

Trabajo Práctico Integrador

CARRERA: Ingeniería en sistemas de información

MATERIA: Paradigmas y lenguajes de programación II

COMISIÓN: “A” (única)

PROFESOR: Mgter. Ing. Agustín Encina

ESTUDIANTE: Alarcon Alex

FECHA: 18-11-2025



Proyecto: Luxury Car's - Concesionaria Premium

Introducción

Este informe detalla el trayecto del proyecto web "Luxury Car's", pasando de un prototipo de **sitio web estático basado en HTML y JavaScript** a una **aplicación web dinámica y escalable** utilizando **PHP, SQL**, y una **estructura modular de backend**. Esta migración representó un avance fundamental en la capacidad del sistema para gestionar datos, interactuar con usuarios y soportar futuras funcionalidades de negocio.

1. Cambio de arquitectura fundamental

El cambio más significativo es el abandono de la presentación puramente del lado del cliente a favor de una arquitectura de tres capas, donde la mayor parte del procesamiento y la generación de contenido ocurre en el servidor.

De Prototipo Estático a Aplicación Dinámica

La versión anterior se caracterizaba por:

- **Contenido Estático:** Las páginas (**.html**) se entregaban tal cual al navegador.
- **Datos en el Cliente:** La información del catálogo de autos se almacenaba en variables JavaScript (ej. en **producto.html** y **comprar.html**).
- **Simulación:** El formulario de compra (**comprar.html**) solo simulaba el proceso, ocultando el formulario y mostrando un mensaje de éxito sin ninguna persistencia de datos reales.

La nueva arquitectura se basa en:

- **Generación Dinámica (PHP):** Todos los archivos de presentación ahora son scripts PHP (**.php**), lo que permite que el código del servidor genere el HTML en tiempo real.
- **Persistencia de Datos (SQL):** Se introduce una **Base de Datos SQL** (evidenciada por **BD.sql** y **db.php**) como el motor central de datos.
- **Gestión de Sesiones y Transacciones:** Se implementan scripts dedicados a manejar el flujo de usuarios y las operaciones de compra.

2. Modularización y Reutilización de Código (Inclusión de PHP)

Se ha implementado el principio de **No Repetir Código (DRY - Don't Repeat Yourself)** mediante la modularización de la interfaz en componentes PHP reusables. Esto simplifica el mantenimiento y asegura la consistencia de la interfaz en todo el sitio.

Componentes de Interfaz Modular

Componente PHP	Función y Beneficio



header.php	Contiene el inicio del documento HTML, la etiqueta <code><head></code> , la inclusión de estilos CSS y la apertura del <code><body></code> . Permite definir metadatos, títulos y scripts esenciales de manera centralizada para todas las páginas.
footer.php	Contiene el código HTML del pie de página, el cierre del <code><body></code> y la inclusión de scripts JavaScript transversales como <code>app.js</code> . Asegura que el pie de página sea idéntico y que todos los scripts necesarios se carguen correctamente al final de cada página.
nav.php	Contiene el código HTML específico de la barra de navegación principal y el botón <i>burger</i> (<code>nav-toggle</code>). Al ser un módulo, se puede actualizar el menú en un solo lugar y ese cambio se reflejará automáticamente en <code>index.php</code> , <code>listado_box.php</code> , <code>comprar.php</code> , etc.

Transformación de Páginas de Presentación

Los archivos HTML originales han sido renombrados y reestructurados como archivos PHP, permitiendo la inyección de los módulos de interfaz:

- `index.html` → `index.php`
- `listado_box.html` → `listado_box.php`
- `listado_tabla.html` → `listado_tabla.php`
- `comprar.html` → `comprar.php`

3. Desarrollo del Backend y Bases de Datos

El cambio más profundo está en la lógica de negocio y datos, que ahora están en el servidor.

Centralización de Datos con SQL

1. **BD.sql:** Este archivo indica la existencia de una **Base de Datos Relacional** (MySQL WorkBench) que almacena la información de los vehículos, usuarios y transacciones.
2. **db.php:** Este script se encarga de establecer la **conexión** entre la aplicación PHP y el motor de la base de datos SQL. Es el puente que permite a otros scripts PHP (como `listado_box.php` o `procesar_compra.php`) realizar consultas, inserciones y actualizaciones.

