 WRC Loeb	Pilotado por Sébastien Loeb, 9 veces campeón. Potente, preciso y campeón del mundo en múltiples temporadas.	145.000 €
	Toyota Yaris WRC	Campeón mundial reciente. Motor 1.6 turbo, más de 380 CV y aerodinámica extrema. Tecnología de fábrica.	250.000 €

Total: 395.000 €

[Pagar todo](#) [Vaciar carrito](#)

© 2025 TUT-RallyShop

Desarrollado como proyecto final DAW - Liceo La Paz

Figura 15: Carrito.

Panel administración:

Desde esta vista, los usuarios con rol administrador pueden gestionar todo el catálogo de coches.

Elementos

- **Listado de coches:** Se muestran en formato de tarjetas con imagen, nombre, precio y dos botones: Editar y Eliminar.
- **Botón “Añadir Coche”:** Lleva al formulario para crear una nueva entrada con imagen y datos técnicos.
- **Acceso restringido:** Solo accesible con la cuenta de administrador.

Proyecto Desarrollo de Aplicaciones Web

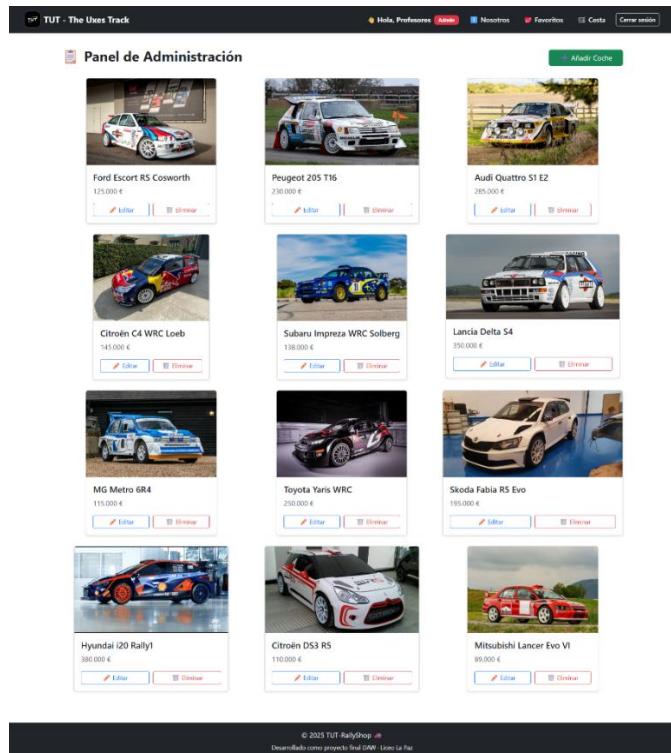


Figura 16: Panel Admin.

Vista Añadir/Editar:

Esta vista permite a los administradores crear o modificar coches directamente desde la web, de forma visual e intuitiva.

Elementos:

- Formulario completo:** Incluye campos para marca, modelo, precio, descripción, stock, año, kilómetros e imagen.
- Subida de imagen:** El administrador selecciona una imagen .jpg que se sube a Cloudinary y se guarda la URL en la base de datos.
- Validación:** Todos los campos son obligatorios. Si falta alguno, no permite guardar.
- Botón “Guardar”:** Inserta o actualiza el coche. Tras la acción, se vuelve automáticamente al panel de administración.

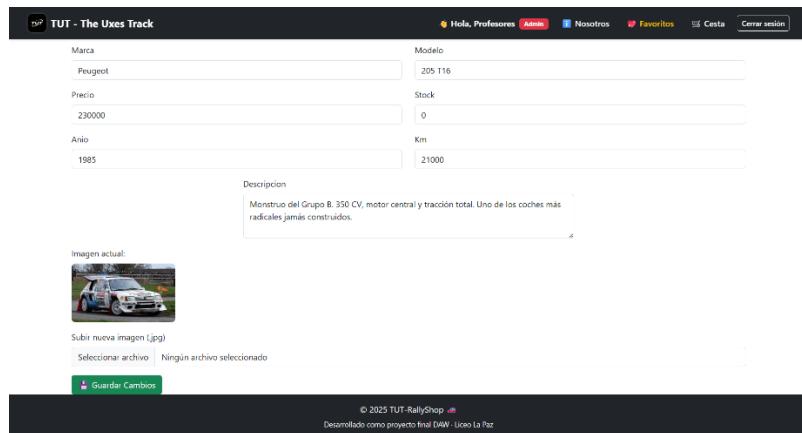


Figura 17: Editar.

Vistas “Iniciar sesión” y “Registrarse”

Estas dos páginas permiten al usuario autenticarse y crear una cuenta en la plataforma de forma segura.

