



INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA

ARQUITECTURA WEB PARA EL DESARROLLO DEL SISTEMA DE CONTROL VEHICULAR

TESIS

**PARA OBTENER EL TÍTULO DE
INGENIERO EN TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIONES**

P R E S E N T A

MARÍA ISABEL ALEJO AQUINO

ASESORES:

LIC. DANIELA LIZBETH HERNÁNDEZ SÁNCHEZ

**MSC. ARNULFO GAMALIEL HERNÁNDEZ
GONZÁLEZ**

CO-ASESOR:

DRA. ANA LILIA SOSA Y DURÁN

MISANTLA, VERACRUZ

JUNIO 2022



INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA
DIVISIÓN DE ESTUDIOS PROFESIONALES
AUTORIZACIÓN DE IMPRESIÓN DE TRABAJO DE TITULACIÓN

FECHA: 03 de Junio de 2022.

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS PROFESIONAL.

A QUIEN CORRESPONDA:

Por medio de la presente hago constar que el (la) C:

MARÍA ISABEL ALEJO AQUINO

pasante de la carrera de INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES con No. de Control 172T0634 ha cumplido satisfactoriamente con lo estipulado por el **Manual de Procedimientos para la Obtención del Título Profesional de Licenciatura** bajo la opción **Titulación Integral (Tesis Profesional)**

Por tal motivo se **Autoriza** la impresión del **Tema** titulado:

“ARQUITECTURA WEB PARA EL DESARROLLO DEL SISTEMA DE CONTROL VEHICULAR”

Dándose un plazo no mayor de un mes de la expedición de la presente a la solicitud del Acto de Recepción para la obtención del Título Profesional

ATENTAMENTE

M.I.I. GRACIELA GUADALUPE AGUILERA ÁLVAREZ
DIVISIÓN DE ESTUDIOS PROFESIONALES



Archivo.

AGRADECIMIENTOS

Agradezco de corazón a mis padres por su apoyo, esfuerzo y comprensión durante mi educación, quienes estuvieron en todo momento motivándome a seguir adelante y nunca rendirme, gracias a mi madre que me acompañó en momentos de desvelo y se preocupó por mi bienestar, a mi padre por siempre confiar en mí y darme buenos consejos. Cada uno de mis logros se los debo a ellos.

A mis hermanos que desde pequeña me enseñaron a luchar por mis metas, por estar a mi lado en todo momento y a guiarme con sus consejos y vivencias.

A mis compañeros, con quienes viví buenas y malas experiencias. Me aportaron sus aprendizajes siempre que lo requerí.

A mis maestros, que me inculcaron las bases del conocimiento, las cuales me van a servir para afrontar la vida laboral.

Agradezco infinitamente a una persona especial para mí, Luis Enrique, que siempre me ha apoyado y motivado a seguir adelante y lograr todos mis propósitos.

A mis asesores, la Lic. Daniela Lizbeth Hernández Sánchez, el MSC. Arnulfo Gamaliel Hernández González y la Dra. Ana Lilia Sosa y Durán por brindarme su apoyo para concluir mi proyecto.

RESUMEN

El presente proyecto tiene como objetivo desarrollar un sistema web para capturar datos representativos de todos los automóviles del padrón vehicular de la Secretaría de Infraestructura y Obras Públicas del Estado de Veracruz, que permita administrarlos correctamente, proporcionando mejoras y otorgando a los usuarios una mejor experiencia de navegación. El sistema permite al personal del Departamento de Recursos Materiales y Servicios Generales llevar un registro de los resguardos al momento en que se realiza una comisión, guardar las cargas de combustible y llevar el control de los mantenimientos a los vehículos.

El Departamento de Recursos Materiales y Servicios Generales se enfrentaba con diferentes problemáticas al momento de utilizar el Sistema de Control Vehicular anterior. Entre ellas estaba, el difícil acceso al programa debido a que no todos los ordenadores podían instalar el software. Por otro lado, había funciones que ya no podían realizarse en dicho sistema o que deseaban poder hacer con el fin de ahorrar tiempo, como, por ejemplo, editar formatos con información de los vehículos, para posteriormente imprimir.

La metodología de ingeniería de software utilizada se basó en el modelo de cascada, lo cual permitió tener una visión clara los requerimientos del sistema, para así poder dividir las actividades que conllevaron a la meta.

El uso de las tecnologías actuales permitió un desarrollo ágil de la aplicación web para finalmente solucionar la problemática en la que se encontraba la Oficina de Control Vehicular, con ello se podrán generar los documentos que se editaban manualmente, evitando así algún error con los datos.

La implementación de librerías como DomPDF, Bootstrap y DataTables, disminuyó la cantidad de líneas de código desarrolladas.

Índice

Capítulo 1 Generalidades del proyecto	1
1.1 Introducción.....	1
1.2 Descripción de la empresa.....	2
1.3 Planteamiento del problema.....	4
1.4 Objetivos	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos.....	5
1.5 Justificación.....	6
1.6 Hipótesis	7
1.7 Propuesta de solución.....	8
1.7.1 Patrón de diseño: Modelo Vista Controlador	9
1.8 Alcances y limitaciones	11
1.8.1 Alcances	11
1.8.2 Limitaciones.....	11
Capítulo 2 Marco teórico	12
2.1 Modelos de ingeniería de software	12
2.1.1 Modelo en cascada.....	12
2.2 Sistema web.....	13
2.2.1 Aplicación web dinámica	13
2.3 Editor de código	14
2.3.1 Visual Studio Code	14
2.4 Lenguajes de programación.....	15
2.4.1 PHP 7.4.22	15
2.4.2 JavaScript.....	15
2.5 Patrón de diseño: MVC (Modelo Vista Controlador)	15
2.6 Framework para desarrollo web.....	16
2.6.1 Framework para backend: Laravel 8.0	17
2.6.2 Framework para frontend: Bootstrap 4.6	17
2.7 Plantilla de panel de administración	17
2.7.1 AdminLte3	18
2.8 Sistema Gestor de Base de Datos (SGBD).....	18

2.9	Generador de reportes PDF	19
2.9.1	Librería DomPdf.....	19
2.10	Estado del arte	20
2.10.1	Sistemas de Control Vehicular.....	20
2.10.2	Arquitecturas Modelo Vista Controlador	21
2.10.3	Software existente en el mercado para control vehicular	21
Capítulo 3 Desarrollo.....		23
3.1	Requisitos	23
3.1.1	Casos de uso.....	23
3.1.2	Cronograma de actividades.....	27
3.1.3	Requerimientos.....	28
3.2	Diseño	30
3.2.1	Modelado de datos	30
3.2.2	Desarrollo del sistema	32
3.2.2.1	Creación del proyecto.....	32
3.2.2.2	Instalación de Plantilla AdminLte3.....	32
3.2.2.3	Creación de la base de datos.....	33
3.2.2.4	Creación de Seeders.....	33
3.2.2.5	Creación de modelos.....	33
3.2.2.6	Creación de controladores.	33
3.2.2.7	Creación de vistas.....	33
3.2.2.8	Creación de rutas.	34
3.2.2.9	Programación de los módulos.	34
3.2.2.10	Creación de formatos.....	35
3.3	Implementación	35
3.4	Verificación.....	36
3.5	Mantenimiento.....	36
Capítulo 4 Resultados		37
4.1	Patrón de diseño: MVC	37
4.1.1	Nivel modelo	37
4.1.1.1	Migraciones.....	37
4.1.1.1.1	Módulo departamento.	37

4.1.1.1.2 Módulo personal.....	38
4.1.1.1.3 Módulo proveedores.	38
4.1.1.1.4 Módulo usuarios.....	38
4.1.1.1.5 Módulo padrón vehicular.....	39
4.1.1.1.6 Módulo resguardos.	39
4.1.1.1.7 Módulo combustible.	40
4.1.1.1.8 Módulo mantenimiento.....	40
4.1.1.2 Modelos.....	41
4.1.2 Nivel vista	42
4.1.2.2 Panel principal.....	42
4.1.2.3 Departamentos.....	43
4.1.2.4 Personal.	45
4.1.2.5 Proveedores.....	46
4.1.2.6 Usuarios.	48
4.1.2.7 Padrón vehicular.	49
4.1.2.8 Resguardos.....	51
4.1.2.9 Control de combustible.....	54
4.1.2.10 Control de serv. de mantenimiento.	56
4.1.2.11 Licencias.....	59
4.1.2.12 Pólizas de seguro.....	60
4.1.2.13 Facturas.....	60
4.1.3 Nivel controlador.....	61
Capítulo 5 Conclusiones	66
5.1 Conclusiones y recomendaciones.....	66
5.1.1 Conclusión.....	66
5.1.2 Recomendaciones.....	66
Capítulo 6 Referencias bibliográficas	67
6.1 Referencias bibliográficas	67
Capítulo 7 Anexos	69
7.1 Anexos	69

Índice de figuras

<i>Figura 1.1</i> Ubicación geográfica de la Secretaría de Infraestructura y Obras Públicas.....	2
<i>Figura 1.2</i> Estructura orgánica en la SIOP.....	3
<i>Figura 1.3</i> Arquitectura física del sistema de web de Control Vehicular.....	8
<i>Figura 2.1</i> Modelo en cascada... ..	13
<i>Figura 2.2</i> Funcionamiento de una aplicación web dinámica... ..	14
<i>Figura 2.3</i> Modelo MVC.	16
<i>Figura 2.4</i> Plantilla AdminLTE.....	18
<i>Figura 3.1</i> Sistema general... ..	23
<i>Figura 3.2</i> Agregar un registro.....	23
<i>Figura 3.3</i> Modificar un registro.....	24
<i>Figura 3.4</i> Borrar un registro.	25
<i>Figura 3.5</i> Modelo de datos entidad-relación.... ..	31
<i>Figura 3.6</i> Diagrama del modelo relacional.....	32
<i>Figura 3.7</i> Cliente SSH PuTTY.....	35
<i>Figura 4.1</i> Migración de la tabla Departamento.....	37
<i>Figura 4.2</i> Migración de la tabla Personal... ..	38
<i>Figura 4.3</i> Migración de la tabla Proveedor.....	38
<i>Figura 4.4</i> Migración de la tabla Usuario.....	39
<i>Figura 4.5</i> Migración de la tabla Padrón Vehicular.....	39
<i>Figura 4.6</i> Migración de la tabla Resguardo.....	40
<i>Figura 4.7</i> Migración de la tabla Combustible... ..	40
<i>Figura 4.8</i> Migración de la tabla Mantenimiento.....	41
<i>Figura 4.9</i> Modelo Proveedor... ..	41
<i>Figura 4.10</i> Formulario para iniciar sesión.. ..	42
<i>Figura 4.11</i> Tablero principal.....	43
<i>Figura 4.12</i> Vista principal de los departamentos.....	43
<i>Figura 4.13</i> Formulario para agregar un departamento.....	44
<i>Figura 4.14</i> Formulario editar departamento.....	44
<i>Figura 4.15</i> Página principal de Personal.....	45
<i>Figura 4.16</i> Formulario agregar personal.....	45
<i>Figura 4.17</i> Formulario para editar personal.....	46
<i>Figura 4.18</i> Modal de detalles de personal.....	46
<i>Figura 4.19</i> Pantalla principal proveedores.....	47
<i>Figura 4.20</i> Formulario agregar un proveedor.....	47
<i>Figura 4.21</i> Formulario modificar al personal.....	48
<i>Figura 4.22</i> Vista principal de los usuarios registrados.....	48
<i>Figura 4.23</i> Formulario de registro de usuarios.....	49
<i>Figura 4.24</i> Vista principal de los vehículos registrados.....	50
<i>Figura 4.25</i> Formulario para añadir un vehículo.....	50
<i>Figura 4.26</i> Formulario para modificar un vehículo.....	51

<i>Figura 4.27</i> Modal de detalles de un vehículo.....	51
<i>Figura 4.28</i> Vista de los resguardos registrados.....	52
<i>Figura 4.29</i> Formulario para añadir un resguardo.....	53
<i>Figura 4.30</i> Formulario para editar un resguardo.....	53
<i>Figura 4.31</i> Modal de detalles de un resguardo.....	54
<i>Figura 4.32</i> Vista de las cargas de combustible registradas.....	55
<i>Figura 4.33</i> Formulario para agregar una compra de combustible.....	55
<i>Figura 4.34</i> Formulario para editar un registro.....	56
<i>Figura 4.35</i> Modal de detalles de una carga de combustible.....	56
<i>Figura 4.36</i> Vista principal de los mantenimientos realizados.....	57
<i>Figura 4.37</i> Formulario para agregar un servicio.....	57
<i>Figura 4.38</i> Formulario para editar un mantenimiento.....	58
<i>Figura 4.39</i> Modal de detalles de un servicio realizado.....	58
<i>Figura 4.40</i> Lista de los mantenimientos pendientes.....	59
<i>Figura 4.41</i> Listado de las licencias vencidas.....	59
<i>Figura 4.42</i> Vista de las pólizas expiradas.....	60
<i>Figura 4.43</i> Listado de las facturas de los mantenimientos y cargas de combustible.....	61
<i>Figura 4.44</i> Método index del controlador Departamento... ..	61
<i>Figura 4.45</i> Método create y store del controlador Departamento... ..	62
<i>Figura 4.46</i> Método edit y update del controlador Departamento.....	63
<i>Figura 4.47</i> Método destroy del controlador Departamento... ..	63
<i>Figura 4.48</i> Método index del controlador Combustible... ..	64
<i>Figura 4.49</i> Método fechasVencimiento del controlador Personal... ..	64
<i>Figura 4.50</i> Método vencimientoPoliza... ..	64
<i>Figura 4.51</i> Métodos para generar formatos PDF del controlador Resguardo.....	65

Índice

<i>Tabla 3. 1.</i> Cronograma de actividades.....	27
<i>Tabla 3. 2.</i> Requerimientos del sistema	28

Capítulo 1 Generalidades del proyecto

1.1 Introducción

Los sistemas informáticos hoy en día se han vuelto un pilar fundamental en las empresas, para el registro de sus operaciones, existe software que se basa en registrar la compra y venta de productos, gestionar a los empleados, controlar el almacén de productos, entre otros.

Las empresas y organizaciones que cuentan con un gran número de bienes tienen a su disposición programas que los ayudan a tener conocimiento de sus bienes activos. Estos sistemas les proporcionan una gran cantidad de ventajas.

En este trabajo se presenta el desarrollo de un sistema web para el Departamento de Recursos Materiales y Servicios Generales de la Secretaría de Infraestructura y Obras Públicas, con el fin de proporcionar mejoras y nuevas funciones al momento de llevar a cabo sus actividades diarias, como lo es asignar un vehículo a un chofer al momento de una comisión y llevar el control de las cargas de combustible. El uso de este sitio web facilitará la interacción entre los usuarios y el sistema, agilizando los procesos y otorgando los beneficios de las tecnologías utilizadas. El sistema implementa una arquitectura de 3 niveles utilizando el patrón de diseño Modelo Vista Controlador, con el fin de facilitar el desarrollo de la aplicación separando la información, la lógica del sistema y la interfaz del usuario.

El desarrollo del sistema de Control Vehicular se llevó a cabo con el framework Laravel 8, tomando en cuenta la metodología en cascada. Inicialmente se realizó un análisis de las actividades que se toman en cuenta al momento de dar de alta un automóvil, resguardar un vehículo, suministrar combustible y llevar a mantenimiento.

El capítulo 1 describe el tema de la tesis, situando el lugar donde se implementa el sistema de Control Vehicular, mostrando sus diferentes apartados, como lo son: introducción, descripción de la empresa, planteamiento del problema, objetivos, justificación, hipótesis, propuesta de solución, finalmente los alcances y limitaciones.

Los trámites o servicios que ofrece la Secretaría a la ciudadanía son:

- Revisión y validación de proyectos para el desarrollo de obras de infraestructura de caminos y carreteras estatales.
- Permisos para realizar trabajos dentro del derecho de vía de los caminos y carreteras estatales.
- Participación en procesos de licitaciones de obras públicas y servicios relacionados.

La dependencia cuenta con diferentes departamentos (figura 1.2) enfocados a las actividades que se llevan a cabo en la dependencia, en el área de Tecnologías de la Información se atienden asuntos relacionados con el soporte técnico de los equipos, el mantenimiento y/o desarrollo de los sistemas informáticos.

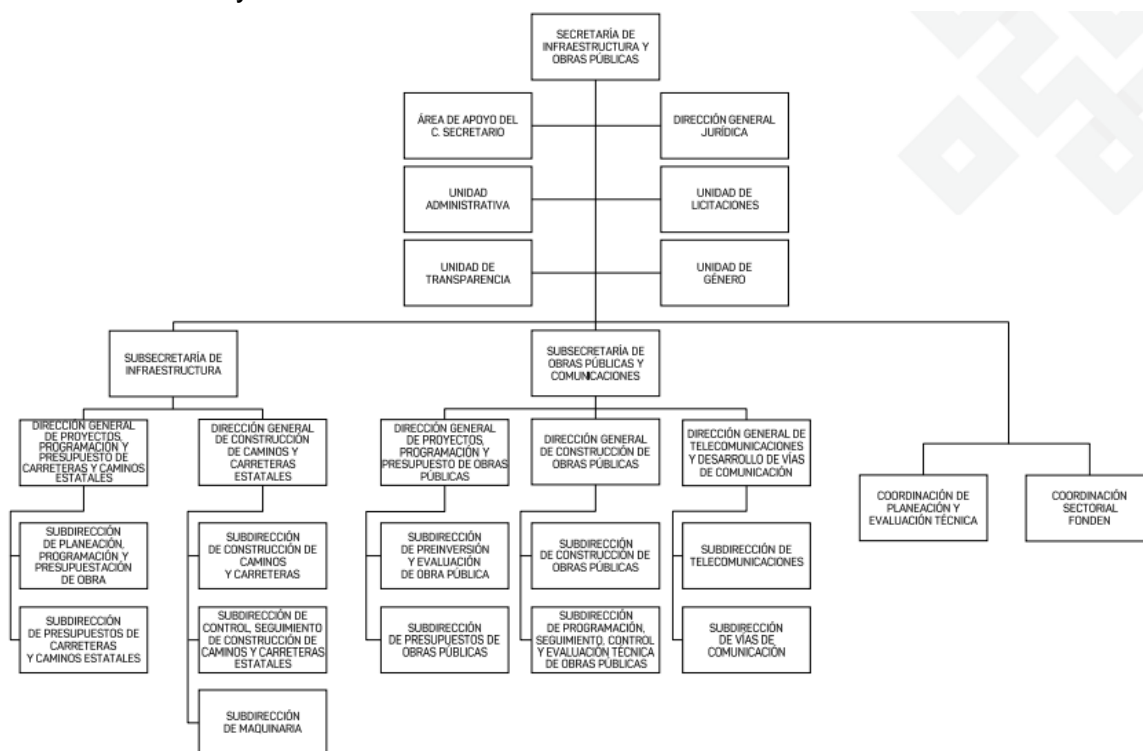


Figura 1.2. Estructura orgánica en la SIOP. (Gobierno del estado, s.f.).

