



**UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**  
*La Universidad Católica de Loja*

**ÁREA TÉCNICA**

**TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y  
COMPUTACIÓN**

**Diseño e implementación del Sistema de Gestión Vehicular para el  
Gobierno Provincial de Loja**

**TRABAJO DE TITULACIÓN**

**AUTOR:** Guamán López, Andrea Silvana  
Ordoñez Ramón, Yonder Leonel

**DIRECTOR:** Guamán Coronel, Daniel Alejandro, Mgs.

**LOJA - ECUADOR**

**2016**



*Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>*

*Febrero, 2016*

## **APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN**

Magister.

Guamán Coronel, Daniel Alejandro.

**DOCENTE DE LA TITULACIÓN**

De mi consideración:

El presente trabajo de fin de titulación: “Diseño e Implementación del sistema de Gestión Vehicular para el Gobierno Provincial de Loja” realizado por Guamán López, Andrea Silvana y Ordoñez Ramón Yonder Leonel, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, febrero de 2016

f) .....

## DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Nosotros, Andrea Silvana Guamán López, Yonder Leonel Ordoñez Ramón, declaramos ser autores del presente trabajo de titulación: Diseño e implementación del sistema de Gestión Vehicular para el Gobierno Provincial de Loja, de la Titulación Ingeniero en Sistemas Informáticos y Computación, siendo Daniel Alejandro Guamán Coronel director del presente trabajo; y eximimos expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certificamos que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de nuestra exclusiva responsabilidad.

Adicionalmente declaramos conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

f. ....

Autor: Andrea Silvana Guamán López

Cédula: 1105603862

f. ....

Autor: Yonder Leonel Ordoñez Ramón

Cédula: 1105168767

## DEDICATORIA

A Dios por permitirme llegar en este momento tan especial en mi vida y haberme dado la salud para lograr mis objetivos, A mis padres, Hugo y Luz, por ser ejemplos de perseverancia y constancia, a mis hermanos, Mary, Diego y Sory, por estar siempre presentes. A mis amigos Andreita, Katy, Karina, Dianita, Yonder, Carlitos y Juan Pablo, que de una u otra forma han contribuido a cumplir con este objetivo. Y finalmente a todos aquellos que marcaron cada etapa en mi camino universitario brindando su apoyo y comprensión.

*Andrea Silvana*

La presente tesis se la dedico a mi familia que gracias a su apoyo logre concluir mi carrera. A mi padre por brindarme los recursos necesarios y estar a mi lado apoyándome y aconsejándome siempre. A mi madre por hacer de mí una mejor persona a través de sus consejos, enseñanzas y amor. A mi abuelita por brindarme su cariño y protección. A mis hermanos por estar siempre presentes, acompañándome.

A todo el resto de familia y amigos Andrea, Karina, Valeria, Nancy y Carlos que de una u otra manera me han llenado de sabiduría para terminar la tesis.

A todos en general por darme el tiempo para realizarme profesionalmente.

*Yonder Leonel*

## **AGRADECIMIENTO**

A Dios que con su bondad nos ayudó alcanzar uno de los logros más importantes de nuestras vidas, por fortalecer e iluminar nuestros corazones y darnos la oportunidad por continuar con nuestros ideales.

A nuestros padres, hermanos y amigos, por ser el motor de fuerza principal en nuestras vidas, por estar incondicionalmente, brindándonos su apoyo y su amor.

Agradecemos a nuestro director de tesis Ing. Daniel Guamán quien nos guió durante la realización de este trabajo, brindándonos su valioso tiempo.

De igual manera extendemos nuestro agradecimiento a la titulación de la carrera de Sistemas Informáticos y Computación quienes nos compartieron sus conocimientos y experiencias profesionales.

Agradecemos al departamento de Gestión vehicular e Infraestructura del Gobierno Provincial de Loja quien nos colaboró con la información necesaria para poder realizar y dar culminación a este proyecto.

En general quisiéramos agradecer a todas las personas que de cierta forma formaron parte de nuestras vidas y de alguna u otra manera nos motivó para el desarrollo de nuestra tesis.

*Andrea y Yonder*

## ÍNDICE DE CONTENIDOS

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN.....	ii
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
ÍNDICE DE TABLAS.....	viii
ÍNDICE DE FIGURAS.....	viii
RESUMEN.....	1
ABSTRACT .....	2
INTRODUCCIÓN.....	3
GLOSARIO DE TÉRMINOS .....	5
CAPÍTULO I.....	6
MARCO TEÓRICO .....	6
1.1 Introducción .....	7
1.2 Metodologías de Desarrollo .....	7
1.3 Metodología RUP.....	9
1.3.1 Definición. ....	9
1.3.2 Dimensiones de RUP. ....	10
1.3.3 Características esenciales de RUP .....	11
1.3.4 Ciclo de vida de RUP .....	12
1.4 Arquitectura de Software.....	13
1.4.1 Importancia de la Arquitectura de Software. ....	13
1.4.2 Patrón Arquitectónico .....	14
1.4.3 Patrón Arquitectónico Modelo Vista Controlador .....	16
1.4.4 Estilos de la Arquitectura del Software .....	17
1.5 Sistema Gestor de Base de Datos (SGBD).....	17
1.5.1 POSTGRESQL.....	18
1.5.2 Procedimientos Almacenados. ....	19
1.6 Tecnología .....	20
1.6.1 Leguajes para desarrollar aplicaciones web. ....	20
1.6.1.1 PHP .....	21
1.6.1.2 Javascript .....	22
1.6.1.3 CSS1 (Cascade Style Sheets – version 1).....	22
1.6.2 Framework CodeInteger.....	23
1.6.3 Framework de estilos Bootstrap.....	24
CAPÍTULO II 25	
DEFINICIÓN DEL PROBLEMA Y ANÁLISIS INICIAL.....	25
2.1 Introducción .....	26
2.2 Políticas.....	26
2.3 Proceso del Negocio.....	26
2.3.1 Descripción del proceso para la entrega de la Orden Movilización. ....	26
2.3.2 Descripción del proceso de entrega de Orden de Viáticos.....	28
2.3.3 Descripción del proceso de asignación de Combustibles. ....	30
2.3.4 Descripción del Proceso de para la Orden de Mantenimiento Correctivo .....	32
2.3.5 Descripción del proceso de Emisión de Lubricantes y Repuestos .....	34
2.4 Problemática.....	36
2.5 Planteamiento de la Solución.....	37
2.6 Beneficios de la solución.....	39
2.7 Resultados Esperados .....	39
CAPÍTULO III.....	41
DESARROLLO DEL SISTEMA DE GESTIÓN VEHICULAR .....	41
3.1 Introducción .....	42
3.2 Metodología de desarrollo de software .....	42
3.3 Fases de la metodología.....	42

3.4 Fase de Inicio del Sistema de Gestión Vehicular .....	42
3.4.1 Análisis del Estado Actual .....	43
3.4.2 Entrevistas .....	43
3.4.3 Documento de Visión .....	44
3.4.3.1 Definición del Problema.....	44
3.4.3.2 Roles de Usuario.....	45
3.4.3.3 Participantes del Proyecto .....	45
3.4.4 Especificación de Requerimientos.....	45
3.4.4.1 Perspectiva del Software .....	46
3.4.4.2 Características del Software .....	46
3.5 Fase de Elaboración del Sistema de Gestión Vehicular .....	48
3.5.1 Especificación de Casos de Uso .....	48
3.5.2 Arquitectura de Software .....	50
3.5.2.1 Descripción del Patrón Arquitectónico MVC.....	51
3.6 Fase de Construcción del Sistema de Gestión Vehicular .....	59
3.6.1 Diseño de interfaz de Usuario .....	60
3.6.1.1 Diseño de la interfaz de ingreso al sistema.....	60
3.6.1.2 Diseño de la Interfaz de la Pantalla Principal .....	60
3.7 Fase de Transición del Sistema de Gestión Vehicular .....	61
CAPÍTULO IV .....	62
PRUEBAS PARA EL SISTEMA DE GESTIÓN VEHICULAR .....	62
4.1 Introducción .....	63
4.2 Tipos de Pruebas.....	63
4.3 Estrategia de las Pruebas .....	63
4.3.1 Técnicas de Especificación de las Pruebas.....	64
4.3.1.1 Planificación .....	64
4.3.1.2 Diseño de las pruebas .....	64
4.3.1.2 Ejecución.....	65
4.3.1.5 Evaluación y cierre .....	66
4.4 Herramientas para Pruebas .....	66
4.5 Configuración del ambiente de las pruebas .....	66
4.6 Ejecución y Resultados de las Pruebas .....	67
4.6.1 Pruebas Unitarias.....	67
4.6.2 Pruebas del sistema .....	68
4.6.3 Pruebas de Interfaz de Usuario .....	69
4.6.4 Pruebas de Rendimiento .....	70
4.6.5 Pruebas de Seguridad.....	71
4.6.2 Pruebas Funcionales.....	72
CAPÍTULO V .....	75
CONCLUSIONES, RECOMENDACIONES.....	75
CONCLUSIONES .....	76
RECOMENDACIONES.....	77
BIBLIOGRAFÍA.....	78
ANEXOS.....	80
ANEXO A: ENTREVISTAS .....	81
ANEXO B: POLITICAS .....	83
ANEXO C: DOCUMENTO DE VISIÓN.....	86
ANEXO D: DOCUMENTO DE REQUERIMIENTOS .....	94
ANEXO E: ESPECIFICACION DE CASOS DE USO .....	104
ANEXO F: DOCUMENTO DE ARQUITECTURA DE SOFTWARE .....	113
ANEXO G: PLAN DE PRUEBAS .....	121
ANEXO H: MANUAL DE USUARIO.....	165
ANEXO I: ACTAS DE ENTREGA .....	184
ANEXO I: VALIDACIÓN DEL SISTEMA. ....	185