Elementos comunes:

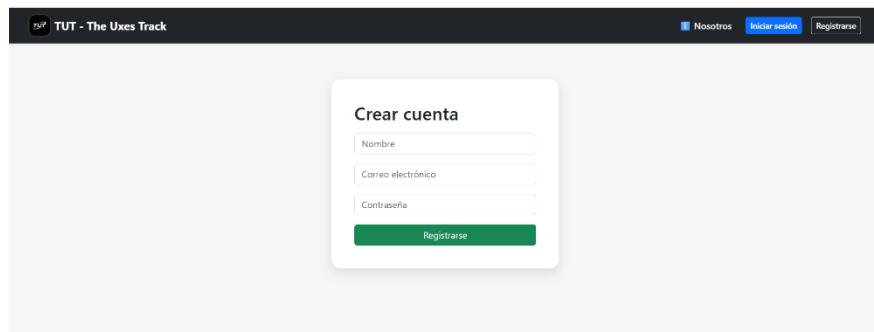
- **Formulario sencillo:** Campos para correo y contraseña (y nombre en registro).
- **Validación:** No permite continuar si los campos están vacíos.
- **Diseño claro y profesional:** Todo centrado, bien espaciado y con mensajes de error visibles.
- **Toasts:** Muestran mensajes tipo “Inicio de sesión correcto” o “Credenciales inválidas”.

Login:

- Al hacer login, se envía una petición al backend y se recibe un token JWT.
- Se guarda el token y los datos del usuario en localStorage.
- Si es admin, redirige con opciones extra.

Registro:

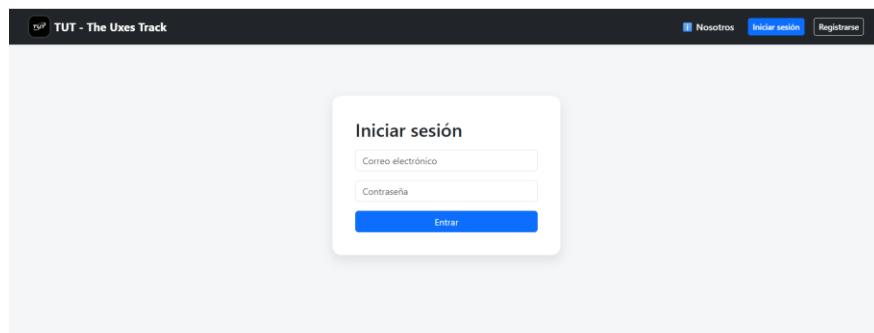
- Registra al usuario en la base de datos y le envía un correo real de bienvenida.
- Tras registrarse, redirige automáticamente al login.



The screenshot shows a registration form titled "Crear cuenta". It contains three input fields: "Nombre" (Name), "Correo electrónico" (Email), and "Contraseña" (Password). Below the fields is a green "Registrarse" (Register) button.



© 2025 TUT-RallyShop 🇺🇸
Desarrollado como proyecto final DAW - Liceo La Paz



The screenshot shows a login form titled "Iniciar sesión". It contains two input fields: "Correo electrónico" (Email) and "Contraseña" (Password). Below the fields is a blue "Entrar" (Enter) button.



© 2025 TUT-RallyShop 🇺🇸
Desarrollado como proyecto final DAW - Liceo La Paz

Figuras 18: Login, Register

Viabilidad tecno-económica.

Análisis del Proyecto

TUT-RallyShop es una aplicación de comercio electrónico que se especializa en la venta de coches de rally. Su diseño busca demostrar que es posible construir un e-commerce funcional y completo desde cero, utilizando herramientas profesionales. Aunque se trata de un proyecto académico, se ha desarrollado con una mentalidad realista, incorporando servicios como Stripe, Cloudinary, JWT, y una base de datos relacional. Esta sección evalúa la viabilidad técnica y económica del proyecto, en caso de que se considere un despliegue real y a gran escala.

Análisis DAFO

Fortalezas

Empleo de tecnologías modernas y gratuitas como React + Vite, Spring Boot y MySQL. El proyecto se desarrolló completamente desde cero, lo que permite tener un control total sobre el código. Se realizó una integración real de servicios como Stripe y Cloudinary, además de contar con un panel de administración funcional y un sistema de pagos activo. La experiencia de usuario ha sido cuidadosamente diseñada, siendo responsive y con características como filtros, favoritos y carrito de compras.

Debilidades

El proyecto ha sido desarrollado por una sola persona, lo que implica recursos limitados. Existe también una dependencia de servicios externos como Cloudinary y Stripe. Aunque la base de datos y la seguridad son funcionales, no están optimizadas para manejar alta concurrencia. Por último, no se ha realizado un despliegue real en la nube, ya que solo está funcionando en un entorno local.

