



TRABAJO DE FIN DE CICLO

CURSO ACADÉMICO 2023/2025

CONVOCATORIA JUNIO

CENTRO EDUCATIVO: IES CASTELAR

SWAPLT

Alumno: Jesús Carrasco Toscano

DNI: 80232744-G

Ciclo: Desarrollo de Aplicaciones Web

Tutor del curso: Juan José Perera Villalba

Tutor del proyecto: Luis Ferreira Gordillo

Fecha de entrega: 01/06/20



Índice

1. [Análisis de Contexto](#)
 - [1.1 Detección de necesidades que justifiquen la utilidad del proyecto](#)
 - [1.2 Estudio de mercado y potenciales clientes](#)
2. [Diseño del Proyecto](#)
 - [2.1 Coste de implantación del proyecto](#)
 - [2.2 Plataforma de ejecución y requisitos](#)
 - [2.3 Aplicación de la Ley de Protección de Datos](#)
 - [2.4 Metodología de desarrollo](#)
 - [2.5 Esquema de base de datos](#)
3. [Organización del Proyecto](#)
 - [3.1 Planning de ejecución del proyecto \(Marzo - Junio 2025\)](#)
 - [3.2 Recursos utilizados para su desarrollo](#)
4. [Manual del Usuario](#)
 - [4.1 Preguntas Frecuentes](#)
 - [4.2 Guía de Uso de la Plataforma](#)
5. [Manual del Programador](#)
 - [5.1 Sistema Operativo](#)
 - [5.2 Entorno de Desarrollo](#)
 - [5.3 Tecnologías utilizadas](#)
 - [5.4 Estructura de Carpetas](#)
 - [5.5 Base de Datos](#)
 - [5.6 Despliegue en Producción](#)

1. Análisis de Contexto

1.1. Detección de necesidades que justifiquen la utilidad del proyecto

En el mercado actual de compraventa de vehículos de segunda mano, muchas de las operaciones siguen dependiendo de intermediarios como casas de compraventa, que adquieren los vehículos a precios muy por debajo de su valor real y los revenden con márgenes excesivos. Esto deja a los usuarios particulares con escaso control sobre el valor de su vehículo y con pocas opciones para realizar transacciones de forma directa y justa.

SWAPLT nace como respuesta a esta problemática, ofreciendo una plataforma nacional pensada para que los usuarios puedan comprar y vender vehículos entre particulares, sin la intervención de intermediarios. El objetivo es devolver el control a los propios usuarios, facilitando un proceso transparente, ágil y sin comisiones, manteniendo al mismo tiempo la seguridad y la claridad en la comunicación entre las partes.

1.2. Estudio de mercado y potenciales clientes a los que les generaría valor añadido

• Tipo de clientes o empresas donde se implantará el proyecto

Este proyecto podrá venderse y podrá ser implantado principalmente entre los siguientes tipos de usuarios:

1. Particulares que desean vender su vehículo directamente sin recurrir a concesionarios o casas de compraventa.
2. Compradores particulares que buscan adquirir vehículos de manera directa y sin comisiones.
3. Pequeños negocios o talleres locales que quieran publicar vehículos reacondicionados o de ocasión sin pasar por grandes plataformas.

- **Utilidad o beneficio que encontrarán los clientes con la implantación y explotación del proyecto**

Los clientes que adquieran este software podrán beneficiarse de diversas funcionalidades que simplifican y mejoran el proceso de compraventa entre particulares:

- Comunicación directa entre vendedor y comprador mediante mensajería integrada.
- Eliminación total de comisiones, al no haber intermediarios en las transacciones.
- Geolocalización de vehículos mediante integración con Google Maps, lo que permite conocer exactamente la ubicación del producto antes de concertar una visita.
- Posibilidad de bloquear usuarios molestos o no deseados, lo que mejora la experiencia de navegación y protege la privacidad del usuario.
- Interfaz sencilla e intuitiva, pensada para usuarios no técnicos.

Aunque SWAPLT aún está en sus primeras fases, se plantea como una alternativa real y en crecimiento frente a grandes plataformas del sector. Se encuentra en evolución constante y con una hoja de ruta que contempla mejoras de posicionamiento, seguridad y funcionalidades avanzadas, con el objetivo de ofrecer un valor añadido que fidelice a sus usuarios.