## ÍNDICE DE TABLAS

Tabla 1. Glosario de Términos.....	5
Tabla 2. Cuadro Comparativo de las metodologías de Desarrollo .....	8
Tabla 3. Patrones arquitectónicos y atributos de calidad .....	15
Tabla 4. Cuadro comparativo de los estilos arquitectónicos.....	17
Tabla 5. Cuadro comparativo de los Gestores de Base de Datos .....	18
Tabla 6. Cuadro comparativo de los lenguajes de programación por guiones .....	21
Tabla 7. Descripción del proceso para la entrega de Orden de Movilización. ....	27
Tabla 8. Descripción del Proceso de la emisión de la Orden de Viáticos. ....	29
Tabla 9. Descripción del proceso de Asignación de Combustible. ....	31
Tabla 10. Descripción del proceso Orden de Mantenimiento Correctivo. ....	33
Tabla 11. Descripción del proceso de Emisión de Lubricantes y Repuestos.....	35
Tabla 12. Problemas existentes dentro del GPL .....	37
Tabla 13. Fases de la metodología RUP y sus artefactos .....	42
Tabla 14. Definición del Problema .....	44
Tabla 15. Roles de usuario .....	45
Tabla 16. Participantes del proyecto .....	45
Tabla 17. Especificación de Requerimientos funcionales.....	46
Tabla 18. Características del Software. ....	47
Tabla 19. Clase y métodos del models .....	52
Tabla 20. Vistas del módulo Administración.....	55
Tabla 21. Vistas del módulo Reportes .....	56
Tabla 22. Vistas del módulo Gestión de Movilización.....	56
Tabla 23. Vistas del módulo de Talleres .....	57
Tabla 24. Métodos de la clase .....	58
Tabla 25. Tipo de pruebas aplicadas al Software.....	63
Tabla 26. Herramientas para Pruebas .....	66
Tabla 27. Ambiente y Configuración de las Pruebas.....	66
Tabla 28. Ejecución de Pruebas Unitarias .....	67
Tabla 29. Resultados de las Pruebas Unitarias.....	67
Tabla 30. Pruebas del sistema ejecutadas.....	69
Tabla 31. Pruebas de Interfaz de Usuario.....	69
Tabla 32. Pruebas de Rendimiento.....	71
Tabla 33. Pruebas Funcionales ejecutadas .....	73
Tabla 34. Resultado de las Pruebas Funcionales. ....	73

## ÍNDICE DE FIGURAS

Figura 1. Disciplinas, fases, interacciones del RUP. ....	10
Figura 2. Funcionamiento de CodeInteger .....	24
Figura 3. Flujo del proceso de entrega de la Orden de Movilización .....	28
Figura 4. Flujo del proceso de entrega de la Orden de Viáticos. ....	30
Figura 5. Flujo de proceso para la asignación de combustibles. ....	32
Figura 6. Flujo de proceso de la Orden de Mantenimiento Correctivo.....	34
Figura 7. Flujo de proceso para emisión de Orden de Lubricantes y Repuestos.....	36
Figura 8. Diagrama de casos de uso .....	49
Figura 9. Arquitectura implementada .....	50
Figura 10. Estructura del modelo, vista y controlador.....	51
Figura 11. Estructura del modelo .....	52
Figura 12. Estructura de la Vista .....	55
Figura 13. Estructura del Controlador .....	57
Figura 14. Interfaz de ingreso al sistema .....	60
Figura 15. Interfaz principal.....	61

Figura 16. Interfaz Reportes .....	61
Figura 17. Ciclo del plan de pruebas.....	64
Figura 18. Modelo de Pruebas en V.....	65
Figura 19. Resultado de las Pruebas Unitarias. ....	68
Figura 20. Resultados de las pruebas de interfaz de usuario.....	70
Figura 21. Resultado de las Pruebas Funcionales .....	74
Figura 22. Interfaz Sistema de Gestión Vehicular .....	97
Figura 23. Diagrama de Casos de Uso. Sistema de Gestión Vehicular.....	112
Figura 24. Diagrama de Clases .....	117
Figura 25. Diagrama Entidad-Relación .....	118
Figura 26. Diagrama de Secuencia del Módulo Movilización .....	119
Figura 27. Diagrama de Secuencia del Módulo Talleres.....	119
Figura 28. Diagrama de Despliegue.....	120
Figura 29. Codificación de la prueba Unitaria Solicitud de Movilización. ....	126
Figura 30. Formulario Solicitud de Movilización .....	127
Figura 31. Resultado de la prueba Unitaria Solicitud de Movilización .....	127
Figura 32. Codificación Prueba Unitaria Orden de Movilización .....	128
Figura 33. Asignación vehículo .....	129
Figura 34. Resultado de la Prueba Unitaria Asignación de vehículo .....	129
Figura 35. Codificación de la prueba Unitaria Orden de Combustible. ....	130
Figura 36. Formulario de Asignación de combustibles.....	130
Figura 37. Resultado de la prueba de unitaria Asignación de vehículo. ....	131
Figura 38. Codificación de la prueba unitaria despacho de Combustible. ....	132
Figura 39. Formulario de despacho de combustibles.....	132
Figura 40. Resultado de la prueba unitaria despacho de combustible. ....	132
Figura 41. Codificación de la prueba unitaria Orden de mantenimiento. ....	133
Figura 42. Formulario de orden de mantenimiento.....	134
Figura 43. Resultado de la prueba unitaria Solicitud de Mantenimiento .....	134
Figura 44. Codificación de la prueba unitaria Reportes.....	135
Figura 45. Vista reportes conductores. ....	136
Figura 46. Resultado de la prueba unitaria de los Reportes.....	136
Figura 47. Codificación de la prueba unitaria del módulo Administración.....	137
Figura 48. Formulario de Registro de Usuario.....	138
Figura 49. Resultado de la prueba unitaria registro de usuario. ....	138
Figura 50. Pantalla de Inicio de Sesión.....	165
Figura 51. Pantalla de Ingreso al Sistema de Gestión Vehicular.....	166
Figura 52. Interfaz de la administración de usuarios.....	166
Figura 53. Interfaz para añadir un Usuario al Sistema. ....	167
Figura 54. Formulario para registrar un usuario al sistema. ....	167
Figura 55. Interfaz para listar las actividades de los usuarios. ....	167
Figura 56. Fecha de búsqueda para listar los usuarios.....	168
Figura 57. Campos de búsqueda para listar los usuarios.....	168
Figura 58. Reporte de logs.....	168
Figura 59. Interfaz para asignar Vehículo chofer.....	169
Figura 60. Formulario para la Asignación de un chofer a un vehículo.....	169
Figura 61. Formulario de visualización de asignación vehículo chofer.....	170
Figura 62. Interfaz de administración del tipo licencias.....	170
Figura 63. Formulario Añadir nueva licencia .....	170
Figura 64. Formulario de actualización tipo licencia.....	171
Figura 65. Interfaz de administración de los combustibles.....	171
Figura 66. Formulario para añadir un nuevo combustible.....	171
Figura 67. Formulario para editar un combustible.....	171
Figura 68. Interfaz de administración de gasolineras.....	172
Figura 69. Formulario para añadir nueva gasolinera.....	172

Figura 70. Formulario para editar gasolinera. ....	172
Figura 71. Interfaz para la administración de vehículos. ....	173
Figura 72. Interfaz para añadir un nuevo vehículo. ....	173
Figura 73. Formulario para añadir un nuevo vehículo. ....	173
Figura 74. Formulario de ingreso vehículo. ....	174
Figura 75. Formulario de actualización de un vehículo. ....	174
Figura 76. Interfaz de Administración del Conductor.....	174
Figura 77. Interfaz para añadir un nuevo conductor.....	175
Figura 78. Formulario para añadir un nuevo conductor.....	175
Figura 79. Formulario para añadir un nuevo Conductor. ....	175
Figura 80. Interfaz para realizar un Búsqueda avanzada de mantenimiento. ....	176
Figura 81. Interfaz de Búsqueda avanzada para el reporte de control de combustible.....	176
Figura 82. Interfaz del reporte de vencimiento de matrícula vehicular.....	177
Figura 83. Interfaz para generar el reporte de Órdenes de Movilización. ....	177
Figura 84. Interfaz para generar reporte de Novedades.....	178
Figura 85. Formulario para registrar una Solicitud de Movilización. ....	178
Figura 86. Interfaz para seleccionar la ruta.....	179
Figura 87. Visualización de la ruta. ....	179
Figura 88. Interfaz de Administración de las solicitudes.....	180
Figura 89. Asignación de Vehículo.....	180
Figura 90. Asignación de Orden de Combustible. ....	180
Figura 91. Formulario para la asignación de Combustible. ....	181
Figura 92. Interfaz de Despacho de combustible. ....	181
Figura 93. Formulario para una Solicitud de Mantenimiento. ....	182
Figura 94. Interfaz para el mostrar el Historial de Mantenimiento.....	182
Figura 95. Interfaz para Administración de Plan de mantenimiento Preventivo.....	183
Figura 96. Interfaz de administración para el plan de mantenimiento.....	183

## RESUMEN

En el presente proyecto de fin de titulación se describe el análisis de las herramientas y tecnologías necesarias para el diseño y desarrollo de un Sistema Web.

Se identificó el problema actual que tiene el Departamento de Gestión Vehicular del Gobierno Provincial de Loja, Institución que no cuenta con un registro confiable del correcto uso de sus vehículos e información relevante de los mismos.

A partir del problema, se propuso desarrollar un software para sistematizar el proceso que la Institución lleva para el control del parque automotor, mantenimiento y despacho de combustible al cumplir una actividad asignada. Se utilizó la metodología de software RUP, cumpliendo con los entregables en cada fase.

Se implementó un sistema, el cual generó información importante para la toma de decisiones acordes a la nueva regulación del control vehicular. Además de entregar un registro del control de consumo de combustible de cada vehículo y kilómetros recorridos por cada ruta asignada. De igual forma, presenta notificaciones de los vehículos que necesitan realizar un mantenimiento correctivo, fechas de vencimiento de matrícula por cada vehículo.

**Palabras claves:** CodeInteger, control vehicular, framework, GPL, mantenimiento, RUP, vehículos.

## ABSTRACT

In this final project, we described analysis and tools needed to design and develop a Web System technologies.

The current problem with the Vehicle Management Department of the Provincial Government of Loja, institution that does not have a reliable record of the correct use of their vehicles and relevant information of these.

From the problem, it was proposed to develop software to systematize the process that the institution takes control of the fleet, maintenance and delivery of fuel to meet an assigned activity. RUP methodology used software, meeting the deliverables of each phase.

A system, which generated important for decision-making in line with the new regulation was implemented vehicular control information. Besides providing a record of the fuel consumption control of each vehicle and kilometers traveled by each assigned route. Similarly, notifications presented vehicles that need to perform corrective maintenance, registration deadlines for each vehicle.

**Keywords:** CodeIgniter, framework, GPL, maintenance, RUP, vehicle control, vehicles.

## INTRODUCCIÓN

El presente proyecto de fin de titulación consiste en desarrollar e implementar un sistema informático de gestión vehicular para el registro de entrada y salida de vehículos que son propiedad del Gobierno Provincial de Loja (GPL) en reemplazo al control manual que actualmente está en uso y dificulta obtener la información real, que permita un control efectivo de estos bienes de la Institución.