Oportunidades

Hay una posibilidad real de escalar la aplicación hacia una tienda de nicho centrada en el automovilismo deportivo. El modelo es fácilmente extensible a otros productos y categorías. Además, hay potencial para monetizar a través de la venta real de coches, merchandising o contenidos premium.

Amenazas

El sector de la automoción en línea presenta una alta competencia. Además, existen costes asociados a la escalabilidad real, como hosting, servicios premium y seguridad. También

Proyecto Desarrollo de Aplicaciones Web

hay que considerar los cambios en las políticas o precios de servicios como Cloudinary o Stripe.

Coste de Implementación Estimado

Concepto	Coste Inicial	Coste Mensual
Ordenador para desarrollo	1.000 €	-
Cloudinary (plan profesional)	-	89 €
Hosting (servidor backend)	-	20 €
Base de datos en la nube	-	15 €
Stripe (comisión por venta)	-	variable*
Dominio + HTTPS	20 €	5 €
Soporte / mantenimiento	-	100 €
Total estimado	1.020 €	229 € + ventas

Figura 19: Tabla comparativa de costes. Elaboración propia.

*Stripe cobra aproximadamente el 1,4% por transacción con tarjeta europea.

Entiendo que deseas ajustar los escenarios de rentabilidad para que reflejen de manera más realista las expectativas de ventas en una tienda online de coches de alta gama como TUT-RallyShop. Dado que los vehículos tienen precios elevados (a partir de 50.000 €) y considerando que el proyecto aún no ha sido lanzado al mercado, es prudente establecer proyecciones conservadoras.

Escenarios de Rentabilidad Realistas

Escenario Pesimista

- **Usuarios mensuales:** 100
- **Pedidos mensuales:** 0
- **Ingresos:** 0 €
- **Gastos:** Costes fijos de servicios como hosting, dominios, y mantenimiento.
- **Resultado:** Pérdidas mensuales mientras se busca tracción en el mercado.

Escenario Realista

- **Usuarios mensuales:** 300
- **Pedidos mensuales:** 1 a 2
- **Ticket medio:** 55.000 €
- **Ingresos brutos mensuales:** 55.000 € a 110.000 €
- **Comisión de Stripe (aprox. 1.5%):** 825 € a 1.650 €
- **Gastos adicionales:** Publicidad, mantenimiento, soporte.
- **Beneficio neto estimado:** Margen reducido o equilibrio financiero.

Proyecto Desarrollo de Aplicaciones Web

Escenario Optimista

- **Usuarios mensuales:** 1.000
- **Pedidos mensuales:** 3 a 4
- **Ingresos brutos mensuales:** 165.000 € a 220.000 €
- **Comisión de Stripe:** 2.475 € a 3.300 €
- **Gastos adicionales:** Publicidad, mantenimiento, soporte.
- **Beneficio neto estimado:** Posibilidad de reinversión en marketing y mejoras técnicas.dojobusiness.com+6mhp.com+6reddit.com+6

Coste de Adquisición de Cliente (CAC) Estimado

- **Inversión en publicidad inicial:** 300 €
- **Usuarios adquiridos:** 100
- **CAC:** 3 €/usuario
- **Conversión estimada:** Si 1 de cada 50 usuarios realiza una compra, el retorno comienza a ser positivo.

Conclusión

El desarrollo técnico de TUT-RallyShop ha demostrado ser viable en un entorno local y académico. Para escalar a producción, se requerirá una estrategia de marketing digital enfocada y posiblemente la diversificación de productos (como merchandising o servicios relacionados) para aumentar la rentabilidad. Aunque las ventas iniciales puedan ser bajas, una base sólida y una propuesta de valor clara pueden posicionar la tienda en un nicho específico del mercado automovilístico.

Trabajo futuro.

A pesar de que TUT-RallyShop ya se presenta como una herramienta funcional y completa para el ámbito académico, hay un amplio campo para mejorar y expandirse si se busca transformarlo en un producto más profesional, escalable y competitivo.

A continuación, se enumeran las principales áreas que podrían mejorarse y las funcionalidades que se podrían desarrollar en el futuro:

1. Implementación completa de la Lista de Deseos

- **Objetivo:** Utilizar la estructura ya establecida en la base de datos para crear una lista de deseos operativa.
- **Acciones específicas:** Activar la tabla lista_deseos para que los usuarios puedan guardar autos que aún no están disponibles o que desean seguir más adelante. Diferenciar visualmente entre los autos añadidos a favoritos y los que están en la lista de deseos. Ofrecer notificaciones (por email u otros medios) cuando un auto vuelva a estar disponible.