El departamento de Tecnologías de la Información está conformado por dos oficinas: Oficina de Gobierno Electrónico y Desarrollo de Aplicaciones y la Oficina de Infraestructura y Soporte Técnico. En la primer oficina, se realizan las actualizaciones

y desarrollo de los sistemas empleados por la secretaría, en el segundo se proporciona el mantenimiento a equipos de cómputo, redes de internet, servidores e impresoras.

1.3 Planteamiento del problema

El primer problema presentado con el sistema de Control Vehicular que anteriormente estaba en uso en la Secretaría de Infraestructura y Obras Públicas era la incompatibilidad del programa con los ordenadores, ya que solo podía usarse en computadoras con características específicas, otra problemática presentada era la imposibilidad de actualizar el sistema debido a que el desarrollador del sistema no se encontraba en la dependencia y al ser codificada en un lenguaje de programación poco usado actualmente, el personal del Departamento de Tecnologías de la Información no tenía conocimientos de como modificarlo.

Cada área de la Secretaría tiene diversas funciones, para ello, ocupan software que en cierta medida optimiza el tiempo de trabajo, algunos programas normalmente usados son Excel, Word y PowerPoint. Por otro lado, la mayor parte de los departamentos cuentan con sistemas desarrollados específicamente para los procesos que efectúan.

Para acceder a los sistemas que manejan en cada departamento, es necesario ir a la plataforma digital Intranet.Siop. Los sistemas deben descargarse para después ejecutarse en el ordenador, lo cual resulta difícil para el personal, así mismo los programas no permiten realizar todas las funciones que se necesitan, por ejemplo, generar los formatos que son elaborados en editores de texto.

Algunos de los sistemas están desarrollados con el lenguaje de programación Delphi, del cual su última versión estable se lanzó en el año 2020, sin embargo, actualmente no es uno de los programas que sean más utilizados. Debido a esto el soporte o actualización en ellos es casi imposible de llevar a cabo.

En el momento que los sistemas dejan de recibir actualizaciones por parte de los desarrolladores son probables a sufrir algunas de las siguientes situaciones:

- Pérdidas en las bases de datos.

- Problemas de seguridad. La protección informática de los datos que se guardan en sistemas de gran importancia es vital tomar en cuenta. Los sistemas que no proporcionan actualizaciones o soporte después de crearlos, tienen problemas cuando se presenta algún incidente que no detectaron a tiempo.
- Baja el rendimiento. Las aplicaciones web a lo largo de su ciclo de vida, requiere de mejoras que son de beneficio para los usuarios de las aplicaciones.

El Departamento de Recursos Materiales y Servicios Generales, requiere del sistema de Control Vehicular para administrar los transportes que son propiedad de la Secretaría, para ello necesita adjuntar imágenes de las unidades, subir los archivos de las facturas de la carga de combustible y mantenimientos e imprimir formatos de los resguardos, que con el sistema actual deben hacer con editores de PDF o con Word. Esto implicaría que los oficios puedan tener errores al añadir los datos de los vehículos.

Es por ello que se propone desarrollar e implementar una arquitectura utilizando el patrón de diseño Modelo Vista Controlador para crear un sistema de Control Vehicular que permita solucionar la problemática presentada en la Secretaría.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar e implementar una arquitectura de tres niveles utilizando el patrón de diseño Modelo Vista Controlador para el desarrollo de la aplicación web “Control Vehicular” en la Secretaría de Infraestructura y Obras Públicas del Estado de Veracruz con el fin de llevar de forma adecuada el resguardo vehicular para las comisiones del personal, utilizando el framework Laravel 8.0.

1.4.2 Objetivos específicos

- Crear diagramas de uso tomando en cuenta las actividades que se llevan a cabo en la oficina de Control Vehicular con el objetivo de conocer el comportamiento del sistema.

- Diseñar la base de datos con el fin de tener una mejor perspectiva de las tablas a crear utilizando el modelo entidad relación.
- Desarrollar la aplicación web utilizando el framework Laravel 8, con el fin de aprovechar los beneficios de las tecnologías que proporciona su implementación.
- Diseñar y crear los diferentes módulos en el proyecto, tomando en cuenta los principios de usabilidad, que permitan una interacción sencilla y fácil entre el sistema y el usuario.
- Investigar las librerías utilizadas para crear documentos .pdf, con el fin de seleccionar la que más se adapte al sistema de Control Vehicular.
- Generar archivos PDF tomando como referencia los formatos proporcionados por la Secretaría, teniendo como propósito reducir el tiempo asignado a la elaboración de reportes.
- Implementar la instalación del sistema web en el servidor de forma remota mediante VNC Viewer para la interacción de los usuarios.
- Realizar las pruebas necesarias en el Sistema de Control Vehicular, con la finalidad de mitigar posibles errores y vulnerabilidades.

1.5 Justificación

Las aplicaciones web proporcionan una serie de beneficios con respecto al software de escritorio, con lo cual se logra aprovechar y ajustar los recursos de una empresa de una forma mucho más práctica que el software habitual, así mismo las aplicaciones web tienen compatibilidad de uso, debido a que son accesibles desde un navegador con internet.

Uno de los lenguajes más usados en el mundo de la programación web, es PHP. Laravel es una plataforma para crear aplicaciones en este lenguaje. Es fácil de utilizar y posee una extensa documentación.

Mediante tecnologías de comunicación e información los sistemas web desarrollados a través del framework Laravel, permiten crear y disponer de forma inmediata los índices de productividad a las dependencias

A diferencia de otros frameworks, Laravel cuenta con una extensa documentación y una amplia comunidad en foros. Las ventajas de este marco de trabajo son:

- Su motor de plantillas Blade, permite reutilizar sus propias variables.
- Al usar la arquitectura Modelo Vista Controlador, separa todas las partes del proyecto, lo cual permite que el desarrollo del sistema sea fácil.
- Con las migraciones es posible realizar cambios en la base de datos sin necesidad de eliminar una tabla, evitando así, perder información.
- Su línea de comandos “Artisan” ofrece al framework facilidad para crear controladores, modelos, migraciones, programar tareas y realizar pruebas.

A diferencia de otros frameworks Laravel cuenta con un mapeador relacional de objetos que hace que la interacción entre las bases de datos sea agradable y permita crear consultas más complejas.

Con la creación del sistema de Control Vehicular se pretende dar solución a los problemas presentados al momento de resguardar un automóvil, adicionar mejoras que permitan ahorrar tiempo y proporcionar un mejor servicio a la ciudadanía.

Normalmente la instalación de un programa de escritorio depende de la arquitectura del CPU (*Central Processing Unit*). En la oficina de Control Vehicular, uno de los problemas presentados es que no todos sus ordenadores soportan los sistemas al momento de ser implantados, es por ello que se pretende diseñar un sistema web que permita el acceso desde cualquier computadora sin importar sus características.

1.6 Hipótesis

Es posible crear una arquitectura de 3 niveles utilizando el patrón de diseño Modelo Vista Controlador con el fin de desarrollar la aplicación web “Control Vehicular”

para la Secretaría de Infraestructura y Obras Públicas del Estado de Veracruz teniendo como propósito llevar de forma adecuada el resguardo vehicular de las comisiones en los diferentes municipios del estado.

1.7 Propuesta de solución

Hoy en día con el desarrollo de nuevas tecnologías, a las empresas se les ha permitido llevar un mejor control de sus bienes, como es el caso de la Secretaría de Infraestructura y Obras Públicas, que posee vehículos para las comisiones realizadas en el estado de Veracruz.

En este proyecto se tiene como objetivo desarrollar un sistema para administrar los vehículos que posee la Secretaría, proporcionando ventajas al personal de la Oficina de Control Vehicular. Para crear el software se diseñarán distintos formularios donde se podrá capturar, modificar y visualizar los módulos: padrón vehicular, resguardos, control de mantenimiento, carga de combustible, departamentos, personal y usuarios del sistema. En la figura 1.3 se muestra la arquitectura física del Sistema de Control Vehicular.

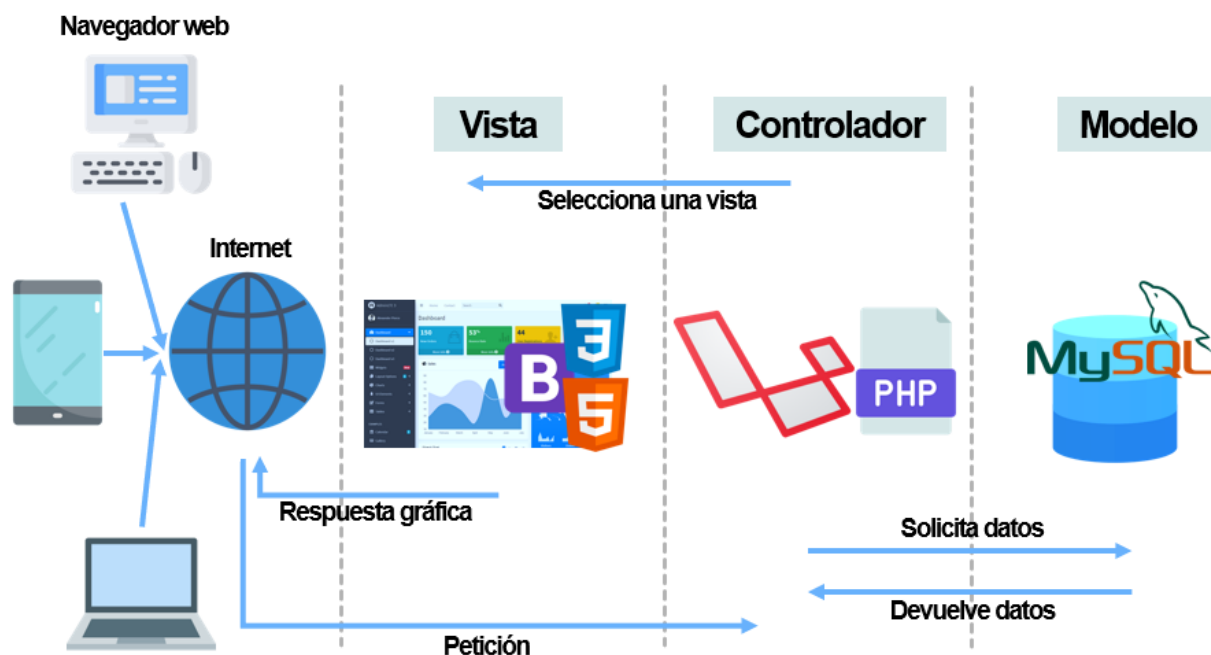


Figura 1.3. Arquitectura física del sistema web de Control Vehicular. Autoría propia.

1.7.1 Patrón de diseño: Modelo Vista Controlador

La arquitectura de la propuesta está conformada por 3 elementos: Modelo, vista y controlador. Cada nivel permite separar la información, la lógica del sistema y la interfaz del usuario.

Nivel vista. Se emplearán las plantillas blade del framework de PHP Laravel 8.0, implementando dentro el panel de administración AdminLTE3 que instala automáticamente Bootstrap 4.6. Los usos de las vistas son los siguientes:

- Muestra una ventana seleccionada por el controlador.
- Responderá al usuario mediante una interfaz gráfica y mostrará información.
- Diseño de formularios para agregar y editar los módulos: departamentos, usuarios, personal, padrón vehicular, control de combustible, resguardos y mantenimientos.
- Diseño de ventanas para visualizar los elementos almacenados en la base de datos.

Nivel controlador. Se hará uso de los controladores del framework Laravel 8, para realizar las siguientes actividades:

- Orquestrará el funcionamiento del sistema recibiendo datos desde la vista.
- Solicitará a los modelos información de la base de datos.
- Recibirá los datos de los modelos.
- Seleccionará la vista que se mostrará al usuario del sistema.
- Controlador para departamentos. Se creará un método para agregar un departamento al que pertenecerá un empleado, actualizar un campo del elemento seleccionado por el usuario, eliminar y mostrar información de la tabla.

- Controlador para personal. Se codificará una función para agregar al personal que lleva a comisión un vehículo, actualizar un dato de un chofer determinado, eliminar y mostrar información de la tabla.
- Controlador para proveedor. Se creará un método para agregar un negocio o empresa que realizará mantenimientos a vehículos que lo requieran, modificar campos de un proveedor, eliminar y mostrar información de la tabla.
- Controlador para usuario. Se implementará una función para agregar un usuario que tendrá acceso al sistema, recuperar su contraseña en caso de olvidarla, eliminar y mostrar un elemento de la tabla.
- Controlador para padrón vehicular. Se codificará un método para capturar un vehículo que sea propiedad de la Secretaría de Infraestructura y Obras Públicas, actualizar datos de un carro, eliminar y mostrar unidades de la tabla, listar los vehículos cuyas pólizas de seguro estén vencidas.
- Controlador para resguardo. Se creará una función para añadir una comisión, donde se seleccionará el personal que tendrá en resguardo un vehículo, modificar, eliminar y mostrar información de la tabla de resguardos. Al momento de agregar un nuevo elemento se codificará un método que actualice el vehículo que va a comisión con los datos del chofer. Del mismo modo dentro de este controlador se generarán los documentos .pdf, como lo son la carta responsiva y el formato de resguardo
- Controlador para combustible. Se creará un método para capturar una carga de combustible proveniente de un resguardo, editar, eliminar y mostrar los elementos de la tabla. Por otro lado, se establecerá una función que listará las facturas recibidas por el personal que tiene el vehículo en resguardo, se actualizará el kilometraje y los litros de un carro.
- Controlador para mantenimiento. Se creará una función para agregar un servicio automotriz, modificar, eliminar y mostrar elementos de la tabla, listar facturas proporcionadas por los proveedores y actualizar la fecha del último mantenimiento realizado a un vehículo.

Nivel modelo. Laravel cuenta con un mapeador relacional de objetos (ORM), con el cual para cada tabla de la base de datos se creará un *model*, mismo que estará vinculado con MySQL. Haciendo uso de ORM, dentro de los modelos se establecerán métodos para definir las relaciones existentes entre las tablas. El *model* recibirá la petición del controlador y en base a ello retornará la información solicitada proveniente del Sistema Gestor de Base de Datos.

1.8 Alcances y limitaciones

1.8.1 Alcances

El sistema de Control Vehicular permite a los usuarios administrar los vehículos de la Secretaría y coordinar la asignación de las unidades a los choferes que asistan a una comisión.

El sistema web permite dar de alta, modificar, mostrar y eliminar la información de los siguientes módulos: departamento, personal, proveedores, padrón vehicular, resguardos, control de combustible y mantenimientos. Una de las funciones importantes del sistema es su capacidad de mostrar las fechas de vencimiento de licencias, pólizas de seguro y próximos servicios vehiculares.

Los usuarios del sistema son empleados adscritos a la oficina de Control Vehicular, quienes deben tener las credenciales requeridas para iniciar sesión.

Los documentos .pdf descargables, extraen los datos capturados de los vehículos y resguardos.

1.8.2 Limitaciones

El presente sistema web tiene como limitación que el formato de orden de salida no podrá ser generado por el programa, debido a que requiere de la firma de manera electrónica del Titular del Departamento de Recursos Materiales y Servicios Generales y que por motivos de seguridad no puede ser concedida.

Capítulo 2 Marco teórico

En este capítulo se hablará sobre las tecnologías utilizadas para desarrollar el Sistema de Control Vehicular que permite al personal del Departamento de Recursos Materiales y Servicios Generales llevar a cabo sus actividades con mayor facilidad.

2.1 Modelos de ingeniería de software

La metodología de ingeniería de software que se empleó para el desarrollo del sistema web fue el modelo en cascada, debido a que su estructura tiene una serie de pasos claros y definidos que permiten enfocarse en una sola sin mezclar las fases, por lo que, si existe un problema dentro de una etapa, puede ser identificada y solucionada de inmediato, por otro lado, el objetivo final se tiene como meta inicial, lo que permite enfocarse únicamente en él, evitando perderse en cada detalle.

2.1.1 Modelo en cascada

El modelo en cascada (figura 2.1) se basa en que los equipos sigan una secuencia de pasos y nunca avancen hasta que se haya completado la fase anterior. Esta estructura es apta para proyectos más pequeños con entregables que son fáciles de definir desde el inicio (Lucidchart, s.f.).

Fases del modelo

1. Requisitos: Es esta etapa se hace un análisis de las necesidades del usuario del sistema, para establecer las funciones que debe realizar.
2. Diseño: Durante esta fase, se debe definir la organización de la estructura y de todos los elementos necesarios para el desarrollo del software.
3. Implementación: En esta etapa se implementa el código con ayuda de los lenguajes de programación.
4. Verificación: Se comprueba que el sistema funcione perfectamente, en caso de que exista un error, se corrige antes en hacer la entrega.

5. Mantenimiento: Debido a que al momento de implementar el sistema pueden surgir errores o no agrada al cliente final se realizan algunos cambios.

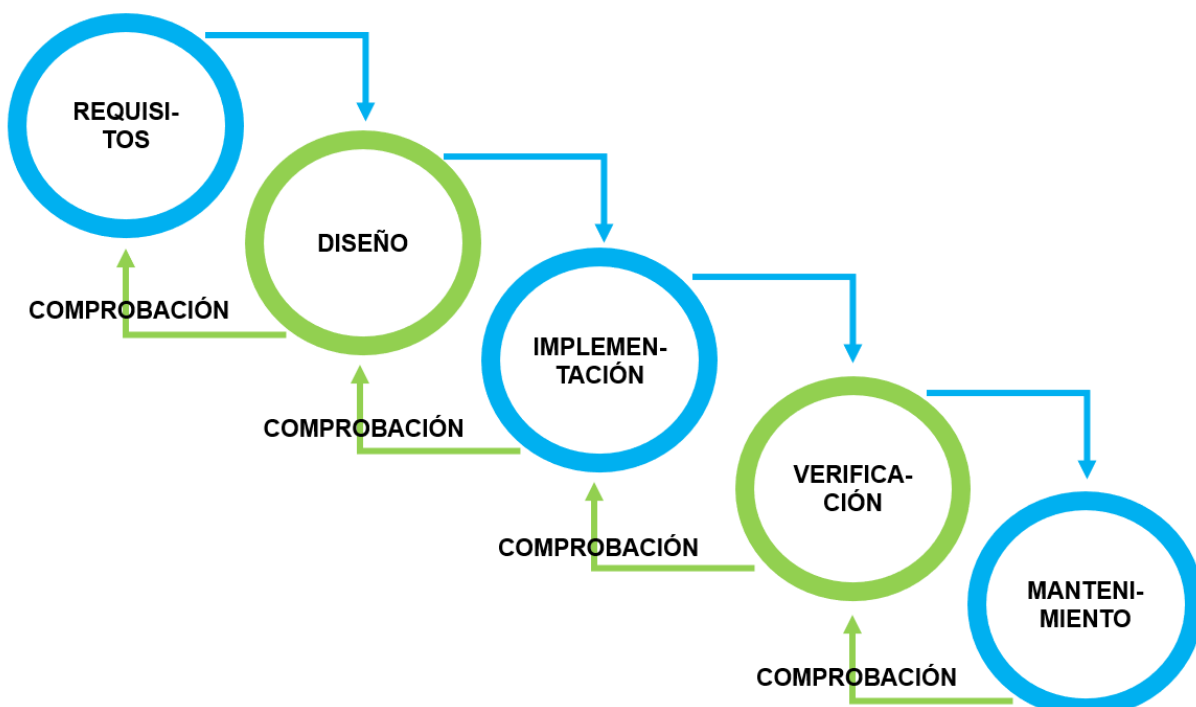


Figura 2.1. Modelo en cascada. (Risso, 2022).