Este proyecto de fin de titulación es de suma importancia ya que permitirá automatizar el control manual que actualmente se está utilizando. Para lo cual se integrará con los demás sistemas que posee la Institución como son: recursos humanos, bienes, y activos, para así evitar la duplicación de información. Además este sistema se instalará en la propia infraestructura montada por el Departamento de Infraestructura Tecnológica.

El proyecto tiene por objetivo principal ***“Diseñar e implementar un Sistema de Gestión Vehicular que permita automatizar el registro de solicitudes de movilización, asignación de combustible, mantenimientos preventivos y correctivos de los vehículos”***. Para dar alcance a este objetivo se debe dar cumplimiento a los siguientes objetivos específicos planteados:

- Identificar la problemática y estado actual del GPL para un análisis eficiente.
- Aplicar la metodología RUP para el desarrollo de software.
- Seleccionar e implementar un modelo de arquitectura para la construcción del software.

La estructura de este proyecto de fin de titulación está constituida por los siguientes capítulos.

**Capítulo 1:** Se detallan los conceptos a tratar dentro del marco teórico como son: las metodologías de desarrollo de software, tecnologías, framework de desarrollo, patrones arquitectónicos y gestores de base de datos con cuadros comparativos que determinan ventajas y desventajas que tiene cada uno de los temas y la justificación del concepto designado para integrarlo en el desarrollo e implementación del sistema.

**Capítulo 2:** Se describen y se visualizan gráficamente los procesos que llevan actualmente la Coordinación de Gestión Vehicular, problemática, análisis de la solución, beneficios y resultados esperados al implementarse el sistema. Estos permiten obtener el panorama de los flujos de procesos del Departamento de Gestión Vehicular y del Departamento de Talleres

y la posible solución que detallan los beneficios que brindará el Sistema de Gestión Vehicular para la automatización de procesos y emisión de ordenes.

**Capítulo 3:** Describe el desarrollo e implementación del Sistema de Gestión Vehicular que cumple con las fases de la metodología RUP que son: inicio, elaboración, construcción y transición con los artefactos usados para la documentación del sistema y los frameworks como son: Codeigniter para la codificación del software que permite incluir y diseñar el patrón arquitectónico MVC(Modelo – Vista – Controlador) y Bootstrap para el diseño de la interfaz gráfica del sistema y el acoplamiento a la arquitectura de tres capas donde la capa de negocio incluye el código HTML, PHP y JavaScript en un servidor, la capa de datos incluye las bases de datos de PostgreSQL y MySQL en otro servidor y la capa de presentación incluye a los clientes que acceden al sistema.

**Capítulo 4:** Contiene las pruebas aplicadas para determinar el correcto POST y GET de datos en el código, la calidad, el funcionamiento y seguridad del Sistema de Gestión Vehicular para evitar datos redundantes e innecesarios, carga rápida y sin contratiempos de los datos, los procesos bien definidos para crear y emitir ordenes, evitar los ataques de inyección SQL y los ataques de Cross Site Scripting, las pruebas usadas son las siguientes: unitarias, de sistema, carga, rendimiento, interfaz de usuario, seguridad, funcionalidad, usabilidad, aceptación y regresión utilizadas para la aceptación del cliente y puesta en marcha del sistema.

**Capítulo 5:** Se describen las conclusiones y recomendaciones obtenidas a partir de la problemática, estado actual, procesos, requerimientos, casos de usos, codificación, pruebas y documentación del Sistema de Gestión Vehicular para el Gobierno Provincial de Loja y posibles trabajos futuros que se podrían dar en cuanto a la integración de nuevas tecnologías como por ejemplo el GPS para el monitoreo de las rutas asignadas.

Durante el desarrollo de este proyecto se brindaron todas las facilidades por parte del **Gobierno Provincial de Loja, Coordinación de Gestión Vehicular y Departamento de Infraestructura Tecnológica** proporcionando información para detallar el proceso y problemática actual. Además se hizo uso de la infraestructura montada para la gestión de los sistemas, para la implementación y pruebas respectivas.

## GLOSARIO DE TÉRMINOS

Tabla 1. Glosario de Términos

TÉRMINO	SIGNIFICADO
API	Interfaz de Programación de Aplicaciones.
BOOTSTRAP	Framework de twitter para desarrollo de aplicaciones web.
CODEIGNITER	Un framework PHP para el desarrollo rápido de aplicaciones web.
CONGOPE	Consortio de Gobiernos Autónomos Provinciales de Ecuador.
CSS	Hojas de Estilos en Cascada.
FRAMEWORK	Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve de referencia.
DLL	Bibliotecas o librerías de Enlace Dinámico.
GIST	Árbol de Búsqueda Generalizada
GPL	Gobierno Provincial de Loja.
HTML	Lenguaje de Marcas de Hipertexto.
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos.
JAVASCRIPT	Lenguaje interpretado que corre todos los navegadores.
JDBC	Conexión de base de datos estándar utilizado por los programas Java.
MVC	Modelo, Vista, Controlador.
PHP	Pre-procesador de Hipertexto.
RUP	Proceso Racional Unificado.
SGBD	Sistema Gestor de Bases de Datos.
UML	Lenguaje Unificado de Modelado.

Elaboración: (Ordóñez & Guaman, 2015)

**CAPITULO I**  
**MARCO TEÓRICO**

## 1.1 Introducción

En el presente capítulo se expondrán conceptos básicos, complementarios y específicos del proyecto como son: la metodología utilizada, framework de desarrollo, patrones de diseño y arquitectura de software, tecnologías para el desarrollo y diseño del software como PHP, JavaScript, HTML5 y CSS .

Se ha seleccionado la metodología RUP para el desarrollo del software debido a que, a través de él se puede realizar un levantamiento exhaustivo de requerimientos, buscando detectar defectos en las fases iniciales, y reducir el número de cambios. “En comparación con las otras metodologías como XP, la ventaja principal de RUP es que se basa en las mejores prácticas probadas que se han intentado en el campo frente a las prácticas inestables de XP” (Mitaritonna A., 2010)

## 1.2 Metodologías de Desarrollo

“Las metodologías de desarrollo han evolucionado de manera significativa, a medida de la complejidad de un producto de software, por ello es que existen numerosas propuestas metodológicas que existen e inciden en distintas dimensiones del proceso de desarrollo” (Letelier & Penadés, 2006).

“Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado” (Laboratorio Nacional de Calidad del Software de INTECO, 2009)

En la actualidad existen metodologías tradicionales, y ágiles. ***Dentro de las metodologías ágiles*** el objetivo principal es permitir a los equipos desarrollar software, responder rápidamente a los cambios que puedan surgir a lo largo del proyecto, mientras en las ***metodologías tradicionales*** ponen mayor énfasis en planificación y el control del proyecto, tiempo, costo y estimación en especificación precisa de requisitos y modelado, con el fin de conseguir un software más eficiente. Dentro las metodologías más utilizadas tenemos:

Tabla 2. Cuadro Comparativo de las metodologías de Desarrollo

METODOLOGÍA	DESCRIPCIÓN	VENTAJAS	DESVENTAJAS	APLICACIONES
<b>Cascada</b>	Siguen una secuencia lógica y cada etapa es directamente dependiente de que se culmine la anterior.	La planificación es sencilla, permite trabajar con personal poco calificado.	Se necesita tener todos los requisitos desde el principio.	Se aplica en situaciones donde el software es simple y de dominio bien conocido.
<b>Incremental</b>	Es la unión de las mejores funcionalidades del modelo cascada y el modelo de prototipos.	Se reduce en tiempo de desarrollo inicial y se implementa la funcionalidad parcial.	No se recomienda para casos en sistemas de tiempo real de alto nivel de seguridad.	Interactivo a los clientes quienes identifican las funcionalidades que proporciona el sistema
<b>Móvil-D</b>	Es un modelo ágil enfocado a grupos pequeños.	Se relaciona de manera constante con el cliente. Para cada función se realiza un ciclo de 3 días para planificar, trabajar en el proyecto y presentarlo.	No Existe definición de costo y tiempo, solo es apto para grupo mínimo de 10 personas.	Para aplicaciones de software Móvil
<b>XP</b>	Modelo en el que se define un plan para desarrollar y liberar software y además poder revisarlo para incorporar funcionalidades.	Trabajo en equipo. Pone énfasis en la comunicación.	No existe definición de costo y tiempo de desarrollo.	Apto para aplicaciones Móviles.
<b>RUP</b>	Se caracteriza por ocupar el modelo iterativo e incremental. Está centrado en una arquitectura.	Tiene estrecha relación con el cliente.	Método pesado por el grado de complejidad que puede ser no muy adecuado	Principalmente para tipos de proyectos web y móviles.

Fuente: (Sibaja J., 2015). Cuadro comparativo de las metodologías de desarrollo.

En la Tabla 2 se han descrito las metodologías más utilizadas por los desarrolladores, una metodología debe ser lo suficientemente precisa como para que todo el mundo la entienda, a la vez que sea adaptable en cada proyecto de desarrollo, lo suficientemente sencilla como para que no resulte muy grave su utilización, a la vez completa y compleja como para su utilización dentro de un equipo.

De acuerdo a estas características se ha escogido la metodología RUP para el desarrollo de este proyecto, siendo de las que requiere mucha disciplina para su implementación ya que inicialmente se debe ajustar la metodología a las necesidades del proyecto. Otra de las características por la cual se escoge esta metodología es por que maneja iteraciones en cada una de sus fases, permitiendo refinarlo cada vez más para su entrega final, y satisfacción con el cliente.

Un aspecto muy importante dentro de esta metodología es la documentación técnica y metodológica, en la aplicación de modelados utilizando el lenguaje UML, documentos de proceso metodológico, pruebas y entregables.

### 1.3 Metodología RUP

#### 1.3.1 Definición.

El antecedente más importante se ubica en el año 1967 con la metodología Ericsson (Ericsson Approach) elaborada por Ivar Jacobson, una aproximación de desarrollo basado en componentes que introdujo el concepto de Casos de Uso, entre los años 1987 a 1995, Jacobson fundo *Rational Software Corporation* y adquiere *Objectory ABy* entre 1995 y 1997. Se desarrolla *Rational Objectory Process (ROP)* a partir de *Objectory 3.8* y del Enfoque Rational (*Rational Approach*) adoptando UML como lenguaje de modelado. Desde ese entonces y a la cabeza de (Grady Booch, Ivar Jacobson y James Rumbaugh), Rational Software desarrolló e incorporó diversos elementos para expandir ROP, destacándose especialmente el flujo de trabajo conocido como modelado del negocio. En junio del 1998 se lanza *Rational Unified Process*.