Unificación y Dinamismo del Producto

El antiguo sistema mantenía la base de datos de autos en varios archivos JavaScript. La nueva estructura lo simplifica radicalmente:

- **Archivo Único de Ficha:** Los archivos separados por auto (`producto_roma.html`, `producto_aventador.html`, etc.) han sido **eliminados**.



- **producto.php:** Reemplaza a todos los archivos anteriores. Este único script PHP recibe el identificador del vehículo (probablemente por la URL) y realiza una consulta a la Base de Datos SQL para obtener y mostrar dinámicamente: nombre, especificaciones, precio e imagen.

Flujo de Transacción Real

El proceso de compra ha sido totalmente reescrito para la gestión de datos persistente:

1. **comprar.php:** Contiene el formulario. Los datos del vehículo y el precio se cargan dinámicamente desde la base de datos, en lugar de ser codificados en JavaScript.
2. **procesar_compra.php:** Es el **punto de destino del formulario**. Este script toma los datos del usuario, realiza validaciones y ejecuta una **transacción de inserción** en la base de datos SQL para registrar la venta.
3. **compra_exitosa.php:** Reemplaza el simple mensaje JS (*success-message*). Es una página independiente que se carga *después* de que **procesar_compra.php** confirma que la transacción ha sido exitosa y persistente.

4. Gestión de Usuarios y Administración

Se incluyó componentes esenciales de gestión de usuarios, lo que abre la puerta a funcionalidades personalizadas y un control administrativo.

Autenticación y Cuentas de Usuario

- **register.php:** Permite a los nuevos usuarios crear una cuenta, almacenando sus credenciales y datos en la base de datos SQL.
- **login.php** y **logout.php:** Implementan el flujo de inicio y cierre de sesión. Esto es un requisito fundamental para futuras funcionalidades como carritos de compra persistentes, historial de pedidos y acceso a áreas restringidas.

Panel de Administración

La creación de una subcarpeta **/admin** y el archivo **listado_compras.php** confirman la adición de un módulo administrativo:

- **Acceso Restringido:** El directorio **/admin** típicamente requiere que el usuario haya iniciado sesión y posea un rol específico (Administrador).
- **listado_compras.php:** Es una herramienta interna que consulta la Base de Datos SQL para listar, filtrar, y gestionar todas las ventas registradas a través de **procesar_compra.php**.

5. Archivos Complementarios

Finalmente, se mantienen o añaden scripts de utilidad:

Archivo	Descripción



app.js	Mantiene la funcionalidad de interfaz del lado del cliente, incluyendo el manejo del menú móvil, las animaciones <i>on-scroll</i> , y la lógica de búsqueda/ordenamiento en las vistas de listado.
public-js-validati on-safe.js	Un script que sugiere la centralización y estandarización de las validaciones de formularios del lado del cliente (para mejorar la experiencia de usuario antes del envío al servidor).
error.php	Un manejador de errores genérico para mostrar mensajes amigables al usuario en caso de fallos del servidor o rutas inexistentes.



CONCLUSIONES

El proyecto de "Luxury Car's" fue un esfuerzo considerable, aunque fue cansador por todos los cambios, se concluye con una experiencia de aprendizaje positiva y valiosa.

El cambio de arquitectura basada en **PHP y SQL** no fue simplemente un cambio de archivos, sino una reingeniería completa de toda la estructura de la página. Tareas como la implementación de la autenticación (**login.php, register.php**) y el establecimiento de un flujo de transacciones real (**procesar_compra.php**) fueron unos buenos desafíos.

Sin embargo, cada hora trabajada en el proyecto se ganó el aprendizaje de nuevas habilidades.



Carrera: Ingeniería en sistemas de
información
Materia: Paradigmas y lenguajes de
programación
Estudiante: Alarcon Alex

Profesor: Mgter. Ing. Agustín Encina
Comisión: "A"

LINK

GITHUB: <https://github.com/aalexalarcon7/LuxuryCars/tree/main>



Carrera: Ingeniería en sistemas de
información
Materia: Paradigmas y lenguajes de
programación
Estudiante: Alarcon Alex

Profesor: Mgter. Ing. Agustín Encina
Comisión: "A"

ANEXOS