2.2 Sistema web

Los sistemas Web o también conocido como aplicaciones Web son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos (Windows, Linux), sino que se alojan en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los 'sistemas Web' tienen funcionalidades muy potentes que brindan respuestas a casos particulares (Addappto, 2015).

2.2.1 Aplicación web dinámica

Las aplicaciones web dinámicas son una herramienta que se vale de una base de datos para cargar la información, y el contenido varía cada vez que un usuario accede a ellas. Por lo tanto, estas aplicaciones están en constante cambio y pueden ofrecer diferente información en función de la interacción de los usuarios, por lo cual

ofrecen una experiencia muy positiva y son ampliamente utilizadas en blogs corporativos y páginas web (Anexia Tecnologías, 2020).

El funcionamiento de estas páginas (figura 2.2) es distinto a las estáticas, cuando el servidor recibe una petición para mostrar el sitio dinámico manda la página a un software especial encargado de finalizar la página, llamado servidor de aplicaciones.

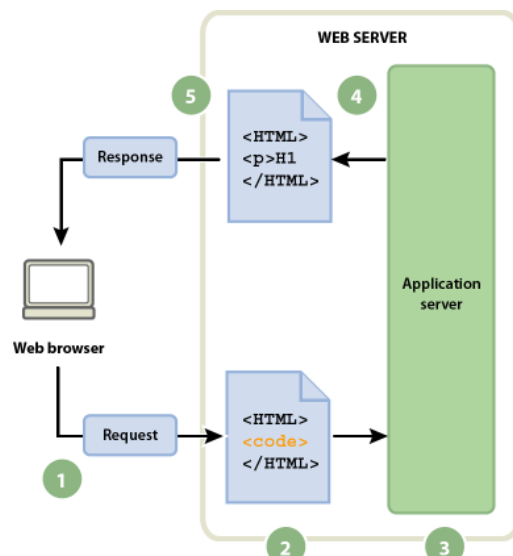


Figura 2.2. Funcionamiento de una aplicación web dinámica. (Adobe, 2021).

2.3 Editor de código

El editor que se utilizará para el desarrollo del Sistema de Control Vehicular es Visual Studio Code, el cual aporta diversas ventajas a diferencia de otros codificadores. Una de las razones por la que fue elegido es que ahorra tiempo al momento de escribir líneas de código, con ayuda de *Snippets* permite autocompletar expresiones con solo escribir una palabra; además, admite abrir varias terminales en el mismo espacio de trabajo, lo que da mucha flexibilidad al proyecto.

2.3.1 Visual Studio Code

Visual Studio Code es uno de los editores de código gratuitos más conocidos en todo el mundo. Su sencillez de uso y su interfaz gráfica son algunos de sus muchos puntos fuertes. Con su instalación se incorporan los lenguajes de JavaScript (junto con

TypeScript y Node.js), HTML y CSS, sin embargo, es posible utilizar los lenguajes Java, C++, PHP, Python, entre otros, instalando las extensiones correspondientes (Rodríguez, 2021).

2.4 Lenguajes de programación

Los lenguajes de programación en los que se codifica el proyecto es PHP en su versión 7.4.22 y JavaScript.

2.4.1 PHP 7.4.22

El PHP es un lenguaje de scripting de código abierto, del lado del servidor, con programación HTML integrada que se utiliza para crear páginas web dinámicas. Las ventajas de PHP son su flexibilidad y su alta compatibilidad con otras bases de datos. Además, PHP es considerado como un lenguaje fácil de aprender (Ryte Wiki, s.f.).

2.4.2 JavaScript

JavaScript es un popular lenguaje de programación de scripts para añadir funcionalidades interactivas y otros contenidos web dinámicos a las páginas web. Algunos ejemplos conocidos de contenido en JavaScript son los formularios rellenables, las presentaciones de galerías de fotos y los gráficos animados (Kinsta, 2022).

2.5 Patrón de diseño: MVC (Modelo Vista Controlador)

El proyecto está basado en el patrón de diseño MVC, con el propósito de separar la lógica de las vistas, facilitando así el desarrollo del sistema.

El patrón MVC (figura 2.3) es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de frameworks basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores (Díaz & Fernández, 2012).

Este modelo está estructurado por 3 componentes: Modelo, vista y controlador.

- El modelo encapsula el comportamiento y los datos correspondientes al dominio de la aplicación. Habitualmente se construye un modelo de clases del problema con el que se esté trabajando independientemente de cómo se vayan a presentar los datos de cara al usuario.
- Las vistas consultan el estado del modelo para mostrárselo al usuario. Por ejemplo, un mismo conjunto de datos puede verse en forma de tabla o gráficamente, en una pantalla o en un informe impreso. Cada una de las formas de mostrar los constituye una vista independiente a los datos, cada una de las vistas delega en el modelo, que es el responsable de obtener los datos y realizar los cálculos necesarios.
- Los controladores, por último, son los encargados de permitir que el usuario realice acciones. Dichas acciones se traducirán en las respuestas que resulten apropiadas, las cuales pueden involucrar simplemente a las vistas o incluir la realización de operaciones sobre el modelo (Berzal, et al., s.f.).

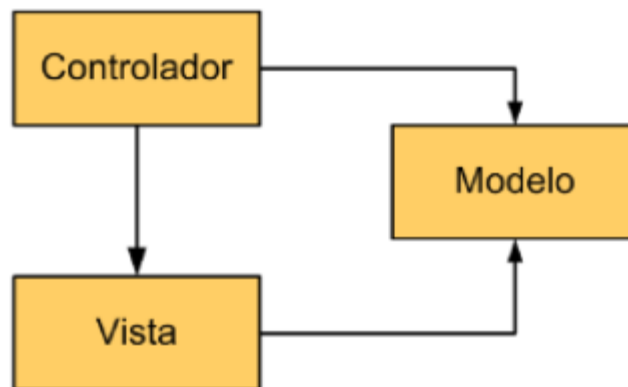


Figura 2.3. Modelo MVC. Berzal et al., s. f.

2.6 Framework para desarrollo web

Son entornos de trabajo que permiten el desarrollo de sitios web dinámicos y servicios web a través de plantillas, librerías y funcionalidades ya creadas. Gracias a estas herramientas se ahorra tiempo y dinero.

El framework que se utiliza para el desarrollo del lado del servidor es Laravel en su versión 8.0 que es uno de los frameworks más actuales y está en frecuente actualización.

Para el desarrollo frontend se utiliza el framework Bootstrap 4.6, mismo que fue sugerido por la Secretaría; además es el que instala por defecto la implementación de la plantilla de administración elegida. Una de sus ventajas es facilitar el modelado de sitios web gracias a su sistema de rejillas.

2.6.1 Framework para backend: Laravel 8.0

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC (García, 2015).

2.6.2 Framework para frontend: Bootstrap 4.6

Bootstrap es un framework CSS y Javascript que está diseñado para la creación de interfaces limpias y con un diseño responsivo, además, ofrece diversas herramientas y funciones, de manera que los usuarios pueden crear prácticamente cualquier tipo de sitio web haciendo uso de los mismos (Axarnet, s.f.).

2.7 Plantilla de panel de administración

Son plantillas personalizables, la mayoría de ellas están basadas Bootstrap, lo que permite que puedan adaptarse a diferentes dispositivos.

AdminLte3 es la plantilla de bootstrap que se utiliza para el diseño de las vistas del proyecto, a sugerencia de la Secretaría de Infraestructura y Obras Públicas, contiene una gran variedad de plugins que facilitan y ahorran las líneas de código; además, provee una plantilla de inicio de sesión que puede ser adaptada para cualquier aplicación web.

2.7.1 AdminLte3

AdminLTE es una plantilla de administración (figura 2.4) totalmente receptiva. Basado en el framework Bootstrap 4.6 y también en el complemento JS/jQuery. Altamente personalizable y fácil de usar. Se adapta a muchas resoluciones de pantalla, desde pequeños dispositivos móviles hasta grandes computadoras de escritorio (Smania, 2021).

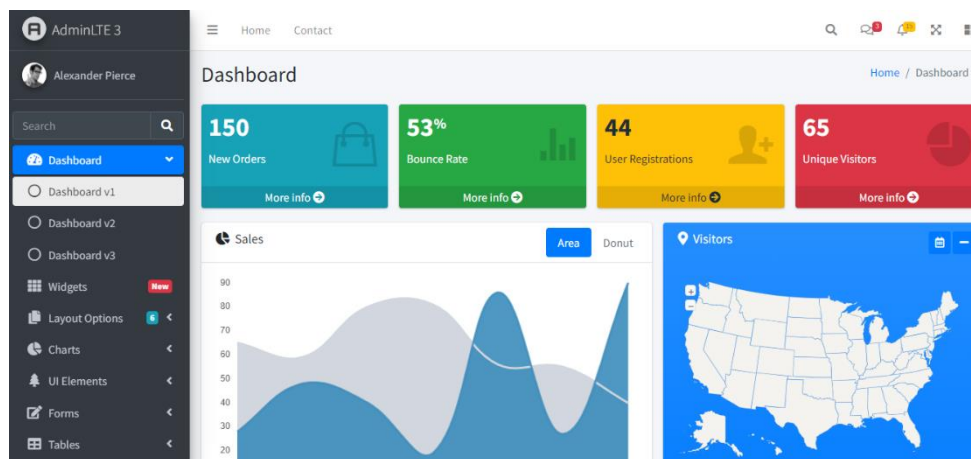


Figura 2.4. Plantilla AdminLTE. (AdminLTE, 2021).

2.8 Sistema Gestor de Base de Datos (SGBD)

Es una serie de programas encargados de crear, gestionar y administrar bases de datos, a su vez sirven como interfaz entre los usuarios y las aplicaciones, permite elegir las estructuras de datos necesarias para el almacenamiento y búsqueda de datos.

La base de datos del sistema se gestiona bajo MySQL, a través de la herramienta PhpMyAdmin utilizando un navegador web. Este SGBD es el que utilizan en el Departamento de Tecnologías de la Información, debido a que es principalmente gratuito y muy fácil de utilizar.

2.8.1 Sistema Gestor de base de datos Relacional: MySQL

MySQL es un sistema gestor de bases de datos (DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado,

es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha, esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo (Dataprix, 2009).

2.9 Generador de reportes PDF

Para generar los formatos PDF del sistema, es necesario hacer uso de la librería, DomPDF, la cual es una potente herramienta capaz de convertir código HTML a un archivo .pdf. A diferencia de otras tecnologías, DomPDF es constantemente actualizada.

2.9.1 Librería DomPdf

Es una librería compatible con CSS codificado en PHP, imágenes GIF, JPEG, BMP, entre otras. Permite generar archivos .pdf a partir de código HTML. Admite hojas de estilo externas, locales o a través de http/ftp (Arjol, 2014).

2.10 Estado del arte

Desde años atrás se han generado aplicaciones que implementan el patrón de arquitectura de software Modelo Vista Controlador, que por sus ventajas al momento de desarrollar programas es muy concurrido. A continuación, se muestra una descripción general de algunos proyectos para el control vehicular, la aplicación de la arquitectura MVC y software existente en el mercado para administrar vehículos de empresas.

2.10.1 Sistemas de Control Vehicular

Ortiz, Moncada y Moncada (2004) exponen en su trabajo de grado para optar por el título de Tecnología de Sistemas un software bajo el nombre “SISCONVEH” para el taller mecánico TECNILUGO que permite llevar un control de mantenimientos realizados a vehículos, administrar clientes y proveedores, facturar servicios y conocer su inventario de repuestos. El sistema utilizó para su desarrollo el lenguaje de programación Visual Basic 6.0. SISCONVEH tiene la capacidad de generar reportes del historial de reparaciones hechas por el taller, existencia de artículos, cotizaciones y facturas a través de la herramienta Crystal Reports.

Herrera y Vargas (2016) presentan en su trabajo de titulación la implementación de un sistema de Gestión Vehicular para el Departamento de Mantenimiento Vehicular de la Escuela Superior Politécnica de Chimborazo utilizando el framework JavaServer Faces basada en la arquitectura MVC y el sistema de base de datos PostgreSQL. Siguiendo la metodología SCRUM se logra llevar la información de los vehículos, choferes asignados, mantenimientos, reportes (existencias, servicios, unidades e información general) y diagnóstico de fallas mecánicas.

Bravo (2015) expone en su tesis de grado para obtener el título de Ingeniero en Sistemas Computacionales el desarrollo de un sistema de control vehicular con integración al sistema de localización automática para la Corporación Nacional de Electricidad. El software permite llevar el control de los vehículos, disponibilidad, choferes asignados, combustible consumido, kilómetros recorridos y ubicación en tiempo real. Implementa PHP, JSON, framework ZEND, Bootstrap y Google Charts.

Como Base de datos utiliza MySQL. Para localizar al vehículo emplea el dispositivo Skypatrol Evolution TT8750

2.10.2 Arquitecturas Modelo Vista Controlador

Jiménez, E., (2017) explica en su informe final para título profesional el desarrollo e implementación de una aplicación web para el manejo de catálogos en la empresa Turbomekanics utilizando la arquitectura Modelo Vista Controlador. La aplicación permite a los usuarios visualizar productos para añadirlos al carrito de compras y a los administradores gestionar el stock de los artículos. El sistema fue codificado con el lenguaje de programación PHP y como gestor de base de datos se empleó MySQL.

González *et al.*, (2021) propone en su artículo de investigación el desarrollo de una aplicación web basada en el patrón MVC para incrementar el desempeño académico en la asignatura matemáticas básicas. La aplicación funciona como material de apoyo para los estudiantes, además les permite visualizar la información vista en el aula de clases. Las tecnologías utilizadas para su desarrollo fueron SQL Server, Bootstrap v4.3, SQL Server Express 2012 y MS Visual Studio Community 2019.

2.10.3 Software existente en el mercado para control vehicular

Tuflota (s. f.) muestra en su página web oficial el desarrollo de un sistema que tiene como propósito administrar la flota vehicular de una empresa. “tuflota” ofrece distintas tarifas, la versión personal tiene características limitadas solo permite la gestión de 3 vehículos y un usuario. Mientras que el plan “tuflota micro” con un costo de \$235 al mes admite controlar 10 vehículos y proporciona usuarios infinitos. Entre las funciones de “tuflota” están: cálculo de rendimiento, bitácora de mantenimientos, inventario de vehículos, recordatorios, operadores, proveedores, incidentes, control vehículos y documentación en la nube. El sistema puede utilizarse sin conexión a internet desde cualquier dispositivo.

CloudFleet (s. f.) expone una aplicación en la nube basada en solicitudes y respuestas HTTP-JSON que le permite consultar, crear, modificar y eliminar recursos

e información en el sistema. El software “cloudFleet” gestiona los mantenimientos y planes de reparación dependiendo del vehículo. Lleva el control del combustible, analiza el consumo por carro y conductor. Maneja un inventario de repuestos a disposición del área de mantenimiento. Muestra la disponibilidad de vehículos y un control del detalle de los kilómetros/millas recorridos en un periodo de tiempo. Utiliza el método webHook para conectarse con aplicaciones externas ante la ocurrencia de un evento específico dentro de la aplicación.

Transportex (s. f.) es una aplicación Cliente-Servidor multiusuario que posibilita la gestión de flotas de transporte terrestre. Las funciones que ofrece son: control detallado de activos, organiza las tareas de mantenimiento avisando de manera automática el próximo servicio, registro de viajes que realizan los vehículos y el control de los conceptos de gastos operativos. Finalmente lleva el inventario de ítems combustibles, materiales y repuestos en almacenes propios. Utiliza como motor de base de datos Microsoft SQL Server Edición Express, en caso de ser necesario usa los servicios de Amazon Web Services.

Capítulo 3 Desarrollo

La metodología en cascada es un modelo secuencial para la gestión de proyectos, está constituida por 5 fases, cada una debe comenzar cuando la anterior ha finalizado. Al ser un modelo lineal permite ver en qué punto del proyecto se encuentra en un momento dado.

Para el desarrollo del sistema se siguieron las etapas de la metodología en cascada.

3.1 Requisitos

Durante la primer etapa se realizó un análisis de los procesos que se llevan a cabo al momento de dar de alta vehículos, departamentos, usuarios, cargas de combustible, servicios de mantenimiento y proveedores, para lo cual se creó un Diagrama de casos de uso.

Como siguiente paso se realizó un cronograma de actividades a seguir para llegar a la meta propuesta.

Finalmente, en base a los casos de uso del sistema se generó una tabla de requerimientos del sistema.

3.1.1 Casos de uso

Son utilizados para documentar el comportamiento del sistema desde el punto de vista del usuario.

Definición de actores:

- Usuario: Son los administradores del sistema.
- Sistema: Representa al programa web que hará las funciones que el usuario solicite.

Los casos de uso para agregar, eliminar y modificar se aplican para todas las tablas de la base de datos, las cuales son, departamentos, personal, proveedores, usuarios, vehículos, resguardos, combustibles, incidentes y mantenimientos.

En la figura 3.1 se visualiza el comportamiento general del Sistema de Control Vehicular.

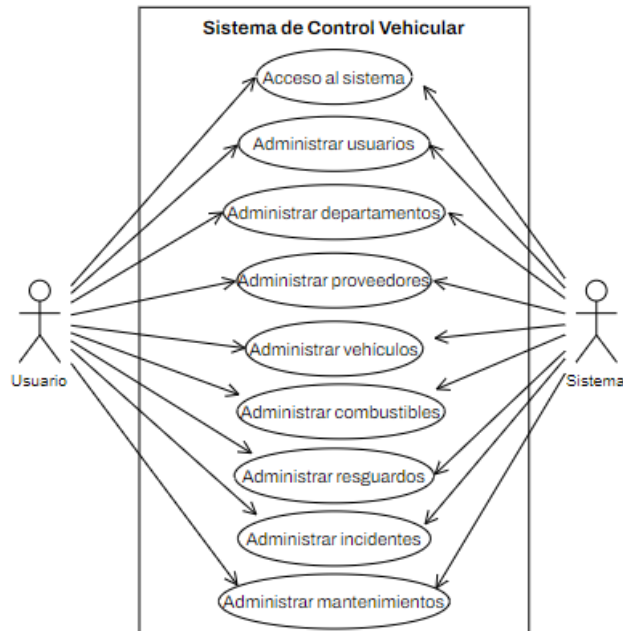


Figura 3.1. Sistema general. Autoría propia.