El RUP es “un proceso de negocio genérico para la ingeniería de software orientada a objetos. Describe una familia de procesos de ingeniería de software relacionados, que comparten una estructura y arquitectura común de procesos. El RUP provee un método disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo, Su meta es asegurar la producción de software de alta calidad, que conozca las necesidades de los usuarios, dentro de un calendario, y presupuesto predecible.” (Gracían S., 2006)

“Las siglas RUP significan ***Rational Unified Process***, un producto del proceso de ingeniería del software que proporciona un enfoque disciplinario para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempos establecidos” (Rueda J., 2006)

Se puede concluir que RUP es una metodología que recolecta las mejores prácticas de Ingeniería de Software, a través de 4 fases con flujos de trabajo que pueden ser concurrentes, a medida que se adapta a proyectos de pequeña y a mediana escala, haciendo un ciclo de vida eficaz, para que al final se entregue al cliente un software o producto de calidad.

Con el uso de esta metodología los usuarios, o parte interesada del proyecto podrán conocer los artefactos elaborados por cada fase de desarrollo, que podrán ser utilizados para próximas versiones o cambios que puedan suceder en el proyecto.

### 1.3.2 Dimensiones de RUP.

Este proceso puede ser descrito en dos dimensiones o ejes según (Metzner & Niño, 2012).

- Eje horizontal: representa tiempo y demuestra los aspectos del ciclo de vida de proyecto de software.
- Eje vertical: representa las disciplinas que agrupan actividades definidas lógicamente por la naturaleza.

La primera dimensión representa el aspecto dinámico del proceso y se expresa en términos de fases, iteración, y finalización de las fases. La segunda dimensión representa el aspecto estático del proceso: como se describen los términos de componentes de proceso las disciplinas, las actividades, los flujos de trabajo los artefactos y los roles.

Como se observa en la Figura 1 en el eje vertical se encuentran los flujos de proceso en relación con cada fase de los ejes horizontal en ciertos plazos de tiempo. Esto nos indica que en las iteraciones más tempranas como: modelo de negocios y requisitos se emplea más tiempo en definir el alcance y los requisitos del proyecto, mientras en las últimas iteraciones se va más tiempo en la implementación y pruebas.

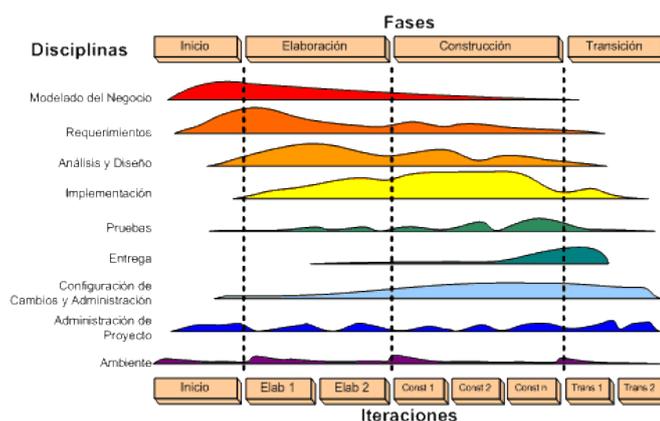


Figura 1. Disciplinas, fases, interacciones del RUP.  
Fuente: (Arcaya L., 2011).

A continuación se describen las disciplinas que maneja RUP (Metzner & Niño, 2012).

- **Modelado de Negocio:** Comprender las necesidades del Negocio.
- **Requisitos:** Traducir necesidades del negocio en comportamientos de un sistema automático.
- **Análisis y Diseño:** Traducir requisitos en una arquitectura software.

- **Implementación:** Crear software que encaje en la arquitectura y realice el comportamiento solicitado.
- **Pruebas:** Asegurarse de que el comportamiento solicitado es correcto, y que todo lo solicitado está presente.
- **Configuración y Gestión del Cambio:** Mantener traza de las distintas versiones de todos los productos.
- **Gestión del proyecto:** Gestionar planificaciones y recursos.
- **Entornos:** Establecer y mantener el entorno de desarrollo.
- **Despliegue:** Todo aquello necesario para instalar el proyecto.

En el desarrollo de este proyecto se trabajará con las disciplinas: modelado del negocio, requisitos, análisis y diseño, implementación y pruebas correspondientes a cada fase de la metodología estas se describirán en el Capítulo 3.

### 1.3.3 Características esenciales de RUP

El proceso de software propuesto por RUP tiene tres características esenciales:

**Casos de uso:** representan los requerimientos base para el desarrollo del sistema que constituyen el punto de partida para las tareas de análisis y diseño.

**Proceso Centrado en la Arquitectura:** la arquitectura involucra los aspectos dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican como tienen que ser construido el sistema y ayuda a determinar el orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo.

**Iterativo e incremental:** es un proceso iterativo e incremental donde el trabajo se divide en partes más pequeñas, permitiendo el equilibrio entre casos de uso y la arquitectura se vayan logrando durante cada proyecto, así durante todo el proceso de desarrollo. El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo (Requisitos, Análisis, Diseño e implementación y Pruebas), esta iteración se analiza cuando se termina. (Laya J., 2014)

De acuerdo a estas características del proyecto se utilizará la característica **proceso dirigido por Casos de Uso** en la cual se especificarán requisitos funcionales, que servirán como guía para el diseño, implementación y prueba del software. Los casos de uso no solo inician el proceso de desarrollo sino que proporcionan un hilo conductor permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso

de desarrollo. También se hará uso de la característica **proceso Centrado en la Arquitectura** que involucra los aspectos estáticos y dinámicos más significativos del sistema que se relacionan con la toma de decisiones indicando cómo tiene que ser construido el sistema y en qué orden se van cumpliendo los requisitos no funcionales entre ellos están la calidad del sistema, reutilización, rendimiento y capacidad de evolución. También se hará uso de la característica del **proceso Iterativo e Incremental**, cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura, cada iteración se va analizando cuando se termina, determinando si existen nuevos requisitos o han cambiado los existentes, afectando las iteraciones siguientes.

#### 1.3.4 Ciclo de vida de RUP

Según Loira (2006) el ciclo de vida del RUP es “una adaptación de la metodología en Espiral la cual permite crear actividades en secuencias semi-ordenadas, entonces el ciclo de vida del RUP organiza las tareas en fases e iteraciones tal como lo hace la metodología en Espiral.”

La ingeniería del software establece una serie de modelos que establecen y muestran por las distintas etapas por las que pasa un producto de software desde su concepción inicial, pasando por su desarrollo, puesta en marcha y posterior mantenimiento hasta la retirada del producto. A estos modelos se los denomina “Modelo de ciclo de vida del software”.

El ciclo de vida de RUP se divide en 4 fases fundamentales que son: Inicio, Elaboración, Construcción y Transición; cada una de estas fases se divide en iteraciones, cada una de las cuales produce una pieza de software.

A continuación se describen las fases del ciclo de vida del RUP según Carrillo (2008):

**Fase Inicio:** En esta fase se identifican las necesidades de los usuarios, alcance del proyecto, y se determinan los principales casos de uso.

**Fase de Elaboración:** Se completan los casos de uso, se establece la arquitectura a utilizar junto con todos los requerimientos se detalla el ambiente de desarrollo, al final de esta fase se debe validar los casos de uso.

**Fase de Construcción:** se completa la funcionalidad del sistema, administrar los cambios de acuerdo a las pruebas realizadas por parte de los usuarios y una mejora total del proyecto.

**Fase de Transición:** Esta fase comienza con una prueba beta preliminar para el usuario, capacitación a los usuarios finales, se realizan los manuales de usuario y manual de programador. Se realiza un plan de pruebas que pueden ser las unitarias y las de aceptación de usuario.

Para el desarrollo del presente proyecto de fin de titulación se hará el empleo de las 4 fases del RUP antes descritas, cumpliendo con la entrega de los artefactos necesarios de acuerdo al proyecto, y creando las iteraciones necesarias para la entrega de un producto de calidad.

#### **1.4 Arquitectura de Software**

Según Len (2003) “La arquitectura de un sistema de software es la estructura o estructuras del sistema, que incluye elementos de software, las propiedades externamente visibles de esos elementos y la relación entre ellos”.

Cuando se habla de estructuras estas pueden ser estructuras dinámicas y estáticas, las dinámicas definen la forma en que el sistema funciona en el tiempo de ejecución. Cuando hablamos de propiedades visibles estamos hablando de iteraciones de un elemento o un sistema con su entorno externo.

Según Alegsa (2010), “una arquitectura de software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción de un software para un sistema de información”.

Otra definición de arquitectura del software según el estándar (IEEE Std 1471-2000) “la arquitectura del software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”.

De acuerdo a estas opiniones podemos decir que una arquitectura de software constituye un conjunto de patrones y abstracciones que proporcionan un marco de referencia necesario para guiar la construcción del software, siendo el diseño de más alto nivel de la estructura de un sistema, programa o aplicación.

La arquitectura de software en un sentido estricto, se define como el conjunto de estructuras que componen el sistema, lo que incluye elementos de software, las relaciones entre los mismos y las propiedades tanto de los elementos como de sus relaciones. (Bourque & Fairley, 2014).

##### **1.4.1 Importancia de la Arquitectura de Software.**

La necesidad del manejo de la arquitectura de software nace de los sistemas grandes o pequeños que se proponen como solución de un problema determinado. En medida que los sistemas crecen en complejidad ya sea por el número de requerimientos o por el impacto que

vaya a dar es necesario establecer técnicas que permitan descomponer los sistemas en piezas que agrupan aspectos específicos del mismo.

“La arquitectura de software define la estructura del sistema, esto puede tener un impacto directo a la satisfacción de los atributos de calidad del sistema los cuales pueden ser (desempeño, usabilidad y modificabilidad) Estos atributos de calidad son parte de los requerimientos no funcionales del sistema”. (Camacho, Cardeso, & Nuñez, 2004).

La arquitectura de software puede considerarse entonces como el puente entre los requerimientos del sistema y la implementación. De igual manera serán de particular importancia las propiedades no funcionales del sistema de software, pues influyen en la calidad del mismo, ya que poseen impacto en el desarrollo y mantenimiento de los sistemas. Entre las propiedades no funcionales más importantes son: eficiencia, mantenibilidad, interoperabilidad, confiabilidad, reusabilidad y modificabilidad. Según Suárez (2008) “cuando hablamos de arquitectura de software, se hace alusión específica a la estructura del software, como la organización de los componentes y relaciones entre ellos: los requerimientos que deben satisfacer el sistema y las restricciones a las que se está sujeto”.