2. Mejora visual del panel de administración

- **Objetivo:** Hacer que el área de gestión de autos sea más intuitiva y visual.
- **Acciones específicas:** Incorporar paginación, filtros y un buscador al panel de administración. Permitir una edición más dinámica de los autos mediante formularios interactivos. Incluir una vista de estadísticas (número de autos, favoritos, ventas...).

3. Mejora de seguridad y autenticación

- **Objetivo:** Reforzar la protección de datos y el acceso a la plataforma.
- **Acciones específicas:** Implementar autenticación multifactor (MFA). Controlar la caducidad y renovación automática del token JWT. Incorporar registros de actividad para usuarios administradores.

4. Internacionalización y accesibilidad

- **Objetivo:** Ampliar el acceso a la aplicación para más usuarios y contextos.

Proyecto Desarrollo de Aplicaciones Web

- **Acciones específicas:** Agregar soporte multilingüe (español/inglés como mínimo). Aplicar normas de accesibilidad web (WAI-ARIA). Optimizar contrastes, tamaños y la navegación por teclado.

5. Despliegue en la nube y escalado

- **Objetivo:** Preparar el proyecto para un entorno de producción real.

- **Acciones específicas:** Desplegar el backend y frontend en un servicio de alojamiento en la nube (ej. Vercel, Railway, Heroku, AWS...). Configurar una base de datos MySQL remota y respaldos automáticos. Implementar almacenamiento de imágenes en un bucket propio (S3 u otro) si se supera el plan gratuito de Cloudinary.

6. Nuevas funcionalidades para usuarios

- **Objetivo:** Mejorar la experiencia de compra y la fidelización.

- **Acciones específicas:** Crear un sistema de valoraciones y reseñas por auto. Implementar un historial de pedidos y seguimiento. Ofrecer recomendaciones automáticas por modelo o marca similar.

7. Estrategia de crecimiento y monetización

- **Objetivo:** Ampliar el proyecto hacia un modelo de negocio real.

- **Acciones específicas:** Integrar métodos de pago adicionales como Bizum o PayPal. Lanzar campañas de marketing digital para promocionar autos destacados. Ofrecer cuentas 'Pro' para vendedores particulares.

Este plan de trabajo permitirá a TUT-RallyShop transformarse de un proyecto académico a una tienda online real, bien posicionada para el sector automotriz de nicho. Con su estructura actual, modular y bien organizada, la aplicación tiene el potencial de escalar sin necesidad de rehacer su base, facilitando la implementación gradual de todas estas mejoras.

Conclusiones.

El desarrollo de TUT-RallyShop ha sido un desafío tanto técnico como personal que me ha permitido aplicar todo lo que he aprendido en el ciclo de Desarrollo de Aplicaciones Web. A lo largo del proyecto, he conseguido alcanzar la mayoría de los objetivos que me propuse desde el principio: construir una tienda online que no solo sea funcional y segura, sino que también luzca bien y ofrezca características útiles como autenticación, un sistema de favoritos, pasarela de pago, subida de imágenes, un panel de administración y control de stock.

Una de las partes más gratificantes ha sido integrar Cloudinary para gestionar las imágenes y utilizar Stripe para procesar los pagos. También fue un gran logro implementar un sistema de autenticación con JWT y desarrollar un panel de administración con roles diferentes, lo que añade una capa extra de control y seguridad. Todo esto lo he creado desde cero, sin recurrir a plantillas o CMS, escribiendo cada línea de código para entender y manejar a fondo cómo funciona el sistema.

No obstante, como es de esperar en un proyecto de esta magnitud, no todo se ha podido completar al 100%. Por ejemplo, la funcionalidad de lista de deseos quedó pendiente de implementación a pesar de que la estructura ya está lista en la base de datos. También me habría gustado desplegar la aplicación de forma completa en la nube, automatizar ciertas tareas y perfeccionar aún más la interfaz visual del panel de administración.

A pesar de las limitaciones de tiempo y recursos, estoy realmente orgulloso del resultado. Este proyecto no solo refleja años de formación técnica, sino también el esfuerzo, la constancia y la evolución que he experimentado como desarrollador. He dedicado muchas horas, resuelto errores, aprendido en el camino y mejorado mi forma de trabajar. TUT-RallyShop ha sido una experiencia intensa, pero muy gratificante.

Me ha permitido descubrir de lo que soy capaz cuando me propongo un objetivo y me organizo adecuadamente. Aunque el proyecto tiene un amplio margen de mejora, lo que he logrado construir es una base sólida, real y útil. Y eso, para mí, ya representa un éxito.

Anexos.