Para agregar un nuevo elemento a la base de datos, el escenario se ejemplifica en la figura 3.2, el usuario debe abrir el formulario de registro, el programa solicita datos que requiera de otras tablas, el administrador capturar los datos del objeto, posteriormente el sistema valida que los campos estén correctos y si cumple con las características guarda la información.

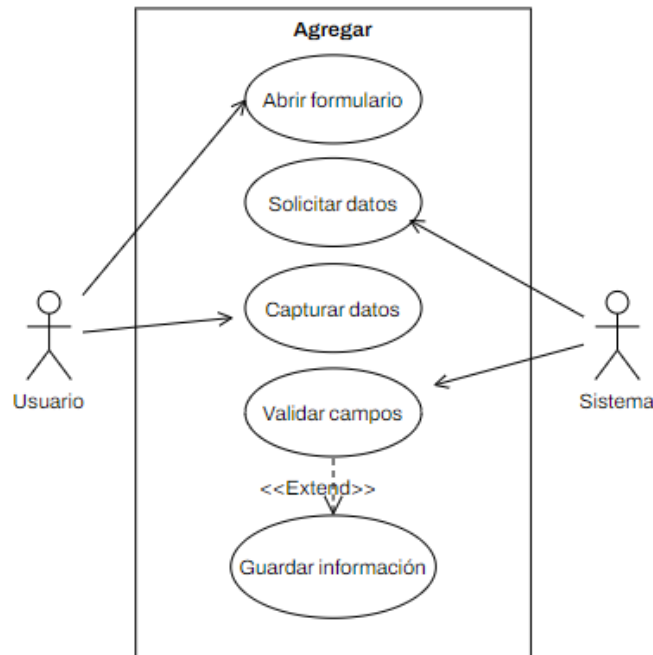


Figura 3.2. Agregar un registro. Autoría propia.

El caso de uso mostrado en la figura 3.3, permite al usuario modificar registros. Primero el usuario elige el registro, abre el formulario de edición, se le solicita al sistema datos necesarios de otros modelos, el administrador captura los datos, para que finalmente el programa valide y guarde el registro nuevamente.

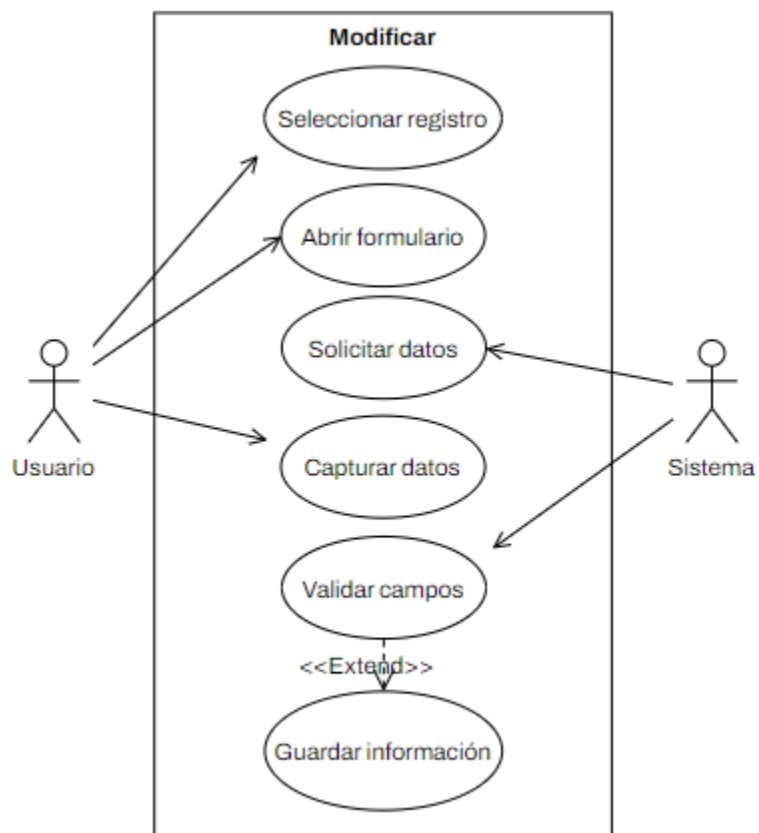


Figura 3.3. Modificar un registro. Autoría propia.

En la figura 3.4 se documenta el caso de uso que corresponde a la función de eliminar un registro.

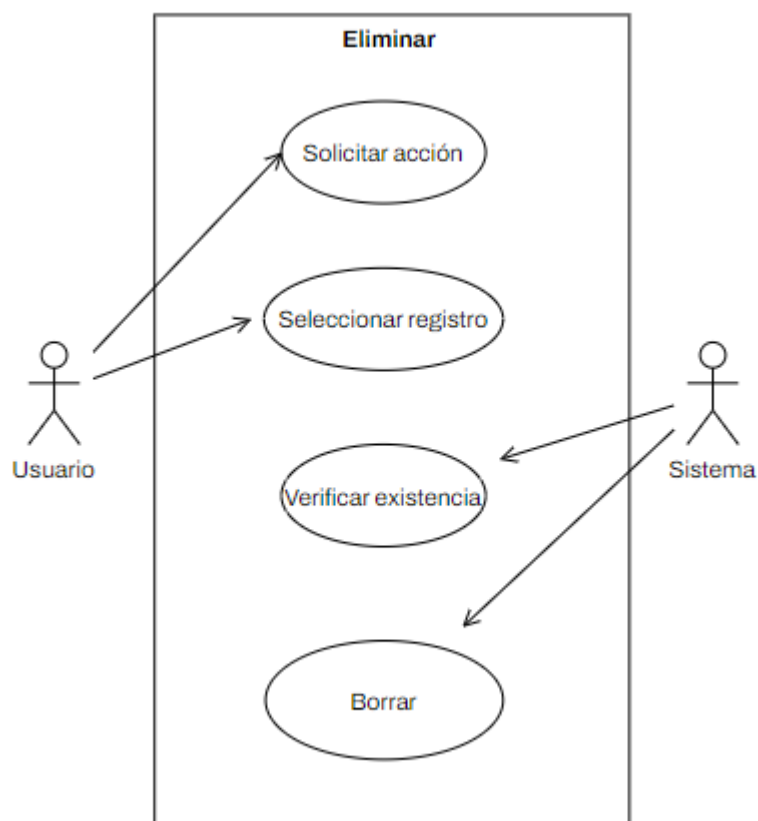


Figura 3.4. Borrar un registro. Autoría propia.

3.1.2 Cronograma de actividades

Con el fin de distribuir los tiempos entre las actividades, se realizó un cronograma (tabla 3.1), el cual muestra las tareas, la fecha de inicio y fin de cada una de ellas.

Tabla 3.1

Cronograma de actividades. Autoría propia.

Nombre	Fecha de inicio	Fecha de fin	Ago.	Sept.	Oct.	Nov.	Dic.	Ene.
Requerimientos del sistema.	02/08/21	04/08/21	■					
Elaboración de casos de uso.	05/08/21	11/08/21	■					
Diseño de la base de datos.	12/08/21	19/08/21	■					
Creación de la base de datos.	20/08/21	27/08/21	■					
Conexión de la base de datos al proyecto.	30/08/21	30/08/21		■				
Creación de los componentes del framework.	31/08/21	06/08/21		■				
Diseño de interfaces.	07/08/21	15/09/21		■				
Programación de los diferentes módulos.	16/09/21	26/10/21			■			
Revisión del funcionamiento.	27/10/21	01/10/21					■	
Corrección de errores.	02/11/21	30/11/21					■	
Pruebas del software.	1/12/21	17/12/21						■
Implementación del sistema.	17/01/22	20/01/22						■

3.1.3 Requerimientos

En la tabla 3.2 se muestran los requerimientos del sistema de Control Vehicular.

Tabla 3.2

Requerimientos del sistema. Autoría propia.

REQUERIMIENTOS DEL SISTEMA	
Módulos del sistema:	
<ul style="list-style-type: none"> • Departamentos • Personal • Proveedores • Usuarios • Padrón vehicular • Resguardos • Incidentes • Control de combustible • Servicio de mantenimiento • Mantenimientos pendientes • Licencias • Pólizas de seguro • Facturas 	
Campos de módulos:	
Departamentos	UP, dirección, departamento.
Personal	Nombre, teléfono, dirección, departamento, no. licencia de manejo, vencimiento de licencia, tipo de personal.
Proveedores	Razón social, localidad, RFC.
Usuarios	Nombre, correo electrónico, contraseña, usuario.
Padrón vehicular	No. económico, placa anterior, placa actual, tarjeta de circulación, estatus, color, marca, línea, tipo, modelo, condiciones, no. serie, no. motor, no. cilindros, imagen, empresa aseguradora, no. póliza de seguro, fecha de emisión y vencimiento de seguro, fecha de pago, ejercicio, costo de póliza, tipo de vehículo, numero de resguardo, UP, personal, dirección, departamento, kilometraje actual y anterior, litros actual y anterior, fecha de último mantenimiento.
Resguardos	Fecha, número de licencia, vigencia de licencia, estatus, observaciones, porcentaje de llantas, llanta de refacción, herramientas, tanque de gasolina, ejercicios, destino, número de resguardo, placa actual, no. económico, personal, departamento, dirección, UP, titular que entrega y autoriza.
Incidentes	Estado del incidente, detalles del incidente, fecha, imagen, personal que reporta, placa del vehículo, tipo de vehículo.
Control de combustible	Ejercicio, fecha de factura, fecha de captura, destino, litros, tipo de compra, tipo de combustible, otros combustibles, importe total, kilometraje actual y anterior, recorrido, rendimiento, ciudad, litros

	anteriores, número de factura, origen de recursos, número de resguardo, personal, placa actual, departamento, archivo de factura.			
Servicios de mantenimientos	Ejercicio, kilometraje, concepto de servicios, observaciones, fecha de factura, estatus, importe, forma de pago, descripción de pago, ciudad de carga, placa actual, estatus de factura, origen de recursos, número de factura, proveedor, personal, número de resguardo, archivo de factura.			
Mantenimientos pendientes	Numero de vehículo, placa, kilometraje actual, marca, línea, tipo, estatus.			
Licencias	Nombre, departamento, número de licencia, vencimiento de licencia, estatus.			
Pólizas de seguro	Número económico, placa, tarjeta de circulación, empresa aseguradora, número de póliza de seguro, fecha de emisión y vencimiento, fecha de pago, costo, estatus.			
Facturas	Combustible	Placa, número de resguardo, número de factura, fecha, importe total, origen de recursos, tipo de compra, tipo de combustible, litros, ciudad, archivo de factura.		
	Mantenimiento	Placa, número de resguardo, número de factura, fecha, importe, origen de recursos, forma de pago, descripción de pago, ciudad, estatus, archivo de factura.		
Funciones para los módulos				
Módulos	Agregar	Mostrar	Eliminar	Editar
Departamentos	x	x	x	x
Personal	x	x	x	x
Proveedores	x	x	x	x
Usuarios	x	x	x	
Padrón vehicular	x	x	x	x
Resguardos	x	x	x	x
Incidentes	x	x	x	x
Cargas de combustible	x	x	x	x
Servicio de mantenimiento	x	x	x	x
Mantenimientos pendientes		x		
Licencias		x		
Pólizas de seguro		x		

La tabla anterior está dividida en 3 partes, la primera indica los módulos que se necesitan agregar al nuevo sistema, la segunda parte muestra los campos que se listan en cada una de las secciones (departamentos, pólizas de seguro, facturas, etc.) y por último se muestra en una lista de verificación las funciones que pueden realizarse en cada módulo.

3.2 Diseño

3.2.1 Modelado de datos

Tomando en cuenta la base de datos ya creada, se hicieron algunas modificaciones, eliminando y agregando atributos.

La base de datos está compuesta por las siguientes tablas:

- Veh_catmarcas. Almacena todas las marcas de vehículos.
- Veh_cattipo. Guarda los tipos de automóviles en base a sus funciones y construcción.
- Veh_catlineas. Tiene como propósito reunir las líneas de automóviles.
- Veh_padron_vehicular. Almacena todos los vehículos que pertenecen a la Secretaría.
- Veh_incidentes. Guarda todos los sucesos repentinos que puedan ocurrirle a un vehículo mientras está en resguardo de un chofer.
- Veh_resguardo. Almacena todas las comisiones o resguardos que puede tener un vehículo.
- Veh_catdir. Guarda las diferentes direcciones generales de los departamentos de la Secretaría de Infraestructura y Obras Públicas.
- Veh_catdepto. Almacena los departamentos dentro de la dependencia.
- Veh_personal. Recopila a todos los choferes que tienen la capacidad de conducir un vehículo y llevarlos a una comisión dentro del estado.

3.2.2.3 Creación de la base de datos.

Tomando en cuenta el modelo Diagrama de Clases (figura 3.6), se hicieron las migraciones, una por cada tabla. Posteriormente se ejecutaron en la consola Artisan, con ello, las tablas se reflejaron en PhpMyAdmin.

3.2.2.4 Creación de Seeders.

Estas clases fueron utilizadas para inicializar tablas de la base de datos, como, por ejemplo: departamentos, direcciones, origen de recursos, proveedores, tipos de vehículo, marcas y líneas de autos.

3.2.2.5 Creación de modelos.

Como siguiente paso se creó un modelo para cada tabla, que sirve como medio de interacción con la base de datos. Cada modelo contiene el nombre de la clase en singular, y dentro de ellos viene una estructura donde muestra el alias de la tabla, llave primaria con su respectivo tipo de dato, los atributos, y por ultimo las relaciones con otras tablas.

3.2.2.6 Creación de controladores.

Los controladores se crearon con el objetivo de atender las solicitudes entrantes como crear, eliminar, editar y actualizar un registro. Estos archivos pueden ser creados en conjunto con los modelos. En este apartado se realizan las consultas mostradas en el sistema.

3.2.2.7 Creación de vistas.

Dentro del directorio *resources/views* se crearon carpetas, cada una corresponde a un módulo del sistema. En los folders se establecieron las vistas (*views*) con la extensión *.blade.php*. Es aquí donde se codificó el *FrontEnd*, lo que se le muestra al usuario. La *view* para mostrar los datos de las tablas se denominó *mostrar+nombre* de la clase, para añadir un nuevo elemento *agregar+nombre* de la clase y finalmente para modificar un objeto *editar+nombre* de la clase.

3.2.2.8 Creación de rutas.

Una vez definidas las vistas y los controladores el paso siguiente fue crear las rutas, mismas que se establecieron en el archivo web.php, el cual es un archivo que carga automáticamente en el proyecto. El propósito de estas fue mostrar cada una de las vistas, y enlazarlas con su controlador.

3.2.2.9 Programación de los módulos.

Los módulos desarrollados fueron los siguientes:

- Panel. Está es la vista principal dentro del sistema. Muestra información introductoria del programa. Como por el ejemplo el número de vehículos registrados, para lo cual se hizo una consulta en los controladores, para que hiciera un conteo a la tabla de “padrón vehicular”.
- Departamentos. Son las áreas dentro de la Secretaría.
- Proveedores. Son empresas que realizan los mantenimientos a los vehículos.
- Personal. Llamados también “choferes”, pueden ser de base o eventuales. Son quienes llevan de comisión a las unidades.
- Usuarios. Este módulo se encarga de gestionar a los usuarios del sistema de Control Vehicular.
- Padrón vehicular. Aquí se administran los vehículos que son propiedad de la SIOP.
- Resguardos. Cada vez que un chofer lleva un carro a una obra o comisión debe llenar esta tabla, misma que guarda información de la unidad, la ciudad a donde viaja y datos de él.
- Mantenimientos. En este apartado se muestra un historial de los servicios realizados a los vehículos.

- Mantenimientos pendientes. La función de este módulo es listar a los vehículos que requieren servicios, en base a el kilometraje (cada 10000 km) o a la fecha (cada año).
- Combustibles. Cada vez que se hace un resguardo el chofer se encarga de llevar al vehículo a cargar combustible y facturar la compra, por lo que debe llenar este formulario y adjuntar de ser posible el documento.

3.2.2.10 Creación de formatos.

Los formatos fueron creados con la librería DomPDF, código HTML5 y CSS3. Los archivos generados son únicamente para los resguardos, donde automáticamente recopila los datos del vehículo que va a comisión y del chofer responsable.

3.3 Implementación

Una vez creada la base de datos y codificado el sistema, se implementó el sistema en el servidor de la secretaría, para lo que fue necesario ingresar con una dirección IP al emulador PuTTY como se muestra en la figura 3.7.

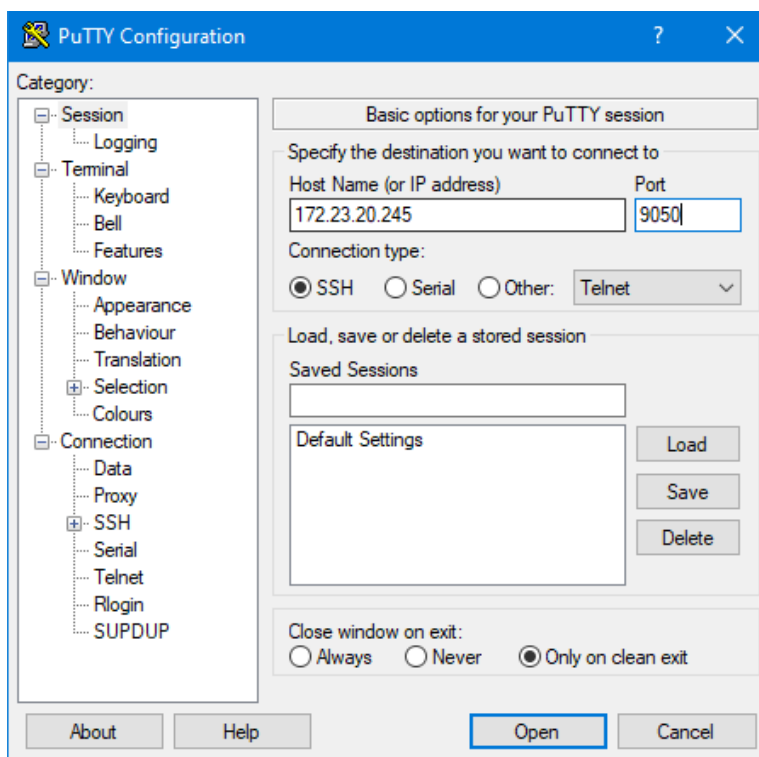


Figura 3.7. Cliente SSH PuTTY. Autoría propia.

Posteriormente, con ayuda del programa VNC Viewer, de manera remota se tomó el control del ordenador servidor, en este paso se implantó el proyecto y finalmente se corrieron las migraciones para crear la base de datos y poder así ejecutar el sistema desde cualquier computadora dentro de la red de la Secretaría.