#### **1.4.2 Patrón Arquitectónico**

Según Buschmann (1996) “define patrón como una regla que consta de tres partes, la cual expresa una relación entre un contexto, un problema y una solución”. Un patrón sigue el siguiente esquema:

- Contexto. Es una situación de diseño en la que aparece un problema de diseño.
- Problema. Es un conjunto de fuerzas que aparecen repetidamente en el contexto.
- Solución. Es una configuración que equilibra estas fuerzas. Ésta abarca: estructura con componentes y relaciones, comportamiento a tiempo de ejecución, aspectos dinámicos de la solución, como la colaboración entre componentes, la comunicación entre ellos, etc.

Podemos decir que los patrones arquitectónicos expresan el esquema de organización estructural fundamental para los sistemas de software. La elección de un patrón arquitectónico es una decisión fundamental para el diseño.

Según Ayala (2011) los patrones de arquitectura de software comprenden las siguientes características:

- El diseño de más alto nivel de la estructura del sistema.
- Los patrones y abstracciones necesarios para guiar la construcción del software.
- Los analistas, diseñadores, programadores, beta testers, etc., que trabajen en una línea común que permita cubrir restricciones y alcanzar los objetivos del sistema.
- Los objetivos del sistema no solamente funcionales, sino de mantenimiento, auditoria, flexibilidad e interacción con otros sistemas.
- Las restricciones que limitan la construcción del sistema acorde a las tecnologías disponibles para su implementación.

En la Tabla 3 se describe cada uno de los patrones arquitectónicos y sus atributos.

Tabla 3. Patrones arquitectónicos y atributos de calidad

<b>Patrón Arquitectónico</b>	<b>Descripción</b>	<b>Atributos Asociados</b>	<b>Atributos en conflicto</b>
<b>Layers</b>	Consiste en estructurar aplicaciones que pueden ser descompuestas en grupos de subtarear, las cuales se clasifican de acuerdo a un nivel particular de abstracción.	Reusabilidad Portabilidad Facilidad de Prueba	Desempeño Mantenibilidad
<b>Pipes and Filters</b>	Provee una estructura para los sistemas que procesan un flujo de datos. Cada paso de procesamiento está encapsulado en un componente filtro (filter). El dato pasa a través de conexiones (pipes), entre filtros adyacentes.	Reusabilidad Mantenibilidad	Desempeño
<b>Blackboard</b>	Aplica para problemas cuya solución utiliza estrategias no determinísticas. Varios subsistemas ensamblan su conocimiento para construir una posible solución parcial ó aproximada.	Modificabilidad Mantenibilidad Reusabilidad Integridad	Desempeño Facilidad de Prueba
<b>Broker</b>	Puede ser usado para estructurar sistemas de software distribuido con componentes desacoplados que interactúan por invocaciones a servicios remotos. Un componente broker es responsable de coordinar la comunicación, como el reenvío de solicitudes, así como también la transmisión de resultados y excepciones.	Modificabilidad Portabilidad Reusabilidad Escalabilidad Interoperabilidad	Desempeño
<b>Model-View-Controller</b>	Divide una aplicación interactiva en tres componentes. El modelo (model) contiene la información central y los datos. Las vistas (view) despliegan información al usuario. Los controladores (controllers) capturan la entrada del usuario. Las vistas y los controladores constituyen la interfaz del usuario	Funcionalidad Mantenibilidad	Desempeño Portabilidad

Fuente: Buschmann et al. (1996), Oriented Software Architecture. A System of Patterns.

En el presente proyecto de fin de titulación se implementará el patrón arquitectónico (MVC) Modelo – Vista - Controlador, siendo uno de los más robustos con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y separación de conceptos.

El fundamento principal es la separación de conceptos en tres capas diferentes, acotadas por su responsabilidad en lo que se llaman Modelos, Vistas y Controladores. Al utilizar este patrón arquitectónico, se crearán procesos que aseguren la calidad en los programas y ésta calidad atiende a diversos parámetros que son deseables para todo el desarrollo como la estructuración de los programas o reutilización del código lo cual influye positivamente en la facilidad de desarrollo y mantenibilidad del software. También es utilizado por múltiples plataformas o lenguajes de programación, dentro de este proyecto se utilizará el lenguaje PHP y CodeIgniter como framework de desarrollo.

### 1.4.3 Patrón Arquitectónico Modelo Vista Controlador

Este patrón se especifica bajo la preposición de dividir la aplicación en tres tipos de elementos; Modelo, Vista y Controlador según la Universidad de Alicante (2015):

- **El Modelo:** contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- **La Vista,** o interfaz de usuario que se compone de la información enviada al cliente y los mecanismos de interacción.
- **El Controlador,** que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

En aplicaciones web la vista se compone del HTML y el código que provee datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la lógica del negocio es en el controlador, responsable de recibir los eventos de entrada desde la vista.

#### **Ventajas.**

Al existir la separación de Vistas, Controladores y Modelos es más sencillo realizar labores de mejora según la página web EcuRed (2015):

- Agregar nuevas vistas
- Agregar nuevas formas de recolectar las órdenes del usuario (Interpretar sus modelos mentales).
- Modificar los objetos de negocios bien sea para mejorar el performance o para migrar a otra tecnología.
- Las labores de mantenimiento también se simplifican y se reduce el tiempo necesario, para ellas, por ende las correcciones solo se deben hacer en un solo lugar y no en varios como sucedería si tuviese una mezcla de presentación e implementación de la lógica del negocio.

- Las vistas también son susceptibles de modificación sin necesidad de provocar que todo el sistema se paralice.

#### 1.4.4 Estilos de la Arquitectura del Software

Cada estilo arquitectónico describe una categoría del sistema que contiene: un conjunto de componentes que realiza una función requerida por el sistema, un conjunto de conectores que posibilitan la comunicación, la coordinación y la cooperación entre los componentes, restricciones que definen como se pueden integrar los componentes que forman parte del sistema. El uso de un estilo arquitectónico permite definir los posibles patrones de una aplicación, además de permitir evaluar arquitecturas alternativas con ventajas y desventajas conocidas ante diferentes conjuntos de requerimientos no funcionales.

A continuación, en la Tabla 4 se describen algunos de los principales estilos arquitectónicos:

Tabla 4. Cuadro comparativo de los estilos arquitectónicos

Estilo Arquitectónico	Descripción
<b>Monolítica</b>	Un modelo en el que el sistema operativo y todos sus servicios residen en un monitor monolítico, es decir un servidor que guarda todo.
<b>Cliente / servidor</b>	Esta se reparte en dos partes distintas sin tener en cuenta la función de cada cual. En esta Arquitectura se puede ver que el servidor brinda la información, y el cliente es quien la recibe.
<b>Tres niveles</b>	Es un tipo de cliente/servidor, consta de tres capas las cuales son: capa de usuario, o modelado del negocio, y capa de almacenamiento o persistencia, este modelo permite que una capa tenga relación solo con otra.
<b>Pipeline</b>	Conocida como tuberías o basadas en filtros, es una cadena de tal manera que la salida de un elemento es la entrada del siguiente es muy común en el desarrollo de programas para poder interpretar comandos.
<b>Orientada a servicios</b>	Como su nombre lo dice, está orientada a servicios ya que se pueden acceder a sus funciones por medio de la red. Es una integración de aplicaciones independientes.
<b>Dirigida por eventos</b>	La estructura como su funcionamiento van ligados a sucesos que pueden ocurrir en los sistemas, promueve la producción detección y consumo en base a eventos.

Fuente: (Camacho, Cardeso, & Nuñez, 2004). Patrones Arquitectónicos.

#### 1.5 Sistema Gestor de Base de Datos (SGBD)

Según Sarria (2006) “Consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a los mismos”. “Los SGBD son una herramienta que permiten a múltiples usuarios acceder a los datos al mismo tiempo y brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen “ (Universidad Técnica Federico Santa María, 2014)

Un SGBD debe permitir según la página web EcuRed (2015):

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.

En la siguiente Tabla 5 se presenta un cuadro comparativo sobre las bases de datos más utilizadas por los desarrolladores.

Tabla 5. Cuadro comparativo de los Gestores de Base de Datos

Sistema Gestor de Base de Datos	Características
<b>PostgreSQL</b>	Alta concurrencia: permite a un proceso escribir en una tabla mientras otros accedan sin necesidad de bloqueos. Los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables debido a la infraestructura GIST. Disparadores triggers, vistas, integridad transaccional, herencia de tablas. Completo lenguaje para la escritura de disparadores y procedimientos almacenados.
<b>Firebird SQLite</b>	Arquitectura Cliente/Servidor sobre protocolo TCP/IP. Soporte de transacciones ACID y claves foráneas, es medianamente estable, Buena seguridad en usuarios y roles. Estándar SQL-92, transacciones de base de datos autónomas. Aislamiento, durabilidad, triggers, la mayor parte de consultas complejas. Tipos de datos inusuales: los tipos se asignan a los valores individuales.
<b>Apache Derby</b>	Apis para JDBC y SQL, soporta cifrado completo, roles y permiso. Posee SQL SCHEMAS, para separar la información en una única base de datos. Soporta internamente procedures, cifrado y compresión, soporte multilenguaje y localizaciones específicas. Transacciones y recuperación ante errores ACID.
<b>MySQL</b>	Integridad referencial, simplicidad en las consultas. Amplio subconjunto del lenguaje SQL. Disponibilidad en gran cantidad de plataformas y sistemas. Posibilidad de selección de mecanismos de almacenamiento que ofrecen velocidad de operación, soporte físico, capacidad, distribución, geografía y transacciones.
<b>Oracle</b>	Soporta todas las funciones que se esperan de un servidor (PL/SQL). Triggers, procedimientos Almacenados, integridad referencial declarativa bastante potente.

Fuente: (Centro de Desarrollo Territorial Holguín, 2015)

En este proyecto se utilizará como motor de base de datos PostgreSQL siendo una de las bases de datos más potente y robustas en el mercado. Posee características tales como estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. Funciona con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. También se lo utilizó de acuerdo a la bases de datos que posee el Gobierno Provincial de Loja.