2. Diseño del proyecto

2.1. Coste de implantación del proyecto

El desarrollo del proyecto comenzó el 15 de marzo de 2025 y ha sido llevado a cabo íntegramente por un único desarrollador. Si bien no se ha registrado con precisión el número total de horas trabajadas, se estima que ha requerido una dedicación constante a lo largo de varios meses.

En términos de costes directos, el proyecto se ha ejecutado en una infraestructura completamente gratuita:

- El dominio se gestiona mediante DuckDNS, lo cual no genera coste alguno.
- El despliegue se ha realizado sobre una instancia gratuita de Google Cloud Platform (GCP), por lo que no se incurre en gastos de servidor.

Coste estimado profesional (si fuese un encargo comercial):

- Desarrollo backend y frontend: $\sim 180 \text{ horas} \times 20 \text{ €/h} \rightarrow \mathbf{3.600 \text{ €}}$
- Documentación, pruebas y despliegue: $\sim 40 \text{ horas} \times 20 \text{ €/h} \rightarrow \mathbf{800 \text{ €}}$

Total estimado del proyecto si fuera contratado a un desarrollador freelance:
 $\approx 4.400 \text{ €}$

2.2. Plataforma de ejecución y requisitos

La plataforma se encuentra desplegada en una **VPS gratuita de Google Cloud**, utilizando el sistema operativo **Ubuntu 20.04 LTS**. Tanto el backend como el frontend están implementados y ejecutándose en este mismo entorno. Se ha utilizado **Docker** para facilitar la contenedorización y despliegue del sistema.

Requisitos del servidor:

- Sistema operativo: Linux (Ubuntu 20.04)
- Memoria RAM: 1 GB + 2 GB de SWAP
- CPU: 2 vCPU
- Almacenamiento: 30 GB

Requisitos del cliente:

- Navegador web moderno (Chrome, Firefox, Edge, Safari...)
- Conectividad a Internet

El software está completamente adaptado para funcionar en **navegadores web** tanto de **ordenadores** como de **dispositivos móviles**, sin necesidad de instalación adicional.

2.3. Aplicación de la Ley de Protección de Datos

En cumplimiento con el **Reglamento General de Protección de Datos (RGPD)**, SWAPLT ha sido diseñado minimizando la recogida de datos personales. Los datos tratados se limitan a:

- Nombre del usuario
- Correo electrónico
- Imagen de perfil (proporcionada por la cuenta de Google)

Los usuarios inician sesión únicamente a través de su cuenta de Google, por lo que se garantiza una capa de seguridad adicional. Las contraseñas en la base de datos están cifradas mediante **algoritmos hash** robustos.

Aunque actualmente **no se requiere aceptar términos y condiciones para registrarse**, estos se encuentran disponibles en el footer del sitio, junto a la política de privacidad y la sección de ayuda.

Además, se implementan **mecanismos de control de acceso** para restringir funcionalidades según el rol o propiedad del usuario. Se garantiza así que los datos privados no sean accesibles por usuarios no autorizados.

2.4. Metodología de desarrollo

Se ha seguido una metodología adaptativa centrada en el desarrollo iterativo, priorizando primero el backend para asegurar una API funcional antes de integrar el frontend:

1. Análisis inicial del problema y objetivos.
2. Desarrollo completo del **backend**, asegurando primero la lógica del negocio y las rutas API.
3. Posterior diseño e implementación del **frontend**, con especial cuidado en la experiencia de usuario.
4. Conexión entre frontend y backend.
5. Pruebas, despliegue y puesta en producción.
6. Redacción de documentación y mejoras incrementales.

2.5. Esquema de base de datos

El diseño de la base de datos ha sido realizado utilizando la herramienta **Mermaid**, con un enfoque relacional. Las tablas más relevantes incluyen:

- usuarios: contiene los datos de registro e identificación de los usuarios.
- vehiculos: almacena información sobre los vehículos anunciados.
- anuncios: relaciona usuarios con vehículos y contiene los datos del anuncio.
- mensajes: permite la comunicación entre usuarios.
- favoritos: permite a los usuarios guardar vehículos de interés.
- bloqueos: gestiona el sistema de bloqueo de usuarios.