3.4 Verificación

En esta etapa se realizaron las pruebas del sistema, comprobando que la información se guardara correctamente en la base de datos, verificando que se mostraran las ventanas únicamente si el usuario tenía una sesión activa, asegurando que los formatos .pdf se descargaran con los datos solicitados y finalmente que el sistema en general no presentara algún inconveniente.

Como siguiente paso se llevaron a cabo pruebas del sistema por parte del Departamento de Recursos Materiales y Servicios Generales para asegurarse del buen funcionamiento de cada uno de los módulos y verificar que cada uno de los campos estuviera correcto.

3.5 Mantenimiento

Una vez realizadas las correcciones correspondientes, el sistema fue utilizado por la Oficina de Control Vehicular, se llevaron a cabo observaciones de posibles requerimientos nuevos.

Capítulo 4 Resultados

4.1 Patrón de diseño: MVC

En este apartado se muestran los resultados obtenidos, algunas imágenes han sido difuminadas por motivo de confidencialidad.

4.1.1 Nivel modelo

4.1.1.1 Migraciones.

Se crearon migraciones, las cuales sirven como un control de versiones de la base de datos.

4.1.1.1.1 Módulo departamento.

La tabla veh_catdepto está constituida por llaves foráneas y una serie de atributos correspondientes al módulo. Dentro de la estructura de la migración (figura 4.1) se definió la relación con la tabla veh_catdir.

```

.....public function up()
.....{
.....    Schema::create('veh_catdepto', function (Blueprint $table) {
.....        $table->string('cve_depto', 15);
.....        $table->string('desc_departamento', 150);
.....        $table->string('cve_dir', 15)->nullable();
.....        $table->string('cveup', 15)->nullable();
.....        $table->timestamps();
.....        $table->primary('cve_depto');
.....        $table->foreign(['cve_dir', 'cveup'])
.....            ->references(['cve_dir', 'cveup'])
.....            ->on('veh_catdir')
.....            ->onUpdate('cascade')
.....            ->onDelete('set null');
.....    });
.....}

```

Figura 4.1. Migración de la tabla Departamento. Autoría propia.

4.1.1.1.2 Módulo personal.

Para el módulo del personal se creó la tabla veh_personal dentro de la migración (figura 4.2), se implementó una llave incremental y los atributos del personal, al final se agregaron las llaves foráneas.

```
public function up()
{
    Schema::create('veh_personal', function (Blueprint $table) {
        $table->bigIncrements('id_personal');
        $table->string('nombre', 150);
        $table->string('telefono', 10)->nullable();
        $table->string('direccion', 250)->nullable();
        $table->string('licencia_manejo', 9);
        $table->date('vencimiento_licencia');
        $table->string('tipo_personal', 10);
        $table->timestamps();
        $table->string('cve_depto', 15)->nullable();
        $table->foreign('cve_depto')->references('cve_depto')->on('veh_catdepto')
        ->onUpdate('cascade')
        ->onDelete('set null');
    });
}
```

Figura 4.2. Migración de la tabla Personal. Autoría propia.

4.1.1.1.3 Módulo proveedores.

En la figura 4.3 se muestra la estructura de la tabla veh_cat_proveedores.

```
public function up()
{
    Schema::create('veh_cat_proveedores', function (Blueprint $table) {
        $table->bigIncrements('cve_proveedor');
        $table->string('razon_social', 150);
        $table->string('localidad', 150);
        $table->string('rfc', 15);
        $table->timestamps();
    });
}
```

Figura 4.3. Migración de la tabla Proveedor. Autoría propia.

4.1.1.1.4 Módulo usuarios.

La migración (figura 4.4) de la tabla users se generó por defecto al crear el proyecto, sin embargo, se le hicieron algunas modificaciones al código.


```

public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('usuario')->unique();
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}

```

Figura 4.4. Migración de la tabla Usuario. Autoría propia.

4.1.1.1.5 Módulo padrón vehicular.

En la figura 4.5 se muestran solo algunos atributos de la tabla veh_padrón_vehicular, de igual manera las restricciones que se le aplican a la base de datos tomando en cuenta las tablas con la que está relacionada.

```

public function up()
{
    Schema::create('veh_padrón_vehicular', function (Blueprint $table) {
        $table->integerIncrements('id_veh');
        $table->integer('no_eco')->nullable();
        $table->string('placa_act', 9);
        $table->string('placa_ant', 9)->nullable();
        $table->string('t_circula', 30)->nullable();
        $table->string('estatus', 15)->nullable();
        $table->string('color', 15)->nullable();
        $table->integer('modelo')->nullable();
        $table->string('condiciones', 15)->nullable();
        $table->string('no_serie', 30)->nullable();
        $table->string('no_motor', 20)->nullable();
        $table->integer('no_cilindros')->nullable();
        $table->string('file_path', 250)->nullable();
        $table->string('empresa_seg', 150)->nullable();

        $table->unsignedBigInteger('cve_marca')->nullable();
        $table->foreign('cve_marca')->references('cve_marca')->on('veh_catmarcas')->onUpdate('cascade')->onDelete('set null');

        $table->unsignedBigInteger('cve_linea')->nullable();
        $table->foreign('cve_linea')->references('cve_linea')->on('veh_catlineas')->onUpdate('cascade')->onDelete('set null');
    });
}

```

Figura 4.5. Migración de la tabla Padrón Vehicular. Autoría propia.

4.1.1.1.6 Módulo resguardos.

Los resguardos se almacenan en la tabla veh_resguardo, misma que fue creada a través de una migración (4.6) y que contiene los campos a llenar en el formulario. También se añadió al final la integridad referencial.

```

public function up()
{
    Schema::create('veh_resguardo', function (Blueprint $table) {
        $table->bigIncrements('id_resguardo');
        $table->string('titular_aut', 50);
        $table->string('titular_ent', 50);
        $table->date('fecha');
        $table->string('no_licencia', 15);
        $table->date('vigencia_lic');
        $table->string('estatus', 15);
        $table->string('observaciones', 250)->nullable();
        $table->double('llanta_del_d', 16, 4)->nullable();
        $table->double('llanta_del_i', 16, 4)->nullable();
        $table->double('llanta_tras_d', 16, 4)->nullable();
        $table->double('llanta_tras_i', 16, 4)->nullable();
        $table->double('llanta_refac', 16, 4)->nullable();
        $table->string('herramientas', 150)->nullable();
        $table->string('tanque_gas', 10)->nullable();
        $table->integer('ejercicio');
        $table->string('sino', 10)->nullable();
        $table->string('destino', 100)->nullable();
        $table->string('placa_act', 9)->nullable();
        $table->string('cve_dir', 150)->nullable();
        $table->string('cveup', 150)->nullable();
        $table->string('cve_depto', 150)->nullable();
        $table->unsignedInteger('id_veh')->nullable();
        $table->foreign('id_veh')->references('id_veh')->on('veh_padron_vehicular')->onUpdate('cascade')->onDelete('set null');
        $table->unsignedBigInteger('id_personal')->nullable();
        $table->foreign('id_personal')->references('id_personal')->on('veh_personal')->onUpdate('cascade')->onDelete('set null');
        $table->timestamps();
    });
}

```

Figura 4.6. Migración de la tabla Resguardo. Autoría propia.

4.1.1.1.7 Módulo combustible.

En la estructura de la migración (figura 4.7) de la tabla veh_combustibles se encuentran solo algunos de los atributos y las relaciones con otras tablas.

```

public function up()
{
    Schema::create('veh_combustibles', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->integer('ejercicio');
        $table->date('fecha_factura')->nullable();
        $table->date('fecha_captura');
        $table->string('destino', 150)->nullable();
        $table->double('lts', 9, 5);
        $table->string('tipo_compra', 15);
        $table->string('tipo_comb', 50)->nullable();
        $table->string('detalle_comb', 100)->nullable();
        $table->double('importe_total', 16, 8);
        $table->double('km_actual', 16, 10);
        $table->timestamps();

        $table->unsignedBigInteger('id_personal')->nullable();
        $table->foreign('id_personal')->references('id_personal')->on('veh_personal')->onUpdate('cascade')->onDelete('set null');
    });
}

```

Figura 4.7. Migración de la tabla Combustible. Autoría propia.

4.1.1.1.8 Módulo mantenimiento.

En la figura 4.8 se visualiza la migración de la tabla veh_mantenimiento que guarda los servicios realizados a los vehículos, primero se muestran solo algunos atributos y relaciones con otras tablas.

```

public function up()
{
    Schema::create('veh_mantenimiento', function (Blueprint $table) {
        $table->bigIncrements('id_man');
        $table->integer('ejercicio');
        $table->double('km', 16, 10);
        $table->string('concepto', 150)->nullable();
        $table->string('observaciones', 100)->nullable();
        $table->date('fecha_fact')->nullable();
        $table->string('estatus', 25)->nullable();
        $table->string('no_factura', 30);
        $table->double('importe_fact', 9, 5);
        $table->string('forma_pago', 30);
        $table->string('desc_pago', 250)->nullable();
        $table->string('origen_factura', 150);
        $table->string('estatus_fact', 15);
        $table->string('placa_act', 9)->nullable();
        $table->string('file_path', 250)->nullable();
        $table->timestamps();

        $table->unsignedBigInteger('cve_origen')->nullable();
        $table->foreign('cve_origen')->references('cve_origen')->on('veh_catorigen')->onUpdate('cascade')->onDelete('set null');
    });
}

```

Figura 4.8. Migración de la tabla Mantenimiento. Autoría propia.

4.1.1.2 Modelos

Los modelos para las tablas, se crearon y estructuraron de manera similar. Cada *model* se nombró de acuerdo a los módulos, y dentro de las clases se asentó el nombre de la tabla en la base de datos, su llave primaria y sus características, después se agregaron todos los atributos incluyendo la clave primaria. Finalmente, en forma de métodos se definieron las relaciones con las otras tablas. La figura 4.9 muestra un ejemplo del modelo Proveedor.

```

class Proveedor extends Model
{
    protected $table = 'veh_cat_proveedores';
    protected $primaryKey = 'cve_proveedor';
    public $incrementing = true;
    protected $keyType = 'integer';
    public $timestamps = false;
    protected $fillable = ['cve_proveedor', 'razon_social', 'localidad', 'rfc'];

    public function mantenimiento()
    {
        return $this->hasMany(Mantenimiento::class, 'cve_proveedor');
    }
}

```

Figura 4.9. Modelo Proveedor. Autoría propia.

4.1.2 Nivel vista

4.1.2.1 Login.

La primera ventana visible para los usuarios del sistema es el “Login” (figura 4.10), donde los datos necesarios para acceder son el usuario y la contraseña.

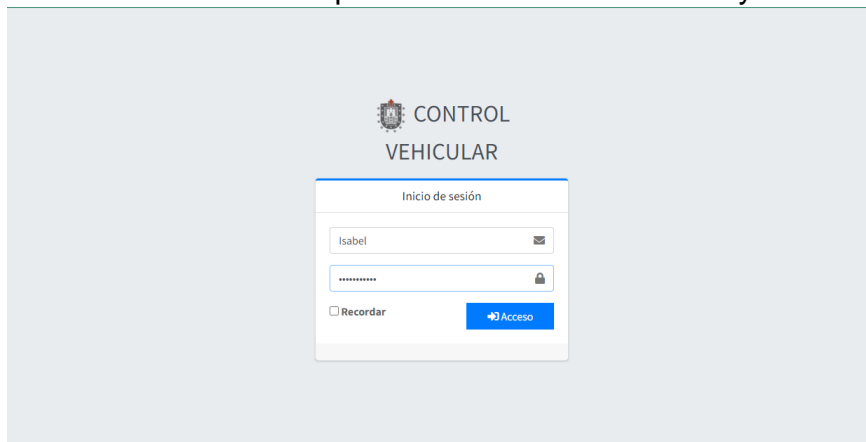
The image shows a login interface for a system titled "CONTROL VEHICULAR". At the top center, there is a logo of a coat of arms followed by the text "CONTROL VEHICULAR". Below this, a white box contains the login form. The form has a title "Inicio de sesión". It features a text input field with "Isabel" entered, a password field with masked characters "*****", a checkbox labeled "Recordar", and a blue button labeled "Acceso" with a right-pointing arrow icon.

Figura 4.10. Formulario para iniciar sesión. Autoría propia.

4.1.2.2 Panel principal.

El panel principal o también llamado “Dashboard”, es una página que muestra información inicial de los módulos principales (figura 4.11). Al lado derecho aparece un menú con los siguientes apartados:

- Catálogos: Contiene los formularios para agregar, mostrar y editar departamentos, personal, usuarios y proveedores.
- Padrón vehicular: Módulo de vehículos propiedad de la Secretaría.
- Resguardos: Guarda el historial de las comisiones a las que va un automóvil.
- Control de combustible: Muestra las cargas de combustible de los vehículos.
- Control de serv. de mantenimiento: Es el módulo encargado de llevar el control de los mantenimientos pendientes y los realizados.
- Licencias: En ese apartado se visualizan las licencias vencidas de los choferes.

- Póliza de seguro: Contiene un listado de las pólizas de seguro vehiculares expiradas hasta la fecha actual.
- Facturas: Estos documentos están divididos por mantenimientos y carga de combustibles.

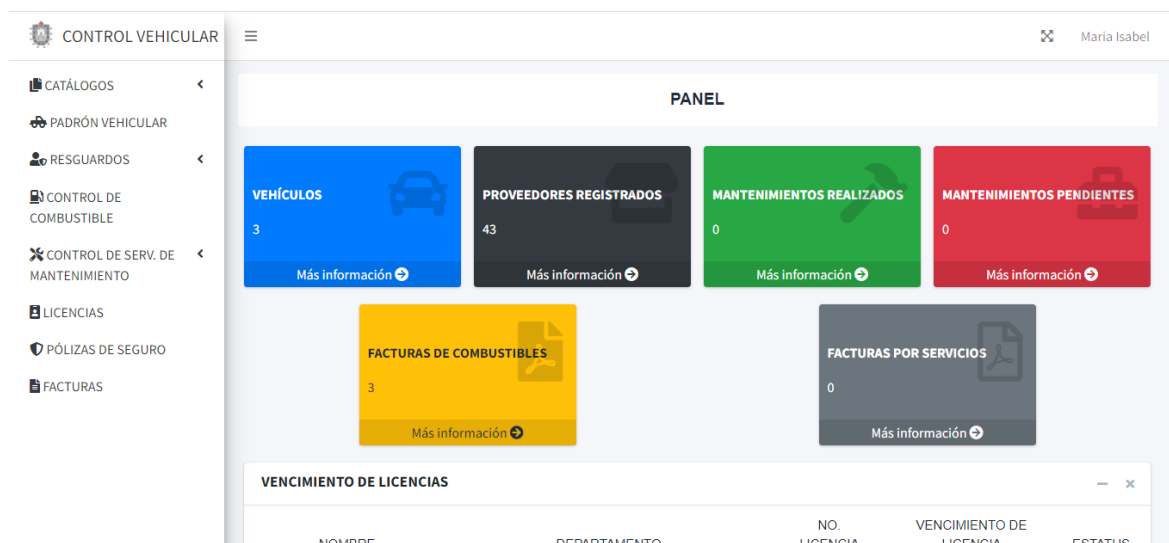


Figura 4.11. Tablero principal. Autoría propia.

4.1.2.3 Departamentos.

La vista principal (figura 4.12) de los departamentos muestra una tabla con las áreas que se van agregando al sistema, como por ejemplo el departamento de la Unidad de Transparencia. La tabla contiene un buscador y permite ordenarlos en orden ascendente y descendente. En la columna de acciones se tienen las funciones de editar y eliminar un departamento.

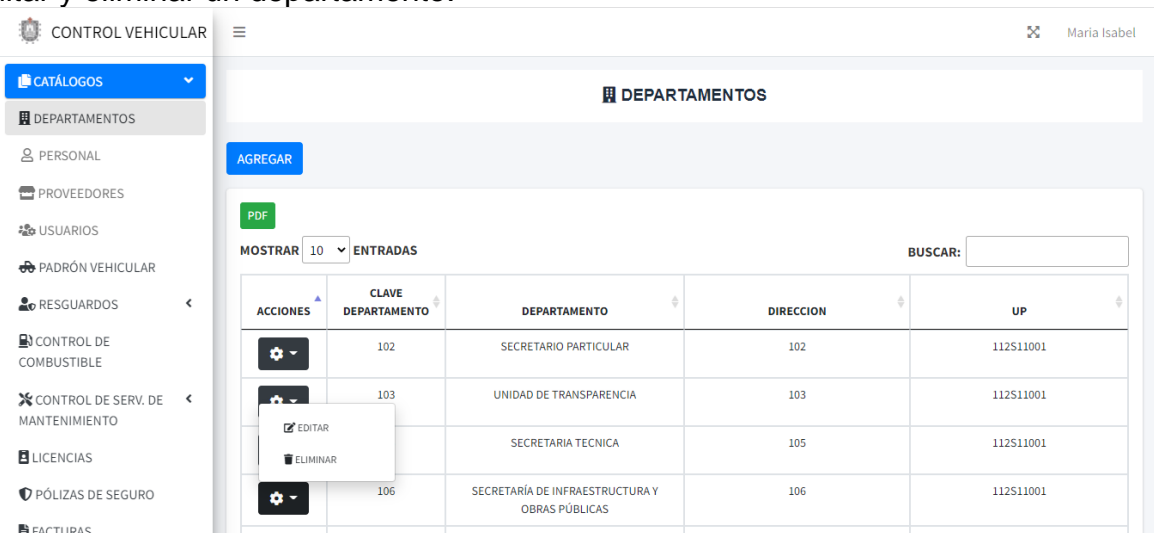
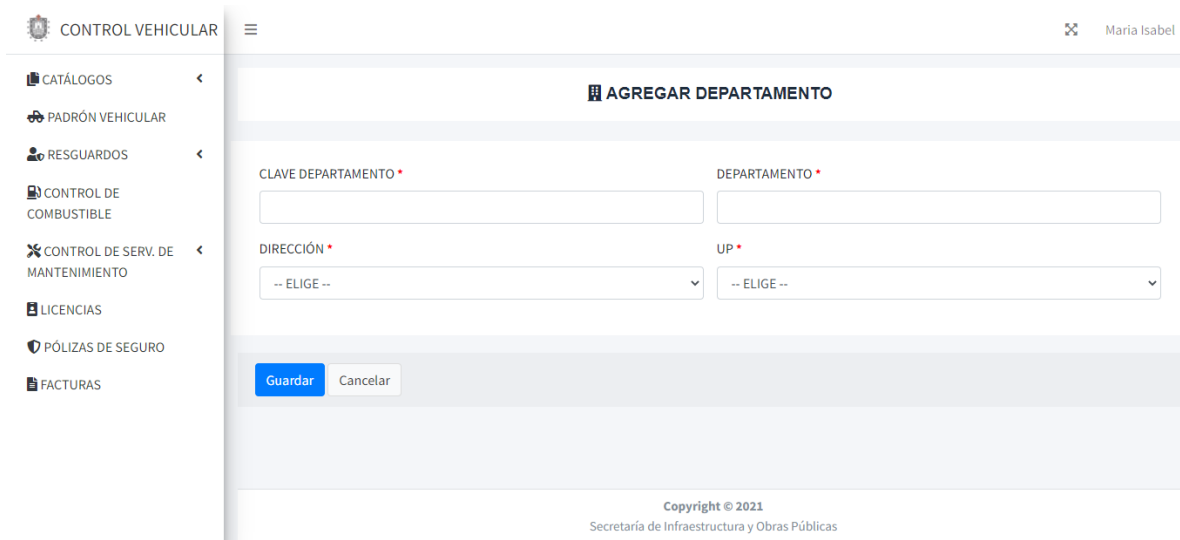


Figura 4.12. Vista principal de los departamentos. Autoría propia.

