



CENTRO UNIVERSITARIO TECNOLÓGICO CEUTEC

INFORME DE PROYECTO FINAL

ASIGNATURAS: DESARROLLO DE APLICACIONES WEB I

SECCION: T92

INTEGRANTES:

Wilson Odiel Flores Herrera T31751073

Luis Fernando Lanza Amador T32411207

Wilmer Javier Sánchez T32421252

CATEDRATICO: Jonie Esteban Miralda Cruz

FECHA: 26/03/2025

INTRODUCCION

El proyecto "Autolote" es una aplicación web diseñada para la gestión eficiente de un lote de autos, permitiendo el registro, administración y control de los vehículos disponibles. Está desarrollado con un backend en Node.js utilizando Express.js, una base de datos MySQL y autenticación mediante JSON Web Tokens (JWT).

Además, se ha implementado un control de versiones mediante Git con la metodología Gitflow, asegurando un desarrollo estructurado y colaborativo. Asimismo, se permite la integración con APIs externas, ya sean públicas o privadas, facilitando el acceso a información adicional relevante, como datos de vehículos o cotizaciones de mercado.

Tecnologías Utilizadas

Backend

Node.js con Express.js

Framework ligero y eficiente para la gestión del servidor, manejo de rutas y lógica de negocio.

Base de Datos

MySQL

Sistema de gestión de bases de datos relacional para el almacenamiento y organización de la información sobre autos y usuarios.

Autenticación

JSON Web Tokens (JWT)

Método seguro para la autenticación y gestión de sesiones de los usuarios.

Control de Versiones

Git con Gitflow

Estructura para el manejo de ramas y desarrollo colaborativo del proyecto.

Características Principales

- Gestión de usuarios con diferentes roles (Admin, Vendedor, Cliente)
- Autenticación mediante JWT para garantizar sesiones seguras
- Gestión completa de vehículos con operaciones CRUD (Crear, Leer, Actualizar, Eliminar)
- Registro de clientes y seguimiento de sus consultas
- Registro y seguimiento de ventas con detalles y tasas de cambio
- Tasas de cambio para ventas en diferentes monedas (Euros, Yen)

Requisitos Previos

- Node.js: Para ejecutar el servidor.
- MySQL: Sistema de base de datos relacional.
- Postman: Herramienta para realizar pruebas de la API.

Estructura del Proyecto

La estructura del proyecto está organizada de la siguiente manera para asegurar modularidad y facilidad de mantenimiento:

autolote/

```
|— config/
|   |— db.js           # Configuración de la base de datos
|— middleware/
|   |— authMiddleware.js # Middleware de autenticación
|— routes/
|   |— authUsers.js     # Rutas de autenticación y usuarios
|   |— vehiculos.js     # Rutas para gestión de vehículos
|   |— clientes.js      # Rutas para gestión de clientes
|   |— ventas.js        # Rutas para gestión de ventas
|— utils/
|   |— currencyConverter.js # Rutas de API de terceros para tasas de cambio
|— .env                # Variables de entorno
|— app.js              # Punto de entrada de la aplicación
|— package.json        # Dependencias y scripts
|— README.md           # Documentación del proyecto
```

Funcionamiento de la Aplicación

Autenticación

- POST /api/login: Iniciar sesión.
- POST /api/usuarios/registro: Registrar un nuevo usuario.

Usuarios

- GET /api/usuarios: Obtener todos los usuarios (solo accesible para admins).
- GET /api/usuarios/:id: Obtener un usuario por su ID.
- PUT /api/usuarios/:id: Actualizar la información de un usuario.
- DELETE /api/usuarios/:id: Eliminar un usuario (solo accesible para admins).

Vehículos

- GET /api/vehiculos: Obtener todos los vehículos disponibles.
- GET /api/vehiculos/:id: Obtener un vehículo específico por su ID.
- POST /api/vehiculos: Crear un nuevo vehículo (solo accesible para admins y vendedores).
- PUT /api/vehiculos/:id: Actualizar la información de un vehículo (solo accesible para admins y vendedores).
- DELETE /api/vehiculos/:id: Eliminar un vehículo (solo accesible para admins).

Clientes

- GET /api/clientes: Obtener la lista de clientes registrados (accesible para admins y vendedores).
- GET /api/clientes/:id: Obtener un cliente específico por su ID.
- POST /api/clientes: Registrar un nuevo cliente.
- PUT /api/clientes/:id: Actualizar la información de un cliente.
- DELETE /api/clientes/:id: Eliminar un cliente (solo accesible para admins).

Consultas

- POST /api/consultas: Registrar una nueva consulta de cliente.
- GET /api/consultas: Obtener una lista de consultas, con filtros opcionales.

Ventas

- POST /api/ventas: Registrar una nueva venta, incluyendo las tasas de cambio en Euros, Lempiras y Yen.
- GET /api/ventas: Obtener un listado de todas las ventas (accesible para admins y vendedores).

Dificultades

- Integración de JWT: Implementar y asegurar la autenticación mediante JSON Web Tokens requirió pruebas exhaustivas para evitar vulnerabilidades.
- Conexión con MySQL: La configuración y optimización de la base de datos presentó retos en la definición de esquemas y manejo de consultas eficientes.
- Consumo de APIs Externas: Adaptar la estructura del código para integrar datos de fuentes externas sin afectar el rendimiento de la aplicación.
- Coordinación en el trabajo en equipo: La implementación del flujo de trabajo con Gitflow demandó una adecuada planificación para evitar conflictos en el código.

Conclusión

Este sistema de gestión de vehículos y clientes está diseñado para proporcionar una solución integral, que cubre desde la autenticación de usuarios hasta el manejo detallado de vehículos, clientes y ventas. La implementación de JWT garantiza una seguridad robusta para las sesiones de usuario, y la estructura modular permite un fácil mantenimiento y escalabilidad del sistema.

Este proyecto está listo para ser extendido con nuevas funcionalidades y mejorar la experiencia del usuario mediante el uso de herramientas como tasas de cambio en tiempo real y un diseño basado en roles.

Imágenes de pruebas del proyecto