### 1.5.1 POSTGRESQL

En el Sistema de Gestión Vehicular desarrollado para el Gobierno Provincial de Loja integra el Sistema de Gestor de Base de Datos PostgreSQL, para el almacenamiento y procesamiento de información recurrente por las siguientes razones según la página web EcuRed (2015):

- Posee características significativas como: subconsultas, los valores por defecto, las restricciones a valores en los campos (constraints) y los disparadores (triggers).

- Ofrece funcionalidades con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Está disponible para 34 plataformas con la última versión estable. Es totalmente compatible con ACID (acrónimo de Atomicity, Consistency, Isolation and Durability; en español: Atomicidad, Consistencia, Aislamiento y Durabilidad).
- Posee una integridad referencial e interfaces nativas para lenguajes como ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG; PYTHON y RUBY.
- PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico.

### **1.5.2 Procedimientos Almacenados.**

Se implementa en el sistema de gestor de base de datos PostgreSQL los procedimientos almacenados o propiamente dicho funciones denominados por el PL/pgSQL, que son porciones de código que pueden ser ejecutados exclusivamente en el servidor y que pueden tener parámetros de entrada y de salida.

Ventajas de los Procedimientos Almacenados:

- La ejecución del procedimiento ocurre en el servidor de bases de datos. Esto probablemente aumentará el rendimiento de la aplicación al no tenerse que enviar datos entre el cliente y el servidor, y no tener que procesar resultados intermedios en el cliente para obtener el resultado final.
- Al tener la lógica de la aplicación implementada en la base de datos no tendremos que implantarla en los clientes, con el consiguiente ahorro de líneas de código redundante y complejidad.
- Para maximizar la seguridad de los datos e información almacenada, se combate el ataque de inyecciones SQL.
- Se reducen las inconsistencias del código y se permite que cualquier usuario o aplicación que cuente con los permisos necesarios pueda acceder al código y ejecutarlo. (Martinez R., 2009)

Tipos de Procedimientos Almacenados según la compañía de Microsoft (2014):

- Definidos por los usuarios, se puede crear en una base de datos definida por el usuario o en todas la bases de datos del sistema excepto en la base de datos Resource.
- Temporales, se almacenan en el temp db y existen dos tipos de procedimientos temporales los locales y globales:

- Locales: solo son visibles en la conexión actual del usuario y se eliminan cuando se cierra la conexión.
- Globales: Son visibles para cualquier usuario después de su creación y se eliminan al final de la última sesión en la que se usa el procedimiento.
- Sistema, están almacenados físicamente en la base de datos interna y oculta Resource y se muestra de forma lógica en el esquema sys de cada base de datos definida por el sistema y por el usuario, que proporcionan una interfaz de SQL Server.
- Extendidos definidos por el usuario, permiten crear rutinas externas, estos procedimientos son archivos DLL que una instancia de SQL Server puede cargar y ejecutar dinámicamente.

Para los procedimientos almacenados se utiliza el lenguaje PL/pgSQL ya que su estructura se emplea mediante bloques. Los objetivos por los cuales las funciones se crean con dicho lenguaje son las siguientes:

- Para crear funciones y disparadores (triggers).
- Añadir estructuras de control al lenguaje SQL.
- Realizar cálculos complejos.
- Heredar todos los tipos, funciones y operadores definidos por el usuario.
- Fácil de usar.
- Paginación de los resultados.
- Limitación de datos.
- Menor redundancia en los resultados. (Martinez R., 2009)

## **1.6 Tecnología**

### **1.6.1 Leguajes para desarrollar aplicaciones web.**

Existen en la actualidad diferentes tecnologías con un mismo propósito: proporcionar contenido dinámico en la web.

Todas estas tecnologías se basan en:

- Scripts que se ejecutan en el servidor web.
- Pueden recibir información del usuario.
- Producen una página web dinámica.
- Los scripts se encuentran incrustados en el código de una página HTML. (Departamento de Lenguajes y Sistemas Informaticos, 2002).

El siguiente Tabla 6 describe las ventajas y desventajas de los lenguajes de programación de aplicaciones web:

Tabla 6. Cuadro comparativo de los lenguajes de programación por guiones

Tecnología	Ventajas	Desventajas
<b>ASP (Active Server Pages)</b>	<ul style="list-style-type: none"> <li>• Sistema recomendado para programadores de Visual Basic.</li> <li>• Fácil de aprender.</li> <li>• Gratuito.</li> <li>• Cientos de componentes para cualquier funcionalidad requerida.</li> <li>• Variada información acerca de la tecnología.</li> </ul>	<ul style="list-style-type: none"> <li>• Actualmente se encuentra disponible para el sistema operativo de Microsoft.</li> <li>• La mayoría de los componentes no son gratuitos.</li> <li>• Los fallos tardan en depurarse.</li> </ul>
<b>CFM (Macromedia ColdFusion)</b>	<ul style="list-style-type: none"> <li>• Muy sencillo de usar ya que emplea etiquetas HTML.</li> <li>• Gran cantidad de etiquetas necesarias para crear una página web.</li> </ul>	<ul style="list-style-type: none"> <li>• Es difícil crear sentencias de control complejas.</li> <li>• Confuso en la manera de empleo de las etiquetas.</li> </ul>
<b>JSP (Sun JavaServer Pages)</b>	<ul style="list-style-type: none"> <li>• Sugerido para programadores en JAVA.</li> <li>• Multiplataforma.</li> <li>• Características: <ul style="list-style-type: none"> <li>○ Orientado a objetos.</li> <li>○ Robusto.</li> <li>○ Excelente gestión de errores.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Complejo para desarrolladores que desconozcan JAVA.</li> <li>• Poco práctico para pequeños proyectos.</li> <li>• Tiempos de desarrollo mayores a diferencia de otras tecnologías.</li> </ul>
<b>PHP (Hypertext Preprocessor)</b>	<ul style="list-style-type: none"> <li>• Gratuito.</li> <li>• Multiplataforma.</li> <li>• El más rápido.</li> <li>• Mejor manejo de memoria.</li> <li>• Fácil de aprender.</li> <li>• Se integra con APACHE (servidor web) y con todo tipo de Bases de Datos.</li> <li>• Amplio soporte en internet.</li> </ul>	<ul style="list-style-type: none"> <li>• El acceso a las bases de datos no está estandarizado, diferentes bases de datos diferentes sintaxis.</li> </ul>

Fuente: (Departamento de Lenguajes y Sistemas Informáticos, 2002)

En el presente proyecto se utilizará como lenguaje de desarrollo PHP junto con el framework CodeInteger, porque está orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de datos, otra característica por la cual utilizar es la capacidad de conexión a la mayoría de los motores de bases de datos que se utilizan en la actualidad, la programación es más segura y confiable de carácter multiplataforma.

### 1.6.1.1 PHP

Según Peinado (2010) “Es un lenguaje de guiones que se ejecuta en el servidor” y Romano, (2015) “Es un lenguaje de script del lado del servidor. Los scripts PHP están incrustados en los documentos HTML y el servidor los interpreta y ejecuta antes de servir las páginas al cliente.”

“Es un lenguaje interpretado con una sintaxis similar a C++ o JAVA. Aunque el lenguaje se puede usar para realizar cualquier tipo de programa, es en la generación dinámica de páginas web donde ha alcanzado su máxima popularidad.” (Duarte & Pérez, 2012).

PHP es un lenguaje de codificación con mayor usabilidad para aplicaciones web e interpretadas por el servidor de tal forma que todo lo codificado se ejecute con total normalidad. Los scripts o guiones incrustados en las etiquetas HTML logran una iteración correcta de las mismas, es utilizable y escalable para la obtención, almacenamiento, integridad y seguridad de la información, así como la dinamización de los aplicativos web.

### **1.6.1.2 Javascript**

Según Pérez (2008) “JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos”.

Según Sánchez (2003) “JavaScript es lo que se conoce como lenguaje script, es decir: se trata de código de programación que se inserta dentro de un documento. JavaScript nace con la idea de potenciar la creación de páginas Web dinámicas para los navegadores”.

Según Navarrete (s.f.) “JavaScript es un sencillo lenguaje de programación, que presenta una característica especial: sus programas, llamados comúnmente scripts, se encuentran en las páginas HTML y se ejecutan en el navegador”.

JavaScript es un lenguaje de programación de scripts los que son invocados en un determinado evento para dinamizar las páginas web, los scripts no solo funcionan con invocaciones, sino al iniciar una determinada sesión. Adicionalmente sirven para concatenar datos, habilitar o deshabilitar campos, limpiar datos y pasó de información transformándolos en archivos seguros.

### **1.6.1.3 CSS1 (Cascade Style Sheets – version 1).**

Según Sánchez (2013) “son hojas de estilo en cascada. Esta tecnología pretende eliminar los problemas que tiene el HTML. HTML tiene el problema de que es difícil controlar exactamente el formato de las páginas Web. Con CSS se pretende crear documentos que especifiquen claramente cómo deben actuar las etiquetas de HTML”

Según Peinado (2014) “Las hojas de estilo en cascada es una forma de asociar información de presentación a documentos HTML y XML. Permite definir estilos para la presentación de los distintos elementos de un documento.”

Según Pérez (2015) “Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor

forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.”

La razón de integrar CSS es para organizar la presentación y aspecto de una página web. Otra razón para utilizar CSS es que se basa en separar lo que es la estructura del documento HTML de su presentación.

### 1.6.2 Framework CodeInteger

En la actualidad existen muchos framework para el desarrollo de aplicaciones web en PHP que trabajan con el **modelo vista controlador** entre las cuales está **CodeInteger**, que permite minimizar la cantidad de código y tiempo de entrega de proyectos.

“Es un framework PHP de gran alcance, construida para codificadores PHP que necesitan una herramienta simple y elegante para crear aplicaciones web con todas la funciones. Su objetivo principal es permitir a los desarrolladores que puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas” (Luna & Freire, 2014)

Para el desarrollo del proyecto se ocuparon las siguientes características:

- Uso del patrón de Diseño Modelo – Vista – Controlador
- Active Record para Base de Datos
- Manejo de Sesión
- Encriptación de Datos
- Seguridad y Filtro de XSS

Se puede evidenciar que CodeInteger es exclusivamente para proyectos pequeños y manejables, por lo que se ajusta a nuestro proyecto.

A continuación se muestra en la Figura 2 el funcionamiento de CODEIGNITER aplicando el patrón arquitectónico Modelo – Vista –Controlador.