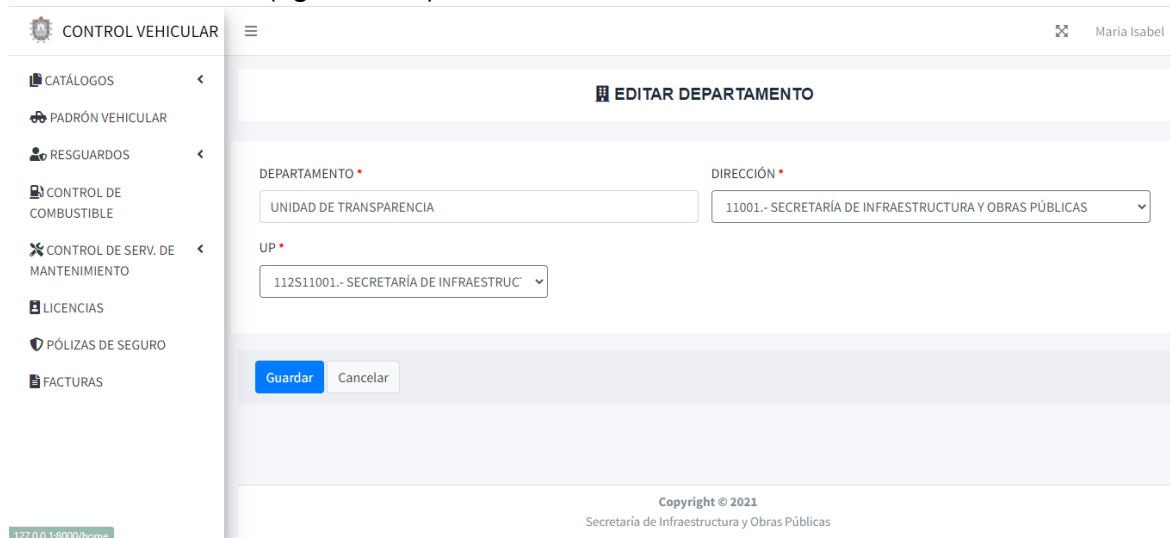
Con el botón agregar ubicado en la vista principal de los departamentos, los usuarios abren una página que se encarga de mostrar un formulario para añadir un nuevo departamento (figura 4.13).



The screenshot shows a web application interface for 'CONTROL VEHICULAR'. On the left is a navigation menu with items: CATÁLOGOS, PADRÓN VEHICULAR, RESGUARDOS, CONTROL DE COMBUSTIBLE, CONTROL DE SERV. DE MANTENIMIENTO, LICENCIAS, PÓLIZAS DE SEGURO, and FACTURAS. The main content area is titled 'AGREGAR DEPARTAMENTO'. It contains four input fields: 'CLAVE DEPARTAMENTO', 'DEPARTAMENTO', 'DIRECCIÓN', and 'UP'. The 'DIRECCIÓN' and 'UP' fields are dropdown menus with '-- ELIGE --' as the selected option. At the bottom of the form are two buttons: 'Guardar' (highlighted in blue) and 'Cancelar'. The footer of the page includes 'Copyright © 2021' and 'Secretaría de Infraestructura y Obras Públicas'.

Figura 4.13. Formulario para agregar un departamento. Autoría propia.

Dentro de la vista de Departamentos se visualiza el botón de acciones que tiene la función de editar (figura 4.14).



The screenshot shows the 'EDITAR DEPARTAMENTO' form in the 'CONTROL VEHICULAR' system. The navigation menu is the same as in Figure 4.13. The main content area is titled 'EDITAR DEPARTAMENTO'. It contains three input fields: 'DEPARTAMENTO' (text input with 'UNIDAD DE TRANSPARENCIA'), 'DIRECCIÓN' (dropdown menu with '11001.- SECRETARÍA DE INFRAESTRUCTURA Y OBRAS PÚBLICAS'), and 'UP' (dropdown menu with '112S11001.- SECRETARÍA DE INFRAESTRUC'). At the bottom of the form are two buttons: 'Guardar' (highlighted in blue) and 'Cancelar'. The footer of the page includes 'Copyright © 2021' and 'Secretaría de Infraestructura y Obras Públicas'. A small URL '127.0.0.1:8000/home' is visible in the bottom left corner.

Figura 4.14. Formulario editar departamentos. Autoría propia.

4.1.2.4 Personal.

El apartado de personal (figura 4.15 muestra los choferes que pueden llevar a resguardos los vehículos de la Secretaría. Dentro de las acciones se encuentra el botón de eliminar que borra del sistema al personal.

The screenshot displays the 'PERSONAL' management interface. On the left is a sidebar with a menu including 'CATÁLOGOS', 'DEPARTAMENTOS', 'PERSONAL', 'PROVEEDORES', 'USUARIOS', 'PADRÓN VEHICULAR', 'RESGUARDOS', 'CONTROL DE COMBUSTIBLE', 'CONTROL DE SERV. DE MANTENIMIENTO', 'LICENCIAS', 'PÓLIZAS DE SEGURO', and 'FACTURAS'. The main content area is titled 'PERSONAL' and features an 'AGREGAR' button. Below it is a table with the following data:

ACCIONES	NOMBRE(S)	TELÉFONO	DEPARTAMENTO	NO. LICENCIA
[Settings icon]	MARIA ISABEL ALEJO AQUINO	2351055187	DEPARTAMENTO DE TECNOLOGIAS DE LA INFORMACIÓN	01N450316
[Edit icon]	JUAN GOMEZ	2351055187	SECRETARIO PRIVADO	815876978

At the bottom of the table, there are navigation controls: 'MOSTRANDO', 'VER DETALLES', 'ANTERIOR', '1', and 'SIGUIENTE'. A footer contains the text 'Copyright © 2021 Secretaría de Infraestructura y Obras Públicas'.

Figura 4.15. Página principal de Personal. Autoría propia.

El apartado de agregar (figura 4.16) tiene la capacidad de añadir un nuevo chofer.

The screenshot shows the 'AGREGAR PERSONAL' form. It is organized into two main sections:

- DATOS PERSONALES:** Contains input fields for 'NÚMERO DE PERSONAL', 'NOMBRE(S) Y APELLIDOS', 'TELÉFONO', and 'DIRECCIÓN'.
- OTROS DATOS:** Contains dropdown menus for 'DEPARTAMENTO' and 'TIPO DE PERSONAL', and a text field for 'NO. LICENCIA'. Below these is a 'VENCIMIENTO DE LICENCIA' field.

The sidebar on the left is identical to the previous screenshot, with 'PERSONAL' selected in the menu.

Figura 4.16. Formulario agregar personal. Autoría propia.

El botón editar (figura 4.17) lleva a un formulario para modificar un chofer,

Figura 4.17. Formulario para editar personal. Autoría propia.

El botón “ver detalles” abre un modal (figura 4.18) que enlista los datos que no se pueden ver en la tabla principal.

Figura 4.18. Modal de detalles de personal. Autoría propia.

4.1.2.5 Proveedores.

Los proveedores encargados de dar mantenimiento a los vehículos, se visualizan en la pestaña de “proveedores” (figura 4.19). En la columna de acciones se encuentran las funciones eliminar, editar y ver detalles.

ACCIONES	ID	RAZÓN SOCIAL	LOCALIDAD	RFC
	1	AUTOZ	XALAPA, VER.	ROSM70070788A
	2	SALAZAR ROSAS CELIA/TAPICERÍA BETOS.	XALAPA, VER.	SARC550108JM6
	3	AUTO SERVICIO "EL GOLPE"	XALAPA, VER.	AALH520807UA1
	4	SERV. AUT. ANDRADE/ROGELIO AGUILAR PENSADO	XALAPA, VER.	AUPR7704152XA
	5	SERVICIO "LA ROTONDA"	XALAPA, VER.	SRO981009491

Figura 4.19 Pantalla principal proveedores. Autoría propia.

El botón para agregar situado en la ventana de proveedores dirige a una ventana (figura 4.20) que permite añadir un nuevo elemento, donde requiere los campos: razón social, localidad y RFC.

Copyright © 2021
Secretaría de Infraestructura y Obras Públicas

Figura 4.20. Formulario agregar un proveedor. Autoría propia.

La ventana editar (figura 4.21) permite modificar un proveedor seleccionado desde la tabla principal (figura 4.19).

Figura 4.21. Formulario modificar al personal. Autoría propia.


4.1.2.6 Usuarios.

Los usuarios son aquellos que tienen acceso al Sistema de Control Vehicular, la página principal del módulo (figura 4.22) muestra a todos los usuarios dados de alta, dentro de las funciones que pueden realizarse con los usuarios se encuentran buscar y eliminar.

ACCIONES	ID	NOMBRE	EMAIL	USUARIO
	1	MARIA ISABEL	ISAAQUI9805@GMAIL.COM	ISABEL

Figura 4.22. Vista principal de los usuarios registrados. Autoría propia.

El botón “agregar” de la ventana principal de usuarios abre un formulario (figura 4.23) donde completan datos, como el nombre, usuario, correo y contraseña.



El formulario de registro de usuarios se encuentra centrado en la pantalla. En la parte superior, se muestra el escudo de la Secretaría y el título "CONTROL VEHICULAR". Debajo, un recuadro blanco con el título "CREAR UNA CUENTA" contiene los siguientes campos de texto:

- NOMBRE COMPLETO (con ícono de persona)
- USUARIO (con ícono de persona)
- CORREO ELECTRONICO (con ícono de correo)
- CONTRASEÑA (con ícono de candado)
- CONFIRMAR CONTRASEÑA (con ícono de candado)

Al final del formulario hay un botón azul con el texto "REGISTRARSE" y un ícono de persona.

Figura 4.23. Formulario de registro de usuarios. Autoría propia.

4.1.2.7 Padrón vehicular.

El módulo de padrón vehicular (figura 4.24), administra los automóviles que son propiedad de la Secretaría, al seleccionar el icono de acciones de una unidad específica se muestran las siguientes funciones:

- **Editar:** Tiene como propósito modificar un vehículo que ya está registrado, para lo cual, se abre un formulario (figura 4.26) con todos los campos ya guardados del objeto.
- **Eliminar:** Borra un vehículo de la base de datos.
- **Ver detalles:** Muestra una vista detallada (figura 4.27) de todos los atributos de un automóvil.
- **Resguardo:** Al hacer clic en el botón, se abre el formulario para agregar un resguardo, donde ya está seleccionada la placa del vehículo en el que se hizo la acción.
- **Mantenimiento:** Abre un formulario para añadir un nuevo mantenimiento del automóvil seleccionado.

The screenshot shows the 'PADRON VEHICULAR' (Vehicle Register) interface. On the left is a navigation menu with options like 'CATÁLOGOS', 'PADRÓN VEHICULAR', 'RESGUARDOS', etc. The main area features a search bar and a table of vehicles. A context menu is open over the first row, showing options: 'AGREGAR', 'EDITAR', 'ELIMINAR', 'VER DETALLES', 'RESGUARDO', and 'MANTENIMIENTO'. The table has columns: 'IMAGEN', 'NO. ECONÓMICO', 'TIPO DE VEHÍCULO', 'PLACA ACTUAL', 'MARCA', 'TIPO', 'LINEA', 'POLIZA SEGURO', and 'T. CIRCULACIÓN'. The first row contains a tractor icon, '501', 'PARTICULAR', 'PLHJ-5', 'FIAT', 'COLORADO', 'DOBLE CABINA', and '500'. Below the table, it says 'MOSTRANDO PÁGINA 1 DE 1' and has navigation buttons 'ANTERIOR', '1', and 'SIGUIENTE'.

Figura 4.24. Vista principal de los vehículos registrados. Autoría propia.

La ventana para agregar vehículos (figura 4.25) permite añadir un nuevo elemento al padrón vehicular, donde a través de un formulario se guarda la información en la base de datos.

The screenshot shows the 'AGREGAR VEHÍCULO' (Add Vehicle) form. It is divided into two sections: 'CONTROL' and 'CARACTERÍSTICAS'. The 'CONTROL' section contains the following fields: 'NO. ECONOMICO' (text input), 'EJERCICIO *' (text input), 'TARJ. CIRCULACION' (text input), 'PLACA ACTUAL *' (text input), 'PLACA ANTERIOR' (text input), and 'ESTATUS *' (dropdown menu with '-- ELIGE --'). The 'CARACTERÍSTICAS' section contains a 'TIPO DE VEHÍCULO *' dropdown menu with '-- ELIGE --'. The interface includes a navigation menu on the left and a user profile 'Maria Isabel' in the top right.

Figura 4.25. Formulario para añadir un vehículo. Autoría propia.

Para editar un vehículo se selecciona desde la tabla principal de padrón vehicular y direcciona a un formulario (figura 4.26) donde es posible modificar sus campos.

Figura 4.26. Formulario para modificar un vehículo. Autoría propia.

Los detalles de cada vehículo se muestran en una ventana modal (figura 4.27).

Figura 4.27. Modal de detalles de un vehículo. Autoría propia.

4.1.2.8 Resguardos.

Los resguardos (figura 4.28), son las comisiones a las que asisten los choferes de la Secretaría, donde tienen a su disposición vehículos que quedan bajo su responsabilidad.

Dentro de las acciones se encuentran las siguientes funciones:

- Editar: Abre un formulario para modificar un registro (figura 4.30).
- Eliminar: Borra un resguardo del sistema.
- Ver detalles: Muestra todos los campos de la tabla de resguardos (figura 4.31), que no es posible visualizar en la página principal.
- Mantenimiento. Abre un formulario para agregar un nuevo mantenimiento.
- Combustibles. Envía al usuario del sistema a un formulario, para añadir una carga de combustible, con los datos del resguardo seleccionado.
- Formato: Descarga en formato PDF los formatos Resguardo, Asignación y Liberación del Vehículo (ver anexo 1), Carta resguardo y Responsiva de Vehículos (ver anexo 2) y Resguardo (ver anexo 3)

Figura 4.28. Vista de los resguardos registrados. Autoría propia.

El botón “agregar” abre un formulario (figura 4.29) para añadir los datos de una comisión.

The screenshot shows the 'AGREGAR RESGUARDO' (Add Insurance) form. On the left is a navigation menu with options: CATÁLOGOS, PADRÓN VEHICULAR, RESGUARDOS, CONTROL DE COMBUSTIBLE, CONTROL DE SERV. DE MANTENIMIENTO, LICENCIAS, PÓLIZAS DE SEGURO, and FACTURAS. The main form area is titled 'AGREGAR RESGUARDO' and contains the following fields:

- NO. VEHICULO: Text input field.
- PLACA *: Dropdown menu with "-- ELIGE --" selected.
- ESTATUS *: Dropdown menu with "-- ELIGE --" selected.
- EJERCICIO *: Text input field.
- DESTINO: Text input field.

Below these fields is a section titled 'UBICACION Y RESPONSABLE LOCAL' (Location and Local Responsible) with the following fields:

- PERSONAL *: Dropdown menu with "-- ELIGE --" selected.
- UP *: Text input field.
- DIRECCION *: Text input field.
- DEPARTAMENTO *: Text input field.

Figura 4.29. Formulario para añadir un resguardo. Autoría propia.

La ventana para editar resguardos (figura 4.30), tiene la capacidad de mostrar los datos ya guardados de una comisión y permite que estos sean modificados.

The screenshot shows the 'EDITAR RESGUARDO' (Edit Insurance) form. The navigation menu is the same as in Figure 4.29. The main form area is titled 'EDITAR RESGUARDO' and contains the following pre-filled fields:

- NO. VEHÍCULO *: 3
- PLACA *: PLHJ-5
- ESTATUS *: ACTIVO
- EJERCICIO *: 2021
- DESTINO: CDMX

Below these fields is a section titled 'UBICACION Y RESPONSABLE LOCAL' (Location and Local Responsible) with the following pre-filled fields:

- PERSONAL *: MARIA ISABEL ALEJO AQUINO
- UP: UNIDAD ADMINISTRATIVA
- DIRECCIÓN: UNIDAD ADMINISTRATIVA
- DEPARTAMENTO: DEPARTAMENTO DE TECNOLOGIAS DE LA INFORMACION

Figura 4.30. Formulario para editar un resguardo. Autoría propia.

En la vista de los resguardos registrados (figura 4.28) no es posible mostrar todos los campos de las comisiones, es por ello que hay un modal (figura 4.31) que permite que dar a conocer los detalles de los resguardos.

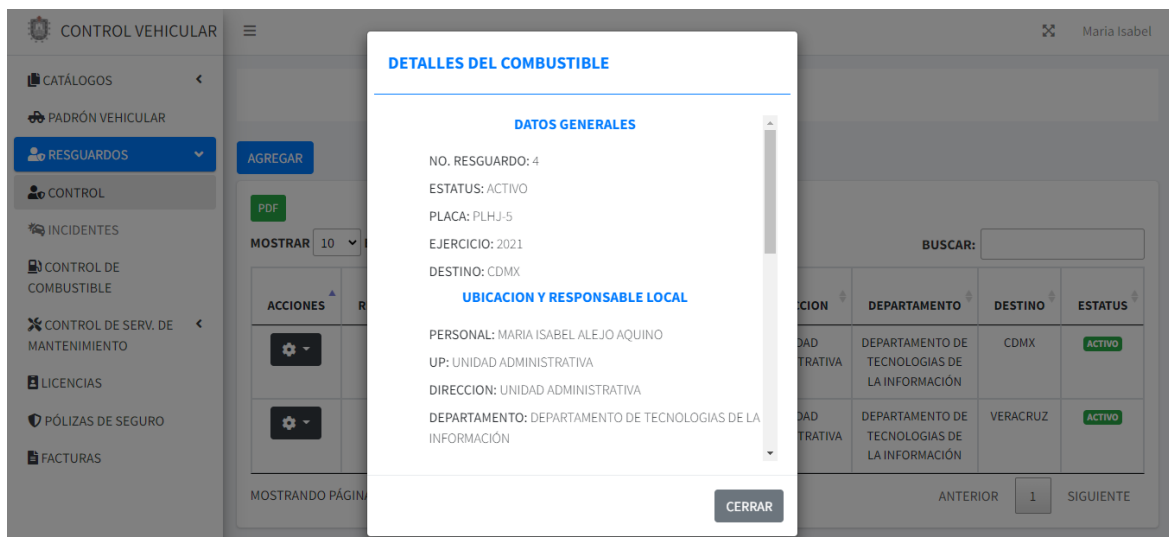


Figura 4.31. Modal de detalles de un resguardo. Autoría propia.