Se adjuntará el esquema visual en formato Mermaid o imagen exportada en los anexos del documento.

3. Organización del Proyecto

3.1. Planning de ejecución del proyecto (Marzo - Junio 2025)

Semana del 15 al 22 de marzo: Inicio del proyecto. Se realiza la planificación inicial y se definen los objetivos funcionales. Se comienza con la configuración del entorno de desarrollo y la base de datos.

Semana del 23 al 31 de marzo: Desarrollo de la arquitectura del backend y creación de los modelos iniciales de la base de datos. Se implementan los primeros endpoints REST.

Semana del 1 al 6 de abril: Finalización del backend base. Pruebas iniciales con Postman. Comienzo del desarrollo del frontend (aproximadamente el 5 o 6 de abril).

Semana del 7 al 15 de abril: Desarrollo de la interfaz principal del frontend. Se conectan las funcionalidades principales con el backend. Se define el diseño general del proyecto.

Semana del 16 al 30 de abril: Integración completa entre frontend y backend. Se implementan funcionalidades como el inicio de sesión con Google y la gestión de anuncios de vehículos.

Semana del 1 al 15 de mayo: Desarrollo de funcionalidades adicionales como el bloqueo de usuarios. Mejora en la interfaz y experiencia de usuario.

Semana del 16 al 20 de mayo: Implementación de Google Maps API para mostrar localización de los vehículos. Pruebas y validación de todo el sistema.

Semana del 20 de mayo al 1 de junio: Despliegue completo del proyecto en la VPS usando Docker. Pruebas finales en entorno de producción y redacción de documentación final.

3.2. Recursos utilizados para su desarrollo

- **Hardware:**
 - Ordenador portátil personal con Windows 11

- **Software y herramientas:**
 - Cursor (editor de código principal).
 - Postman para pruebas de las API.
 - Git y GitHub para el control de versiones y despliegue.
 - Google Chrome (modo desarrollador) para testeo en modo móvil.
 - Notion para toma de notas y planificación.
 - Google Docs para la redacción del informe.

- **Infraestructura:**
 - VPS gratuita de Google Cloud con Ubuntu 20.04.
 - Docker para contenerización de los servicios (backend, frontend, base de datos).
 - Git para transferencia del proyecto a la VPS.

4. Manual del Usuario

4.1. Preguntas Frecuentes

¿Cómo puedo registrarme en SWAPLT?

Para registrarte en la plataforma:

1. Accede a la página principal de SWAPLT.
2. Haz clic en el botón “**Registrarse**”.
3. Inicia sesión mediante tu cuenta de Google.
4. ¡Listo! Ya puedes comenzar a usar SWAPLT.

¿Cómo puedo publicar mi vehículo?

1. Inicia sesión con tu cuenta.
2. Dirígete a la sección “**Publicar Vehículo**”.
3. Rellena todos los campos requeridos: marca, modelo, año, kilometraje, etc.
4. Sube fotografías del vehículo (se recomienda que sean claras y de buena calidad).
5. Revisa todos los datos y haz clic en “**Publicar**”.

¿Cómo funciona el proceso de intercambio entre usuarios?

1. Encuentra un vehículo que te interese mediante la búsqueda.
 2. Contacta con el usuario propietario del vehículo.
 3. Propón un intercambio o una posible compraventa.
 4. Si ambas partes están de acuerdo, se coordina una cita física.
 5. Se realiza la inspección y, si todo está conforme, se formaliza el intercambio mediante la documentación legal correspondiente.
-

4.2. Guía de Uso de la Plataforma

Búsqueda de Vehículos

Puedes buscar entre cientos de vehículos utilizando filtros avanzados:

- **Marca, modelo y año.**
- Rango de **precio mínimo y máximo.**
- **Kilometraje**, tipo de cambio, combustible, etc.
- Ubicación mediante el **mapa integrado de Google Maps.**