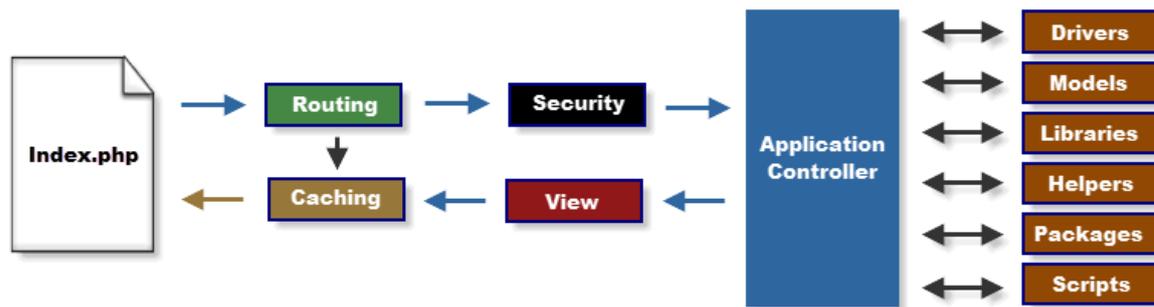


Figura 2. Funcionamiento de CodeIgniter  
 Fuente: (Alvarez M., 2009)

### 1.6.3 Framework de estilos Bootstrap.

El objetivo de un framework CSS es ahorrar las tareas básicas al emplear hojas de estilo, normalmente los framework CSS se componen de uno o varios archivos.

Según Mestras (2013) “Framework de twitter para desarrollo de aplicaciones web, es sencillo y ligero de usar puede bastar con un fichero CSS y una JavaScript, basado en los últimos estándares de desarrollo web.”

Según Ledesma (2008) “Es un método general a partir del cual pueden cubrirse diferentes objetivos de análisis de datos, incluyendo la estimación de intervalos de confianza (IC) o el Test de significación estadística.”

Bootstrap es un framework que nos permite hacer uso de los CSS ya normalizados para las páginas web, ahorrando tiempo y código, convirtiéndose en un framework interactivo y fácil de usar, nos permite hacer uso de sus estilos tan solo llamándolos en los encabezados.

**CAPITULO II**  
**DEFINICIÓN DEL PROBLEMA Y ANÁLISIS INICIAL**

## **2.1 Introducción**

La utilización de técnicas de levantamiento de información, políticas y análisis de los flujos de los procesos de la Institución se emplean para la obtención de los requerimientos funcionales y no funcionales del sistema, aquellas técnicas son; la entrevista y la observación que son usadas para conocer a fondo los procesos que se llevan a cabo por parte de la Coordinación de Gestión Vehicular para la creación, control y emisión de ordenes y así plantear la posible solución para el desarrollo e implementación del Sistema de Gestión Vehicular que cumpla los requerimientos determinados durante el levantamiento e indagación de la información recolectados, ordenados y analizados.

En esta sección se describen políticas, normas y procesos que se emplean al control de los bienes públicos, control de combustibles, emisión de ordenes como son: las de movilización, mantenimientos, combustibles y repuestos para los vehículos del Gobierno Provincial de Loja. Además se incluye la problemática existente y las posibles soluciones.

## **2.2 Políticas**

En este apartado se detallan las políticas que rigen a la institución como entidad pública, las políticas que se presentan en el Anexo B determinan la manera correcta en que se realiza la movilidad y el mantenimiento preventivo o correctivo de los vehículos.

Las políticas influyen en el debido funcionamiento del sistema ya que ningún vehículo debe movilizarse y repararse sin portar con las órdenes de movilización y mantenimiento. Por tal razones el sistema integra en sus módulos la creación y la emisión de dichos documentos, así como también la obtención de reportes de todos los movimientos del sistema ya que la Contraloría General de Estado exige entregarlos como medida de control del parque automotor de las entidades públicas.

## **2.3 Proceso del Negocio**

La presente sección detalla los procesos para emitir ordenes de movilización, viáticos, combustibles, mantenimientos, lubricantes y/o repuestos que se presentan en la Tabla 7, 8, 9, 10 y 11. Además se visualiza los flujos de los procesos para emitir dichas ordenes en la Figura 3, 4, 5, 6 y 7.

### **2.3.1 Descripción del proceso para la entrega de la Orden Movilización.**

En la Tabla 7 se describe el proceso para la entrega de las ordenes de movilización.

Tabla 7. Descripción del proceso para la entrega de Orden de Movilización.

<b>ORDEN MOVILIZACIÓN</b>					
PRN 01					
DESCRIPCIÓN					
<p>El proceso permite a los empleados del GPL crear solicitudes de movilización para los vehículos, tal solicitud debe incluir en sus campos los datos personales de los solicitantes, motivo, número de personas, datos del vehículo, razones de la solicitud, ruta, fecha y hora de salida, fecha y hora de llegada. La solicitud es aprobada por el coordinador de movilización, y asigna el vehículo y conductor disponibles para determinada fecha, emite la orden de movilización, previamente elaborada en el software de la contraloría, y después registrándola en un formato Excel.</p>					
ENTRADAS					
<ul style="list-style-type: none"> <li>• FECHA DE LA SOLICITUD</li> <li>• FECHA DE SALIDA</li> <li>• FECHA DE LLEGADA</li> <li>• HORA DE SALIDA</li> <li>• HORA DE LLEGADA</li> <li>• MOTIVO</li> <li>• RUTA</li> <li>• NUMERO DE PERSONA</li> <li>• NOMBRES DE LAS PERSONAS QUE VIAJAN</li> </ul>					
ACTIVIDADES					
No	Nombre	Descripción	Roles participantes	Rol responsable	Resultado
1	Crear Solicitud de Movilización	El solicitante cuando requiera de un vehículo, entrega una solicitud con los datos de entrada anteriormente descritos	Solicitante	Empleado Prefectura	Solicitud de Movilización.
2	Aprobación	Se aprueba o se niega la solicitud según criterios de la Coordinación de Gestión Vehicular. Donde se procesan datos adicionales extraídas de bases de datos externas con los datos registrados de los vehículos como son: kilometraje de inicio, número de placa, marca/modelo, color y número de matrícula.	Coordinación de Gestión Vehicular	Coordinación de Gestión Vehicular	Firma de aprobación.
3	Asignación Vehículo/chofer	Se asigna el vehículo y el conductor, en caso de no estar disponible el conductor se puede escoger otro y asignarlo a la unidad escogida.	Coordinación de Gestión Vehicular	Coordinación de Gestión Vehicular	Orden de Movilización.
SALIDAS					
<ul style="list-style-type: none"> <li>• MOTIVO</li> <li>• No. OCUPANTES</li> <li>• FECHA DE SALIDA</li> <li>• FECHA DE LLEGADA</li> <li>• HORA DE SALIDA</li> <li>• HORA DE LLEGADA</li> <li>• RUTA</li> <li>• KILOMETRAJE DE INCIO</li> <li>• CONDUCTOR:                             <ul style="list-style-type: none"> <li>○ Nombres y apellidos</li> <li>○ Cedula</li> <li>○ Teléfono</li> </ul> </li> <li>○ Puesto que ocupa</li> <li>○ Nombre de la unidad a la que pertenece.</li> <li>• VEHÍCULO:                             <ul style="list-style-type: none"> <li>○ Número de Placa</li> <li>○ Marca/Modelo</li> <li>○ Color</li> <li>○ Numero matricula</li> </ul> </li> <li>• NOMBRE DEL SERVIDOR SOLICITANTE</li> <li>• DEPARTAMENTO</li> <li>• FECHA DE LA SOLICITUD</li> </ul>					

Elaboración: (Ordóñez & Guaman, 2015)

En la Figura 3 se describe el flujo de proceso que se lleva para entregar la solicitud de movilización entre los actores involucrados se encuentra el empleado, y el coordinador de Gestión de vehicular.

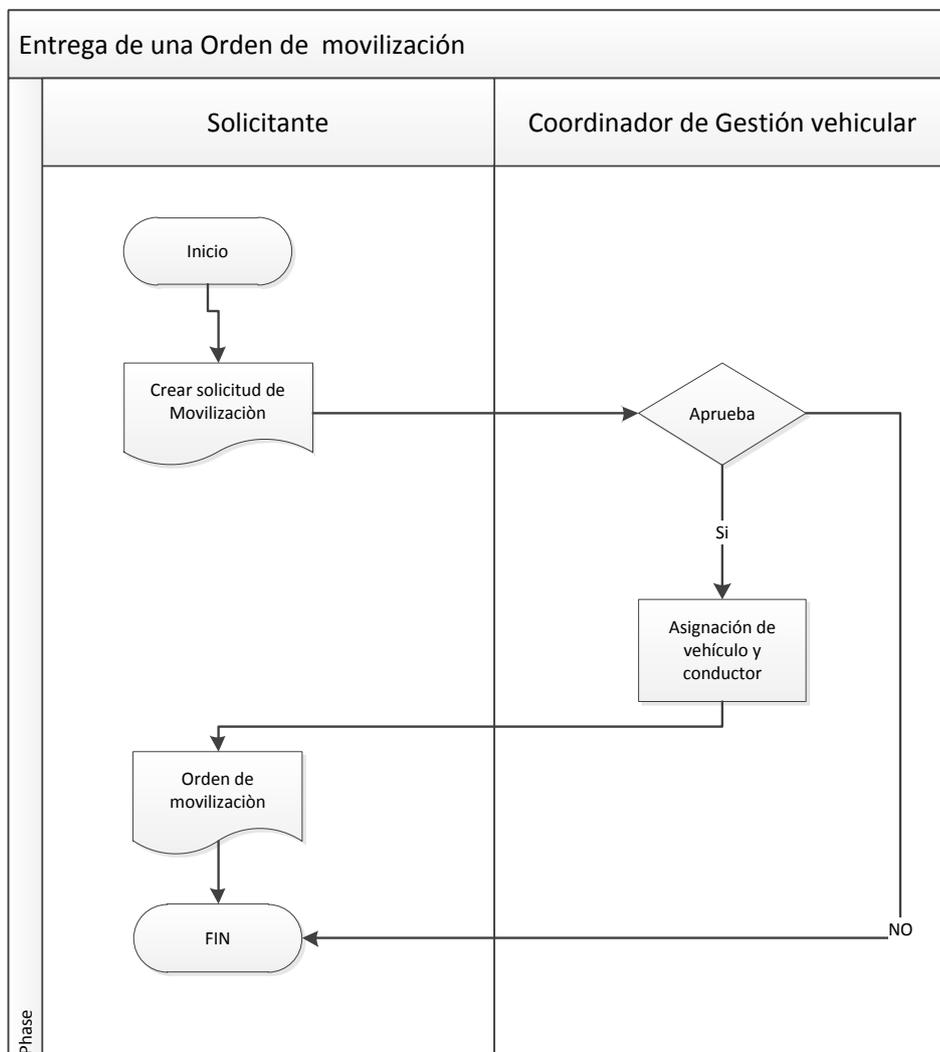


Figura 3. Flujo del proceso de entrega de la Orden de Movilización  
Elaboración: (Ordóñez & Guaman, 2015)

### 2.3.2 Descripción del proceso de entrega de Orden de Viáticos

En la Tabla 8 se describe el proceso para la entrega de una orden viáticos generada a partir del registro de datos de la orden de movilización. En esta se incluyen los mismos datos generados anteriormente de la solicitud de movilización, esta orden va dirigida directamente al chofer que va a salir a cumplir dicha comisión.