4.1.2.9 Control de combustible.

La vista principal (figura 4.32) de este módulo lista todas las cargas de combustible de los vehículos.

Las acciones de cada combustible son:

- Editar: Abre una nueva ventana (figura 4.34), donde se visualiza la carga seleccionada para ser editada en alguno de sus atributos.
- Eliminar: Borra una carga de combustible.
- Ver detalles: Abre una ventana modal (figura 4.35), donde se cargan todas las características de la compra del combustible.

CONTROL VEHICULAR

Maria Isabel

CONTROL DE COMBUSTIBLE

Agregar nuevo

PDF

MOSTRAR 10 ENTRADAS

BUSCAR:

ACCIONES	PLACA ACTUAL	FECHA CAPTURA	FECHA FACTURA	NO. FACTURA	TIPO COMBUSTIBLE	LITROS	IMPORTE	TIPO DE COMPRA
	PLHJ-5	2022-01-27	2022-01-27	ELCE-000913		20	200,2	EFFECTIVO
		2022-01-27	2022-01-20	ELCE-000913	MAGNA	50	300	EFFECTIVO
		2022-01-27	2022-01-28	ELCE-000913	MAGNA	70	300	VALE

MOSTRANDO PÁGINA 1 DE 1

ANTERIOR 1 SIGUIENTE

Figura 4.32. Vista de las cargas de combustible registradas. Autoría propia.

El botón “agregar” abre una nueva vista (figura 4.33), para adicionar un nuevo objeto.

CONTROL VEHICULAR

Maria Isabel

AGREGAR COMBUSTIBLE

DATOS GENERALES

NO. RESGUARDO * PLACA * FECHA DE CAPTURA * DESTINO

EJERCICIO * RESPONSABLE * DEPARTAMENTO *

DETALLE

TIPO DE COMBUSTIBLE * LITROS * LITROS ANTERIORES OTROS

CANTIDAD IMPORTE TOTAL * NO. FACTURA * ARCHIVO

Figura 4.33. Formulario para agregar una compra de combustible. Autoría propia.

Para editar una carga de combustible es necesario abrir la ventana “editar combustible” (figura 4.34) a través del botón editar.

Figura 4.34. Formulario para editar un registro. Autoría propia.

Los detalles del combustible se muestran en una ventana flotante (figura 4.35) dentro de la pestaña de control de combustible.

Figura 4.35. Modal de detalles de una carga de combustible. Autoría propia.

4.1.2.10 Control de serv. de mantenimiento.

Los servicios realizados a los vehículos se muestran en la ventana de mantenimientos (figura 4.36).

Las acciones para los mantenimientos son:

- Editar: Abre una ventana para modificar un servicio (figura 4.38).

- Eliminar: Borra un servicio realizado a un vehículo.
- Ver detalles: Muestra los pormenores de un registro en una ventana emergente (figura 4.39).

CONTROL VEHICULAR

MARIA ISABEL

MANTENIMIENTOS

Agregar nuevo

PDF

Mostrar 10 ENTRADAS

Buscar:

ACCIONES	PLACA	FECHA	CONCEPTO	KM	NO. FACTURA	IMPORTE	PROVEEDOR	ESTATUS
	PLHJ-5	2022-02-10	SERVICIO DE KILOMETRAJE	1000	ELCE-000913	\$ 2500	LLANTA FACIL/ANTONIO M. LOPEZ LUNAGOMEZ	EN SERVICIO

ANterior 1 SIGUIENTE

Copyright © 2021
Secretaría de Infraestructura y Obras Públicas

Figura 4.36. Vista principal de los mantenimientos realizados. Autoría propia.

Para agregar un nuevo servicio de mantenimiento se tiene que hacer clic en el botón “agregar nuevo” donde abre un formulario con los campos requeridos.

CONTROL VEHICULAR

MARIA ISABEL

AGREGAR MANTENIMIENTO

DATOS GENERALES

NO. RESGUARDO * NO. VEHÍCULO * PLACA * PERSONAL *

KM * CONCEPTO OBSERVACIONES

EJERCICIO ESTATUS *

FACTURA

NO. FACTURA * ARCHIVO FECHA FACTURA

Figura 4.37. Formulario para agregar un servicio. Autoría propia.

La ventana “editar mantenimiento” (figura 4.38) permite modificar campos de algún servicio realizado a un vehículo.

Figura 4.38. Formulario para editar un mantenimiento. Autoría propia.

Los detalles de cada servicio de mantenimiento se muestran en una ventana modal (figura 4.39).

Figura 4.39. Modal de detalles de un servicio realizado. Autoría propia.

En el momento preciso de que un vehículo requiera un servicio se muestra en la página de “Mantenimientos pendientes” (figura 4.40).

MANTENIMIENTOS PENDIENTES

MOSTRAR 10 ENTRADAS BUSCAR:

NO. VEHICULO	PLACA	KM ACTUAL	MARCA	LÍNEA	TIPO	ESTADO
Nothing found - sorry						

SIN REGISTROS DISPONIBLES ANTERIOR SIGUIENTE

Copyright © 2021
Secretaría de Infraestructura y Obras Públicas

Figura 4.40. Lista de los mantenimientos pendientes. Autoría propia.

4.1.2.11 Licencias.

Al agregar un nuevo personal, se debe poner como dato obligatorio, la fecha de expiración de su licencia, con esta información se tiene un control del vencimiento de las mismas. En la pestaña “Licencias” se visualizan los permisos de conducir que están vencidos. (figura 4.41).

LICENCIAS VENCIDAS

MOSTRAR 10 ENTRADAS BUSCAR:

NOMBRE	DEPARTAMENTO	NO. LICENCIA	VENCIMIENTO DE LICENCIA	ESTADO
JUAN GOMEZ	SECRETARIO PRIVADO	81S876978	2022-01-28	VENCIDA
MARIA ISABEL ALEJO AQUINO	DEPARTAMENTO DE TECNOLOGIAS DE LA INFORMACIÓN	01N450316	2022-01-26	VENCIDA

MOSTRANDO PÁGINA 1 DE 1 ANTERIOR 1 SIGUIENTE

Copyright © 2021
Secretaría de Infraestructura y Obras Públicas

Figura 4.41. Listado de las licencias vencidas. Autoría propia.

4.1.2.12 Pólizas de seguro.

La mayor parte de los vehículos de la Secretaría, cuenta con una póliza de seguros por cualquier accidente que pudiera ocurrir en las comisiones. Para ello es preciso saber el vencimiento de las mismas, en la ventana de “Pólizas de seguro” (figura 4.42), se tiene un listado de las pólizas que deben renovarse.

The screenshot displays the 'VENCIMIENTO DE PÓLIZAS DE SEGURO' (Insurance Policy Expiry) section of the 'CONTROL VEHICULAR' system. The interface includes a sidebar with navigation options like 'CATÁLOGOS', 'PADRÓN VEHICULAR', 'RESGUARDOS', 'CONTROL DE COMBUSTIBLE', 'CONTROL DE SERV. DE MANTENIMIENTO', 'LICENCIAS', 'PÓLIZAS DE SEGURO', and 'FACTURAS'. The main content area shows a table with the following data:

NO. ECONÓMICO	PLACA ACTUAL	T. CIRCULACIÓN	EMPRESA DE SEGURO	NO. POLIZA DE SEG.	FECHA DE EMISIÓN	FECHA DE VENCIMIENTO	FECHA DE PAGO	COSTO	ESTATUS
501	PLHJ-5	500			2021-11-01	2021-12-30	2021-12-08		VENCIDA

Below the table, it indicates 'MOSTRANDO PÁGINA 1 DE 1' and navigation buttons for 'ANTERIOR', '1', and 'SIGUIENTE'. The footer contains 'Copyright © 2021 Secretaría de Infraestructura y Obras Públicas'.

Figura 4.42. Vista de las pólizas expiradas. Autoría propia.

4.1.2.13 Facturas.

Los archivos de las facturas y otros detalles se muestran en la pestaña de “Facturas” (figura 4.43) y están divididas de acuerdo a la actividad para las que se hayan generado, las cuales son combustibles y mantenimientos.

En la misma página se listan ambos tipos de facturas. Es posible adjuntar el archivo de la factura, es por ello que hay un ícono en forma de carpeta para visualizar la factura electrónica o en otro caso un comprobante de compra.

PLACA	NO. RESGUARDO	NO. FACTURA	FECHA	IMPORTE TOTAL	ORIGEN DE RECURSOS	FORMA DE PAGO	DESC. PAGO	CIUDAD	ESTATUS	ARCHIVO
PLHJ-5	5	ELCE-000913	2022-02-10	2500	2% NOMINA	CHEQUE	SERVICIO 9000 KM	XALAPA	PAGADO	

Figura 4.43. Listado de las facturas de los mantenimientos y cargas de combustible.

Autoría propia.

4.1.3 Nivel controlador

En este nivel se muestra únicamente el controlador del módulo Departamento, todos los métodos empleados en este *controller* son reutilizados para las tablas restantes, a diferencia de otras funciones que son explicadas más adelante.

En la figura 4.44 se visualiza el método *index* con el cual se hace una consulta de los elementos de un módulo almacenados en la base de datos y que posteriormente son mostrados en una vista.

```
public function index()
```

```
{
    ... $departamento = Departamento::all();
    ... return view('admin.departamentos.mostrarDepto', compact('departamento'));
}
```

Figura 4.44. Método *index* del controlador Departamento. Autoría propia.

En la figura 4.45 se muestran los métodos *create* y *store* que se encargan de abrir la ventana del formulario de registro y guardan los datos capturados.

```

public function create()
{
    $sup = Direccion::all();
    $direcciones = Direccion::all();
    return view('admin.departamentos.agregarDepto', compact('up', 'direcciones'));
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $departamento = new Departamento();
    $departamento->cve_depto = $request->cve_depto;
    $departamento->cveup = $request->cveup;
    $departamento->cve_dir = $request->cve_dir;
    $departamento->desc_departamento = $request->desc_departamento;
    $departamento->save();
    return redirect()->action([DepartamentoController::class, 'index']);
}

```

Figura 4.45. Método *create* y *store* del controlador Departamento. Autoría propia.

En la figura 4.46 se aprecia el código de los métodos *edit* y *update*. El primero se encarga de enviar a la vista de edición los datos del elemento que se va a editar y el segundo guarda y actualiza la información que se ha capturado en las cajas de texto del formulario para posteriormente volver a la vista principal del módulo.


```

public function edit($cve_depto)
{
    $supe = Direccion::all();
    $direcciones = Direccion::all();
    $departamento = Departamento::find($cve_depto);
    return view('admin.departamentos.editarDepto', compact('upe', 'direcciones'), [
        'departamento' => $departamento
    ]);
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $cve_depto
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $cve_depto)
{
    $departamento = Departamento::findOrFail($cve_depto);
    $departamento->cveup = $request->cveup;
    $departamento->cve_dir = $request->cve_dir;
    $departamento->desc_departamento = $request->desc_departamento;
    $departamento->save();
    return redirect()->action([DepartamentoController::class, 'index']);
}

```

Figura 4.46. Método *edit* y *update* del controlador Departamento. Autoría propia.

A continuación, la figura 4.47 muestra el método para eliminar un departamento, el cual recibe todos los campos del elemento seleccionado para posteriormente borrarlo de forma definitiva de la base de datos.

```

public function destroy(Departamento $departamento)
{
    $departamento->delete();
    return redirect()->action([DepartamentoController::class, 'index']);
}

```

Figura 4.47. Método *destroy* del controlador Departamento. Autoría propia.

En la figura 4.48 se muestra el método *index* que es aplicado para los módulos Combustible, Mantenimiento y Resguardo, dentro del código, antes de mostrar la vista principal se actualizan los vehículos con los datos del último elemento almacenado, como, por ejemplo, para capturar una compra de combustible se escribe el kilometraje

actual del vehículo que a su vez debe ser actualizado a la unidad en resguardo cada vez que se realiza una carga.

```
public function index(Request $request)
{
    $last_placa = Combustible::latest('id')->value('placa_act');
    $last_lts = Combustible::latest('id')->value('lts');
    $last_ant = Combustible::latest('id')->value('lts_ant');
    $last_km_ant = Combustible::latest('id')->value('km_ant');
    $last_km_act = Combustible::latest('id')->value('km_actual');

    $affected = DB::table('veh_resguardo')
        ->join('veh_padron_vehicular', 'veh_resguardo.id_veh', '=', 'veh_padron_vehicular.id_veh')
        ->join('veh_combustibles', 'veh_resguardo.id_resguardo', '=', 'veh_combustibles.id_resguardo')
        ->where('veh_padron_vehicular.placa_act', $last_placa)
        ->update(['veh_padron_vehicular.lts_actual' => $last_lts]);

    $combustible = Combustible::all();
    return view('admin.combustibles.mostrarCombustible', [
        'combustible' => $combustible
    ]);

    return view('combustible', compact('combustible'));
}
```

Figura 4.48. Método *index* del controlador Combustible. Autoría propia.

El método *fechasVencimiento* (4.49) permite a la vista mostrar las licencias del personal que están vencidas a la fecha actual.

```
public function fechasVencimiento()
{
    $personal = DB::table('veh_personal')
        ->join('veh_catdepto', 'veh_personal.cve_depto', '=', 'veh_catdepto.cve_depto')
        ->select('veh_catdepto.*', 'veh_personal.*')
        ->where('vencimiento_licencia', '<=', Carbon::now()->format('Y-m-d'))->get();
    return view('admin.vencimientos.licencias', ['personal' => $personal]);
}
```

Figura 4.49. Método *fechasVencimiento* del controlador Personal. Autoría propia.

La función *vencimientoPoliza*(figura 4.50) se utiliza en el controlador de Vehículo para mostrar en una tabla los automóviles cuya póliza de seguro este expirada.

```
public function vencimientoPoliza()
{
    $polizas = DB::table('veh_padron_vehicular')
        ->where('vigencia_seg', '<=', Carbon::now()->format('Y-m-d'))->get();
    return view('admin.vencimientos.poliza_seguro', ['polizas' => $polizas]);
}
```

Figura 4.50. Método *vencimientoPoliza* del controlador Vehículo. Autoría propia.

En la figura 4.51 se visualizan 3 métodos en el controlador de Resguardo, que son utilizados para generar los documentos PDF:

- Formato para resguardo, asignación y liberación del vehículo.
- Carta Resguardo y Responsiva de Vehículos.
- Resguardo

```

public function generarPdf($id_resguardo)
{
    ... $resguardo = Resguardo::find($id_resguardo);
    ... $dia = Carbon::now('America/Mexico_City')->format('d');
    ... $mes = Carbon::parse()->isoFormat('MMMM');
    ... $year = Carbon::now('America/Mexico_City')->isoFormat('YYYY');
    ... $pdf = PDF::loadView('admin.reportes.responsiva', [
    ... |... 'resguardo' => $resguardo
    ... ], compact('dia', 'mes', 'year'));
    ... $pdf->setPaper('landscape');
    ... return $pdf->download('responsiva.pdf');
} ...
public function imprimirResguardo($id_resguardo){
    ... $dia = Carbon::now('America/Mexico_City')->format('d');
    ... $mes = Carbon::parse()->isoFormat('MMMM');
    ... $year = Carbon::now('America/Mexico_City')->isoFormat('YYYY');
    ... $resguardo = Resguardo::find($id_resguardo);
    ... $pdf = PDF::loadView('admin.reportes.resguardo', ['resguardo' => $resguardo], compact('dia', 'mes', 'year'));
    ... return $pdf->download('resguardo.pdf');
} ...
public function imprimirResguardoCarta($id_resguardo)
{
    ...
    ... $dia = Carbon::now('America/Mexico_City')->format('d');
    ... $mes = Carbon::parse()->isoFormat('MMMM');
    ... $year = Carbon::now('America/Mexico_City')->isoFormat('YYYY');
    ... $resguardo = Resguardo::find($id_resguardo);
    ... $pdf = PDF::loadView('admin.reportes.resguardoHojas', [
    ... |... 'resguardo' => $resguardo
    ... ], compact('dia', 'mes', 'year'));
    ... return $pdf->download('resguardo-pdf.pdf');
}

```

Figura 4.51. Métodos para generar formatos PDF del controlador Resguardo. Autoría

Capítulo 5 Conclusiones

5.1 Conclusiones y recomendaciones

5.1.1 *Conclusión.*

En este reporte se presenta el desarrollo de un sistema web para el control de vehículos de la Secretaría de Infraestructura y Obras Públicas, con el objetivo de sustituir al software antecesor con ayuda de tecnologías actuales, la aplicación web es capaz de adicionar mejoras, una de ellas es generar formatos .pdf que anteriormente se editaban con la paquetería de Office. La aplicación web implementa una arquitectura Modelo Vista Controlador. La primer capa administra todos los datos almacenados en el SGBD. La segunda capa muestra de manera gráfica información con la que el usuario puede interactuar. La tercer capa controla las acciones y eventos del usuario. Un aspecto considerado para el desarrollo del sistema fueron los requerimientos, con ayuda de diagramas se representaron las actividades que permiten a la aplicación web cumplir con sus funciones. Los módulos del sistema se diseñaron con ayuda del framework Laravel 8.0 y la plantilla AdminLTE. Posteriormente se utilizó la librería DomPDF para generar los formatos .pdf con las características proporcionadas por la secretaría. Para la implementación del sistema se utilizaron dos programas: VNC Viewer y PuTTY. Una vez instalados se realizaron las pruebas necesarias y se comprobó que cada módulo funcionaba correctamente.

5.1.2 *Recomendaciones.*

Es aconsejable, brindar actualizaciones al sistema con el objetivo de dar una mayor seguridad a la información, además de ir añadiendo mejoras cada vez que es lanzada una nueva versión del framework con el que fue codificado. Por otro lado, es necesario realizar manuales del funcionamiento del programa, debido a que en ocasiones los usuarios finales no conocen todas las características de su sistema, por lo cual el uso es incorrecto. Al guardar grandes cantidades de información, se debe proporcionar mantenimiento a las tablas de la base de datos y crear copias de seguridad.

