

RENTME - Aplicación web de renting de vehículos

1. Descripción y justificación del proyecto

1.1 Descripción

Este proyecto consiste en el desarrollo de una **aplicación web de renting de vehículos**, destinada a ofrecer a los usuarios una plataforma donde puedan alquilar diferentes tipos de vehículos de forma rápida, sencilla y segura.

A través de esta plataforma, los clientes podrán:

- Registrarse y autenticarse.
- Consultar la disponibilidad de vehículos.
- Seleccionar el vehículo que se adapte a sus necesidades.
- Realizar reservas.
- Gestionar sus alquileres desde su cuenta personal.

La aplicación también contará con un **panel de administración**, desde el cual se podrá:

- Gestionar el catálogo de vehículos.
- Revisar las reservas activas.
- Modificar configuraciones.
- Aplicar la lógica de negocio del servicio.

1.2 Justificación

La idea de este proyecto surge debido a la creciente demanda de digitalización en el sector del alquiler de vehículos, donde muchas empresas buscan ofrecer una experiencia de usuario más fluida y moderna.

Este proyecto permite:

- Aplicar conocimientos adquiridos como el desarrollo backend con **Laravel**, el diseño de bases de datos con **MySQL** y el despliegue local mediante **Apache2**.
- Adquirir nuevas habilidades como el desarrollo frontend con **React** y la implementación de un sistema completo con autenticación y roles.

Además, la aplicación tiene potencial para evolucionar y escalar con nuevas funcionalidades en el futuro.

1.3 Alcance

 **Funcionalidades incluidas:**

- Registro e inicio de sesión de usuarios.
- Vista del catálogo de vehículos, filtrable por tipo, precio y disponibilidad.
- Proceso de reserva de vehículos.

- Panel de usuario con historial de reservas.
- Panel de administración para gestión de vehículos, usuarios y reservas.
- Sistema de roles: cliente / administrador.

❌ Funcionalidades fuera de alcance (por ahora):

- Integración de pagos online.
- Notificaciones por correo electrónico o SMS.

1.4 Valoración de alternativas existentes

Algunas plataformas similares en el mercado:

- Rentalcars: Muy completa pero compleja y orientada a múltiples proveedores.
- Enterprise Rent-A-Car: Opción robusta con integración empresarial.
- Zity / Wible: Apps de carsharing enfocadas al alquiler por minutos.

RENTME busca ofrecer una solución más sencilla, intuitiva y adaptable, pensada para un negocio de tamaño medio o para autónomos con flota propia.

1.5 Stack tecnológico elegido

Tecnología	Uso	Justificación
React	Frontend	SPA rápida y modular, ideal para experiencias de usuario fluidas.
Laravel	Backend	Framework robusto, seguro y con herramientas integradas como autenticación y Eloquent.
MySQL	BBDD	Relacional, eficiente para este tipo de aplicación.
Apache2	Servidor	Compatible con Laravel y comúnmente usado en despliegues locales.
Github	Control de versiones	Facilita el trabajo iterativo y el seguimiento por parte del profesorado.

2. Objetivos, requisitos y casos de uso

2.1 Objetivos del proyecto

- Desarrollar una aplicación web funcional para la gestión de alquiler de vehículos.
- Implementar un sistema de autenticación con roles (clientes y administradores).
- Diseñar una interfaz intuitiva y responsive.
- Aplicar las buenas prácticas de desarrollo en frontend, backend y base de datos.
- Documentar todo el proyecto y facilitar su seguimiento mediante GitHub.

2.2 Requisitos del proyecto

Funcionales:

- Registro e inicio de sesión de usuarios.
- Vista del catálogo de vehículos.
- Filtro por tipo de vehículo, disponibilidad y precio.
- Proceso de reserva: selección de vehículo, fechas, confirmación.
- Gestión de reservas desde el perfil del cliente.
- Panel de administración con CRUD de vehículos.
- Asignación de roles y control de accesos.

No funcionales:

- Contraseñas cifradas (bcrypt).
- Validaciones tanto en frontend como backend.
- Disponibilidad: acceso 24/7 desde cualquier navegador moderno.
- Eficiencia: carga rápida, navegación fluida.

Requisitos de interfaz:

- Diseño adaptado a dispositivos móviles (responsive).
- Menú claro con accesos diferenciados por rol.
- Formularios validados y accesibles.
- Feedback visual ante acciones del usuario (éxito/error).

2.3 Casos de uso principales

Caso de uso 1: Registro de cliente

- El usuario accede al formulario de registro.
- Introduce sus datos y envía.
- El sistema valida y crea la cuenta.

Caso de uso 2: Realizar reserva

- El usuario inicia sesión y accede al catálogo.
- Filtra los vehículos y selecciona uno disponible.
- Define fecha de inicio y fin del alquiler.
- Confirma la reserva.
- El sistema actualiza la disponibilidad.

Caso de uso 3: Gestión de flota (Administrador)

- El administrador inicia sesión en su panel.
- Puede añadir, modificar o eliminar vehículos.
- Visualiza las reservas activas e historial.



3. Modelo y diseño de la base de datos

(Ejemplo de tablas básicas)

users -> id, name, email, password, role (cliente/admin), created_at

vehicles -> id, marca, modelo, tipo, precio_dia, disponible, imagen_url

reservations -> id, user_id, vehicle_id, fecha_inicio, fecha_fin, estado, total_precio

Relaciones:

- Un usuario puede tener muchas reservas.
- Un vehículo puede tener muchas reservas (con fechas no solapadas).
- Cada reserva pertenece a un vehículo y un usuario.



4. Metodología y control de versiones

Se utilizará una metodología iterativa, planificando varias entregas y revisiones, basadas en avances funcionales.

Se usará Git y GitHub para versionar el código, documentar cada iteración mediante commits descriptivos y permitir que el profesorado supervise el desarrollo.



Última actualización: Abril 2025



Repositorio pendiente de subida para seguimiento por el profesorado