Gestión de Favoritos

1. Al visualizar un vehículo, puedes hacer clic en el icono de “**Corazón**” para guardarlo como favorito.
 2. Los vehículos favoritos aparecen en tu perfil para fácil acceso.
 3. Puedes eliminarlos o compararlos desde tu panel de usuario.
 4. Si el propietario actualiza el precio o elimina el anuncio, recibirás una notificación (si estás conectado).
-

Otras Funcionalidades Destacadas

- **Sistema de bloqueo de usuarios:** puedes bloquear a otros usuarios para evitar que vean tus anuncios o se comuniquen contigo.
- **Responsive Design:** SWAPLT funciona perfectamente tanto en escritorio como en móviles.
- **Footer con enlaces legales:** Accede a los **Términos de uso**, **Política de privacidad** y **Centro de ayuda** desde el pie de página en cualquier momento.

5. Manual del Programador

5.1. Sistema Operativo

- **Desarrollo local:** Windows 11
- **Servidor de producción (VPS):** Ubuntu 20.04

5.2. Entorno de Desarrollo

- **Editor de código:** Cursor (basado en Visual Studio Code)
- **Herramientas auxiliares:**
 - Postman (para pruebas de API)
 - Google Chrome Developer Tools (para pruebas responsive)
 - Git y GitHub (control de versiones y despliegue)
 - Docker (contenedores de backend y frontend)
 - Notion (para documentación durante el desarrollo)
 - Google Docs (documentación final)

5.3. Tecnologías utilizadas

- **Backend:**
 - Framework: Laravel v12
 - Lenguaje: PHP 8+
 - Servidor embebido para desarrollo: `php artisan serve`
 - Autenticación: JWT
 - Base de datos: MySQL
 - ORM: Eloquent
 - Cifrado: Hash en contraseñas
 - Middleware personalizado para CORS y roles
 - API RESTful
- **Frontend:**
 - Framework: Angular v14
 - Lenguaje: TypeScript
 - Arranque en desarrollo: `ng s`
 - Consumo de API mediante `HttpClient`
 - Arquitectura por módulos (e.g. `admin/`, `components/`, `services/`, etc.)
 - Responsive design verificado con herramientas de Google Chrome
 - Integración con Google Maps API para geolocalización de vehículos

5.4. Estructura de Carpetas

Backend (Laravel):

- `/app/Http/Controllers` → Controladores de la API
- `/app/Http/Middleware` → Middleware personalizados (CORS, autenticación, roles)
- `/app/Models` → Modelos de Eloquent
- `/database/migrations` → Migraciones de la base de datos
- `/database/seeder`s y `/database/factories` → Datos de prueba
- `/routes/api.php` → Rutas de la API REST
- `.env` → Configuración del entorno (BD, API Keys, Email, etc.)

Frontend (Angular):

- `/src/app/components` → Componentes reutilizables
- `/src/app/pages` → Páginas principales
- `/src/app/services` → Servicios para conexión con la API
- `/src/app/admin` → Módulos de administración
- `/src/environments/` → Configuraciones por entorno (desarrollo, producción)

5.5. Base de Datos

- **Gestor:** MySQL
- **Modelo relacional:** Generado mediante Mermaid.js
- **Entidades principales:**
 - Usuarios (datos básicos y rol)
 - Vehículos (atributos, dueño, localización)
 - Imágenes (asociadas a vehículos)
 - Favoritos
 - Mensajes entre usuarios
 - Visitas a anuncios
 - Valoraciones
 - Reportes
 - Bloqueos de usuarios

5.6. Despliegue en Producción

- **Servidor:** VPS de Google Cloud
- **Sistema operativo:** Ubuntu 20.04
- **Recursos:** 2 vCPU, 1 GB de RAM, 30 GB de almacenamiento + 2 GB de SWAP
- **Dominio gratuito con DuckDNS**
- **Docker:** Usado para contenerizar backend y frontend
 - Docker Compose para levantar los servicios
 - Red interna para comunicar Laravel y MySQL
- **Web server / proxy inverso:** Caddy
 - Configuración para HTTPS automático y redirección a Angular y Laravel
- **Monitorización:** Uptime Kuma