Capítulo 6 Referencias bibliográficas

6.1 Referencias bibliográficas

- Addappto. (2015). *Addappto*. Obtenido de Addappto: <http://www.addappto.com/que-es-un-sistema-web/>
- AdminLTE. (2021). *AdminLTE*. Obtenido de AdminLTE: <https://adminlte.io/themes/v3/>
- Adobe. (2021). *Adobe*. Obtenido de Adobe: <https://helpx.adobe.com/es/dreamweaver/using/web-applications.html>
- Anexia Tecnologías. (2020). *Anexia Tecnologías*. Obtenido de Anexia Tecnologías: <https://tecnologias.anexia.es/blog/que-son-las-aplicaciones-web-y-las-paginas-web-dinamicas#.YdYVjGjMLIU>
- Arjol, O. G. (2014). *Oscargascon*. Obtenido de Oscargascon: <https://oscargascon.es/uso-de-dompdf-para-generar-pdf-con-php-html-y-css/>
- Axarnet. (s.f.). *Axarnet*. Obtenido de Axarnet: <https://axarnet.es/blog/bootstrap>
- Berzal, F., Cortijo, F. J., & Carlos, J. (s.f.). *Elvex*. Obtenido de Elvex: <https://elvex.ugr.es/decsai/csharp/pdf/web/4-web-practice.pdf>
- Dataprix. (2009). *Dataprix*. Obtenido de Dataprix: <https://www.dataprix.com/es/bases-datos-master-software-libre-uoc/bases-datos-mysql>
- Díaz, Y., & Fernández, Y. (2012). *Telemática*. Obtenido de Telemática: <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15>
- García, F. (2015). *Arsys*. Obtenido de Arsys: <https://www.arsys.es/blog/programacion/que-es-laravel>
- Gobierno del estado. (s.f.). *Secretaría de Infraestructura y Obras Públicas*. Obtenido de Secretaría de Infraestructura y Obras Públicas: <http://www.veracruz.gob.mx/infraestructura/wp-content/uploads/sites/17/2021/03/ESTRUCTURA-ORG%C3%81NICA-SIOP-2020.pdf>
- Gobierno del Estado. (s.f.). *Veracruz Gobierno del Estado*. Obtenido de Veracruz Gobierno del Estado: <http://www.veracruz.gob.mx/infraestructura/transparencia/quienes-somos/>
- Google. (s.f.). *Google Maps*. Obtenido de Google Maps: <https://www.google.com.mx/maps/place/SIOP/@19.528683,-96.9000703,16.83z/data=!4m5!3m4!1s0x0:0x7758082952d75f9f!8m2!3d19.5289951!4d-96.8984983>

- Kinsta. (2022). *Kinsta*. Obtenido de Kinsta: <https://kinsta.com/es/base-de-conocimiento/que-es-javascript/>
- Lucidchart. (s.f.). *Lucidchart*. Obtenido de Lucidchart: <https://www.lucidchart.com/blog/es/pros-y-contras-de-la-metodologia-de-cascada>
- Risso, I. (2022). *Crehana*. Obtenido de Crehana: <https://www.crehana.com/blog/desarrollo-web/modelo-en-cascada/>
- Rodríguez, M. (2021). *Cursos Gis*. Obtenido de Cursos Gis: <https://www.cursosgis.com/como-instalar-y-empezar-a-usar-visual-studio-code-en-5-pasos/>
- Ryte Wiki. (s.f.). *Ryte Wiki*. Obtenido de Ryte Wiki: <https://es.ryte.com/wiki/PHP#:~:text=El%20PHP%20es%20un%20lenguaje,un%20lenguaje%20f%C3%A1cil%20de%20aprender.>
- Santiago González, L., Huerta Mendoza, J., Mendoza Luna, Y., Rodríguez Rodríguez, A., Vargas Requena, D., Martínez Rodríguez, J., & Malacara Navejar, J. (2021). *Tecnología educativa revista*. Obtenido de Tecnología educativa revista: <https://www.terc.mx/index.php/terc/article/view/187>
- Smania, D. (2021). *GitHub*. Obtenido de GitHub: <https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Usage>

Capítulo 7 Anexos

7.1 Anexos

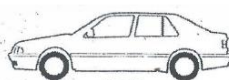
Anexo 1. Formato para resguardo, asignación y liberación del vehículo.

DEPARTAMENTO DE RECURSOS MATERIALES Y SERVICIOS GENERALES
OFICINA DE CONTROL VEHICULAR
RESGUARDO, ASIGNACIÓN Y LIBERACIÓN DEL VEHÍCULO
XALAPA, VER., A 06 DE FEBRERO DE 2022

1.- ASIGNACIÓN ()				2.- LIBERACIÓN ()				HORA DE ENTREGA:															
DATOS DEL RESGUARDATARIO																							
NOMBRE				CARGO				ADSCRIPCIÓN															
MARIA ISABEL ALEJO AQUINO								DEPARTAMENTO DE TECNOLOGIAS DE LA INFORMACIÓN															
PERIODO DE ASIGNACIÓN								COMISIÓN A REALIZAR															
ASIGNACION:				TEMPORAL.- ()				PERMANENTE.- ()															
DATOS DEL VEHÍCULO																							
N° ECON.		N° PLACA		MARCA		TIPO		N° PÓLIZA SEGURO		CILINDRAJE		COLOR											
501		PLHJ-5		FIAT		COLORADO				5		NEGRO											
KM ACTUAL			N° MOTOR			N° T. CIRCULA			N° SERIE			MODELO		GASOLINA									
1.-ASIGNADO			2.-LIBERADO			52WCV10338			500			1HGBH41JXMNI09186			2010		E 1/4 1/2 3/4 F () () () ()						
CONCEPTO		B	R	M	OBSERVACIONES			CONCEPTO		B	R	M	OBSERVACIONES			CONCEPTO		B	R	M	OBSERVACIONES		
PLACAS								CRISTAL MEDALLON								ENCENDEDOR							
PINTURA								TAPA CAJUELA								ESPEJO RETROVISOR							
COFRE								SALPICADERA TRASERA								VICERAS							
LIMPIADORES								DEFENSA TRASERA								MANIJAS INTERIOR							
PARRILLA								CALAVERAS								ALFOMBRA							
TOPE								MOTOR								CODERAS							
DEFENSA DELANTERA								TAPON ACEITE								CINTURON DE SEGURIDAD							
UNIDAD DE LUZ								BAYONETA								PLAFON							
BICELES								ACUMULADOR								ASIENTOS							
CUARTOS DELANTEROS								MARCHA								VESTIDURA							
CRISTAL PARABRISAS								GENERADOR								RADIO							
CRISTALES LATERALES								BOBINA								BOCINAS							
PUERTAS								DISTRIBUIDOR								CLIMA							
MANIJAS EXTERIORES								PURIFICADOR								TAPON DE RUEDA							
MOLDURAS								CARBURADOR								REFLEJANTES							
ESPEJOS								RADIADOR Y TAPÓN								GATO							
TAPON GASOLINA								TABLERO MARCADOR								LLAVE DE BIRLOS							
ANTENA								CAJUELA DE GUANTES								CABLES DE CORRIENTE							
																OTROS							
LLANTAS DELANTERAS				LLANTAS TRASERAS				LLANTA DE REFACCION				OBSERVACIONES											
DERECHA % V				DERECHA % V				SI % V															
IZQUIERDA % V				IZQUIERDA % V				NO % V															
MARCAS:				MARCAS:				MARCAS:															
SERIE:				SERIE:				SERIE:															

Debido a que las multas tienen un plazo muy amplio de llegar, este compromiso de pago lo hago bajo protesta de decir verdad, que entregaré a la Secretaría de Infraestructura y Obras Públicas en el momento que me presente el informe correspondiente de la infracción, así como entregar la unidad conforme la recibo de acuerdo a la solicitud y/o requerimiento número de ____ fecha ____

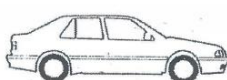
ASIGNADO:	
ENTREGA	RECIBE
LIBERADO:	
ENTREGA	RECIBE



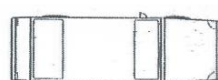
PILOTO



FRENTE



COPILOTO



ARRIBA



ATRÁS

○ Raspón ✕ Golpe ✓ Estrellado

Anexo 2. Carta Resguardo y Responsiva de Vehículos.



UNIDAD ADMINISTRATIVA

DEPARTAMENTO DE RECURSOS MATERIALES Y SERVICIOS GENERALES

**Asunto: CARTA RESGUARDO
Y RESPONSA DE VEHÍCULOS RMCVR-1028/2021**

**SECRETARÍA DE INFRAESTRUCTURA Y OBRAS PÚBLICAS
DEL GOBIERNO DEL ESTADO DE VERACRUZ DE IGNACIO DE LA LLAVE
P R E S E N T E**

El (la) que suscribe C.MARIA ISABEL ALEJO AQUINO adscrito (a) a DEPARTAMENTO DE TECNOLOGIAS DE LA INFORMACIÓN de la Secretaría de Infraestructura y Obras Públicas, con número de personal, 10 credencial de elector _____ y licencia de manejo 01N450316, recibo en este acto el vehículo marca 3 modelo 3, con placas de circulación, PLHJ-5 No. económico 3 propiedad Y/O/ resguardo de esta Dependencia, para utilizarlo en la comisión encomendada el (los) día(s) _____ del _____ del mes de _____ del 2021.

Lo anterior con fundamento en el artículo 5 de la Ley de Responsabilidades Administrativas para el Estado de Veracruz de Ignacio de la Llave; artículos 88 fracción II y 93 de la Ley de Adquisiciones, Arrendamientos, Administración y Enajenación de Bienes Muebles del Estado de Veracruz de Ignacio de la Llave; artículos 45 fracción 1,46,56,65 fracción I y 11,68,72,77 fracción XII de la Ley de Tránsito y Seguridad vial para el Estado de Veracruz de Ignacio de la Llave; artículo 19 fracción XX del Reglamento Interno de la Secretaría de Infraestructuras y Obras Públicas del Estado de Veracruz de Ignacio de la Llave; por lo que manifiesto estar enterado de las responsabilidades, reconociendo que a partir de este momento, queda bajo mi resguardo el vehículo señalado en el párrafo anterior, así como cubrir por cuenta propia los daños materiales que sufra la unidad, ocasionados por descuidos, falta de pericia, negligencia o irresponsabilidad en la forma de conducir, así como en el caso de no reportar oportunamente hechos que acontezcan a la oficina de control vehicular perteneciente al Departamento de Recursos Materiales de la unidad administrativa de esta Secretaría.

En este orden de ideas, el responsable del vehículo será quien se hará cargo por cualquier tipo de multa y/o infracción que tuviera como origen por algún tipo de violación al Reglamento de la Ley de Tránsito y Transporte para el Estado de Veracruz de Ignacio de la Llave, así como al Reglamento de Tránsito en Carreteras y Puentes de Jurisdicción Federal, o cualquier Ley aplicable en el rubro. Una vez leído la presente responsiva, me comprometo a utilizar la unidad únicamente para la comisión encomendada y a devolverla a la Oficina de Control Vehicular, una vez concluida la comisión, en las mismas condiciones en la que me fue entregada.

ATENTAMENTE

Xalapa, Ver., 06 de febrero del 2022

C. MARIA ISABEL ALEJO AQUINO

L.E Héctor Martínez Oliver.- Titular del Departamento de Recursos Materiales y Servicios Generales.- Para su conocimiento.
L.C. Juan Antonio Andrade Sánchez.- Jefe de la Oficina de Control Vehicular.- Mismo fin.
Archivo/Minutario

Coronel Pablo Frutis 4
CP 91190 Xalapa, Veracruz.
Tel. 01 228 841 61 20

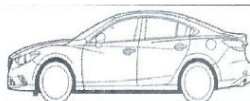
Anexo 3. Resguardo



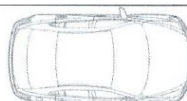
RESGUARDO DE VEHICULOS PROPIEDAD DEL GOBIERNO DEL ESTADO DE VERACRUZ DE IGNACIO DE LA LLAVE											
DEPENDENCIA/ENTIDAD						PLACAS DE CIRCULACION ANTERIOR		PLACAS DE CIRCULACION ACTUAL			
SECRETARÍA DE INFRAESTRUCTURA Y OBRAS PÚBLICAS						DEF-65		PLHJ-5			
DATOS DEL VEHÍCULO											
MARCA		TIPO DE VEHÍCULO		MOD.	CIL.	COLOR		PÓLIZA DE SEGURO N°			
FIAT		COLORADO		2010	5	NEGRO					
NÚMERO DE MOTOR	NÚMERO DE SERIE	NO. TARJ. CIRC.	KILOMETRAJE	ÚLTIMA VERIFIC.	PRÓXIMA VERIFIC.	FECHA ÚLTIMO MANTENIMIENTO		FECHA PRÓXIMO MANTENIMIENTO			
52WCV10338	1HGBH41JXMN109186	500	50000								
REVISIÓN FÍSICA (B=BUENO, R=REGULAR, M=MALO, N/A=NO APLICA)											
CONCEPTO	B	R	M	N/A	OBSERVACIONES						
PLACA DELANTERA					MANIJAS DE PUERTAS						
PARRILLA					BAYONETAS (MOTOR/TRANSM.)						
COFRE					RADIADOR						
SALPICADERA DELANTERA					MARCHA						
UNIDAD DE LUZ					MARCADORES DE TABLERO						
ESPEJOS					MOTOR DE LIMPIADORES						
SALPICADERA TRASERA					CLUTCH						
TAPA CAJUELA					ACUMULADOR						
PLACA TRASERA					BOBINA						
DEFENSAS					ALTERNADOR						
LLANTAS					DISTRIBUIDOR						
RINES					CRISTALES						
FRENOS					PUERTAS						
AMORTIGUADORE											
ACCESORIOS (B=BUENO, R=REGULAR, M=MALO, N/A= NO APLICA)											
ALARMA	VISERAS	CINTURON DE SEGURIDAD	ANTENA	ELEVADORES ELÉCTRICOS	TAPONES DE RUEDAS	LLANTA DE REFACCIÓN Y % DE VIDA	GATO				
ASIENTOS	TAPETE	RADIO	ENCENDEDOR	GUANTERA	TUERCAS DE SEGURIDAD		EXTINTOR				
TECHO INTERIOR	ALFOMBRA	BOCNAS	CLMA	CODERAS	TRIANGULOS REFLEJANTES	LLAVE DE BIRLOS	CABLES DE CORRIENTE				
NIVELES DE FLUIDOS											
	DEPÓSITO DE FLUIDOS				OBSERVACIONES		DEPÓSITO DE FLUIDOS				OBSERVACIONES
COMBUSTIBLE	1/4	1/2	3/4	LL		FRENOS	1/4	1/2	3/4	LL	
ACEITE	1/4	1/2	3/4	LL		DIRECCIÓN HIDRÁULICA	1/4	1/2	3/4	LL	
REFRIGERANTE	1/4	1/2	3/4	LL		TRANSMISIÓN	1/4	1/2	3/4	LL	
OBSERVACIONES:											
(LLENADO MANUAL CON TINTA AZUL) INSPECCIÓN DE LAS CONDICIONES FÍSICAS DEL VEHICULO (LLENADO MANUAL CON TINTA AZUL)											



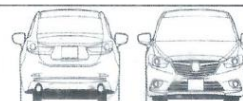
LADO DERECHO



LADO IZQUIERDO



TOLDO, COFRE Y CAJUELA



FRENTE Y PARTE TRASERA

AL FIRMAR ESTE RESGUARDO ME OBLIGO A:

- UTILIZAR LA UNIDAD EXCLUSIVAMENTE PARA ASUNTOS OFICIALES.
- ATENDER EL MANTENIMIENTO PREVENTIVO Y CORRECTIVO CON CARGO AL PRESUPUESTO DE LA DEPENDENCIA, ENTIDAD DE ADSCRIPCIÓN, DEPARTAMENTO U OFICINA QUE RECIBE LA UNIDAD EN EL PRESENTE RESGUARDO
- NO PERMITIR EL USO DEL VEHÍCULO A TERCERAS PERSONAS AJENAS AL SERVICIO PÚBLICO.
- CUBRIR POR CUENTA PROPIA LOS DAÑOS MATERIALES QUE SUFRA LA UNIDAD, OCASIONADOR POR DESCUIDO, NEGLIGENCIA O IRRESPONSABILIDAD DEL SERVIDOR PÚBLICO, INCLUSIVE CUANDO LA RESPONSABILIDAD NO LE SEA INPUTABLE, COMO EN EL CASO DONDE NO REPORTE LOS HECHOS CON OPORTUNIDAD A LA UNIDAD ADMINISTRATIVA DE SU ADSCRIPCIÓN.
- QUEDA PROHIBIDO CUBRIR CON CARGO AL PRESUPUESTO AUTORIZADO A LAS DEPENDENCIAS O ENTIDADES, CUALQUIER MULTA QUE TUVIERA COMO ORIGEN, VIOLACIONES AL REGLAMENTO DE LA LEY DE TRÁNSITO Y TRANSPORTE PARA EL ESTADO DE VERACRUZ DE IGNACIO DE LA LLAVE O RETRASOS EN EL PAGO DE IMPUESTOS, DERECHOS, OBLIGACIONES Y/O TRÁMITES ANTE LAS AUTORIDADES LOCALES Y FEDERALES, LAS UNIDADES ADMINISTRATIVAS REQUERIRÁN EL PAGO A LOS SERVIDORES PÚBLICOS RESPONSABLES.
- EN CASO DE ACCIDENTE Y/O ROBO FAVOR DE REPORTARLO INMEDIATAMENTE AL NÚMERO DE LA ASEGURADORA IMPRESO EN LA POLIZA DE SEGURO CORRESPONDIENTE ASÍ COMO A SU ÁREA ADMINISTRATIVA.
- CUANDO POR CUALQUIER CIRCUNSTANCIA DEJE DE ESTAR EL PRESENTE A CARGO DEL RESGUARDANTE SIGNADO EN LA PARTE INFERIOR DEL MISMO, SOLICITAR INMEDIATAMENTE SU CANCELACIÓN, DE NO SER ASÍ, EL OPERADOR ACEPTA LA RESPONSABILIDAD POR EL MAL USO O DETERIORO QUE OTROS PUDIERAN DAR AL VEHÍCULO.
- EN CASO DE INCUMPLIMIENTO A LO EXPRESAMENTE SEÑALADO, SE ACEPTA DE HECHO LA RESPONSABILIDAD JURÍDICA QUE DE ESTO RESULTARA PROCEDENTE.



FOTOGRAFÍAS DEL VEHICULO

FRENTE	PARTE TRASERA
LADO IZQUIERDO	LADO DERECHO
MOTOR	TABLERO/INTERIOR

CARGO, PUESTO O COMISIÓN, NOMBRE Y FIRMA DEL SERVIDOR PÚBLICO QUE RECIBE FÍSICAMENTE LA UNIDAD	Vo. Bo. TITULAR DEL ÁREA USUARIA	RESPONSABLES DEL DEPARTAMENTO DE CONTROL Y MANTENIMIENTO VEHICULAR DE SEFIPLAN	
LIC. JAME QUINO SOSA	DR. ARTURO GARCÍA MARQUEZ	ING. JUAN CARLOS LARA CASTILLO	ING. EDHER EXELENTÉ GONZÁLEZ
JEFE DEL DEPTO. DE RECS. MATS. Y SERV. GRALES.	JEFE DE LA UNIDAD ADMINISTRATIVA	JEFE DE LA OFICINA DE CONTROL VEHICULAR	JEFE DEL DEPARTAMENTO DE CONTROL Y MANTENIMIENTO VEHICULAR

XALAPA, VER., A 06 DE FEBRERO DE 2022