Tabla 8. Descripción del Proceso de la emisión de la Orden de Viáticos.

<b>ORDEN DE VIÁTICOS</b>					
PRN 02					
DESCRIPCIÓN					
<p>Una vez registrada y emitida la orden de movilización al solicitante, se entrega una orden de viáticos al conductor que ha sido asignado, tal orden tiene el formato determinado por la contraloría donde se incluyen datos del vehículo y chofer para la entrega de viáticos.</p>					
ENTRADAS					
<ul style="list-style-type: none"> <li>• NOMBRE DEL CONDUCTOR</li> </ul>					
ACTIVIDADES					
No.	Nombre	Descripción	Roles participantes	Rol responsable	Resultado
1	Registrar Conductor	Se registran los datos del conductor, y se procede a entregarla al departamento financiero para el depósito de viáticos.	Conductor	Coordinación de Gestión Vehicular.	Orden de Movilización
SALIDAS					
<ul style="list-style-type: none"> <li>• FECHA DE LA SOLICITUD</li> <li>• NÚMERO DE LA SOLICITUD PARA CUMPLIMIENTO DE SERVICIOS INSTITUCIONALES.</li> <li>• CONDUCTOR: <ul style="list-style-type: none"> <li>• Nombres y apellidos</li> <li>• Cedula</li> <li>• Teléfono</li> <li>• Puesto que ocupa</li> <li>• Nombre de la unidad a la que pertenece.</li> </ul> </li> <li>• FECHA DE SALIDA</li> <li>• FECHA DE LLEGADA</li> <li>• HORA DE SALIDA</li> <li>• HORA DE LLEGADA</li> <li>• VEHICULO: <ul style="list-style-type: none"> <li>○ Marca Modelo del vehículo</li> <li>○ Número</li> </ul> </li> <li>• MOTIVO</li> <li>• RUTA</li> <li>• TIPO DE TRANSPORTE</li> <li>• DATOS PARA LA TRANSFERENCIA: <ul style="list-style-type: none"> <li>○ Nombre del banco</li> <li>○ Tipo de cuenta</li> <li>○ Número de cuenta</li> </ul> </li> <li>• KILOMETRAJE DE SALIDA Y LLEGADA</li> <li>• NOMBRE DEL SERVIDOR SOLICITANTE</li> <li>• NOMBRE DEL RESPONSABLE DE LA UNIDAD SOLICITANTE</li> <li>• NOMBRE DE LA AUTORIDAD NOMINADA O SU DELEGADO</li> <li>• NOMBRE DE LOS SERVIDORES QUE INTEGRAN LOS SERVICIOS INSTITUCIONALES.</li> </ul>					

Elaboración: (Ordóñez & Guaman, 2015)

En el Figura 4 se describe el flujo de proceso para la entrega de una orden de viáticos entre los actores involucrados de la Coordinación de Gestión Vehicular y los conductores.

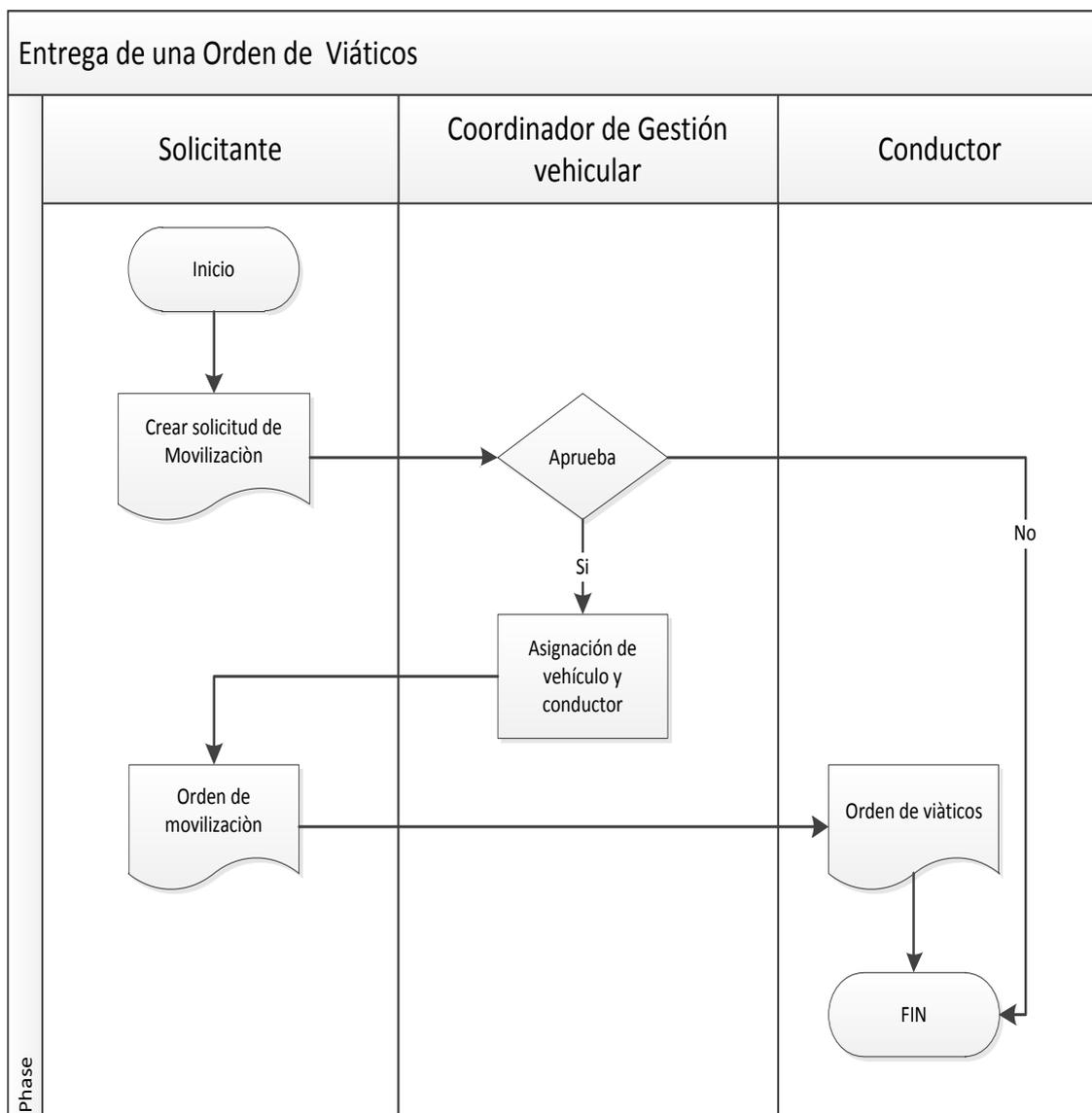


Figura 4. Flujo del proceso de entrega de la Orden de Viáticos.  
Elaboración: (Ordóñez & Guaman, 2015)

### 2.3.3 Descripción del proceso de asignación de Combustibles.

En la Tabla 9 se describe el proceso para generar una orden de combustibles, documento final que se entrega al chofer en la cual se detallan la gasolinera para despachar el combustible, los galones de combustibles asignados como datos de entrada, los datos de salida del vehículo y nombre del conductor.

Tabla 9. Descripción del proceso de Asignación de Combustible.

<b>ORDEN DE COMBUSTIBLE</b>						
PRN 03						
DESCRIPCIÓN						
<p>El proceso de emitir la orden de combustible, se realiza cuando existe la orden de movilización aprobada conjuntamente con la orden de viáticos, se da lugar a la emisión de la orden de combustibles donde se asigna el combustible requerido de acuerdo a las actividades que se realizarán.</p> <p>Este proceso acaba cuando el conductor entrega al coordinador de movilización un recibo de despacho del combustible, con la cantidad real asignada por la gasolinera, este valor es registrado.</p>						
ENTRADAS						
<ul style="list-style-type: none"> <li>• CANTIDAD DE GALONES DE COMBUSTIBLE</li> <li>• FECHA DE LA ORDEN</li> <li>• NOMBRE DEL CONDUTOR</li> <li>• NOMBRE DEL VEHICULO</li> <li>• PLACA DEL VEHICULO</li> <li>• GASOLINERA</li> <li>• KILOMETRAJE DE SALIDA</li> <li>• HORA DE SALIDA</li> <li>• HORA DE LLEGADA.</li> </ul>						
ACTIVIDADES						
No.	Nombre	Descripción	Roles participantes	Rol responsable	Resultado	
1	Crear Orden	Se registra la cantidad de galones de combustible, la fecha en que se crea la orden, nombre del conductor, nombre y placa del vehículo, el nombre de la gasolinera, kilometraje del vehículo con el que sale, hora de salida y llegada.	Conductor	Administración de Gestión Vehicular.	Orden de Combustibles	
2	Despacho	Se registra la cantidad asignada real de despacho de combustibles una vez receptado el ticket por el conductor.	Administración de Gestión Vehicular	Administración de Gestión Vehicular	Despacho	
SALIDAS						
<ul style="list-style-type: none"> <li>• FECHA DE LA ORDEN</li> <li>• PARTIDA DE COMBUSTIBLE</li> <li>• GASOLINERA</li> <li>• VEHÍCULO: <ul style="list-style-type: none"> <li>○ Nombre del vehículo</li> <li>○ Placa</li> <li>○ Número</li> <li>○ Kilometraje de salida y llegada</li> <li>○ Destino</li> </ul> </li> <li>• COMBUSTIBLE: <ul style="list-style-type: none"> <li>○ Tipo de combustible</li> <li>○ Cantidad de galones</li> <li>○ Cantidad de galones en letras</li> </ul> </li> </ul>						

Elaboración: (Ordóñez & Guaman, 2015)

En la Figura 5 se describe el flujo de proceso para la entrega de una orden de combustibles entre los actores involucrados que se encuentran en la Coordinación de Gestión Vehicular.

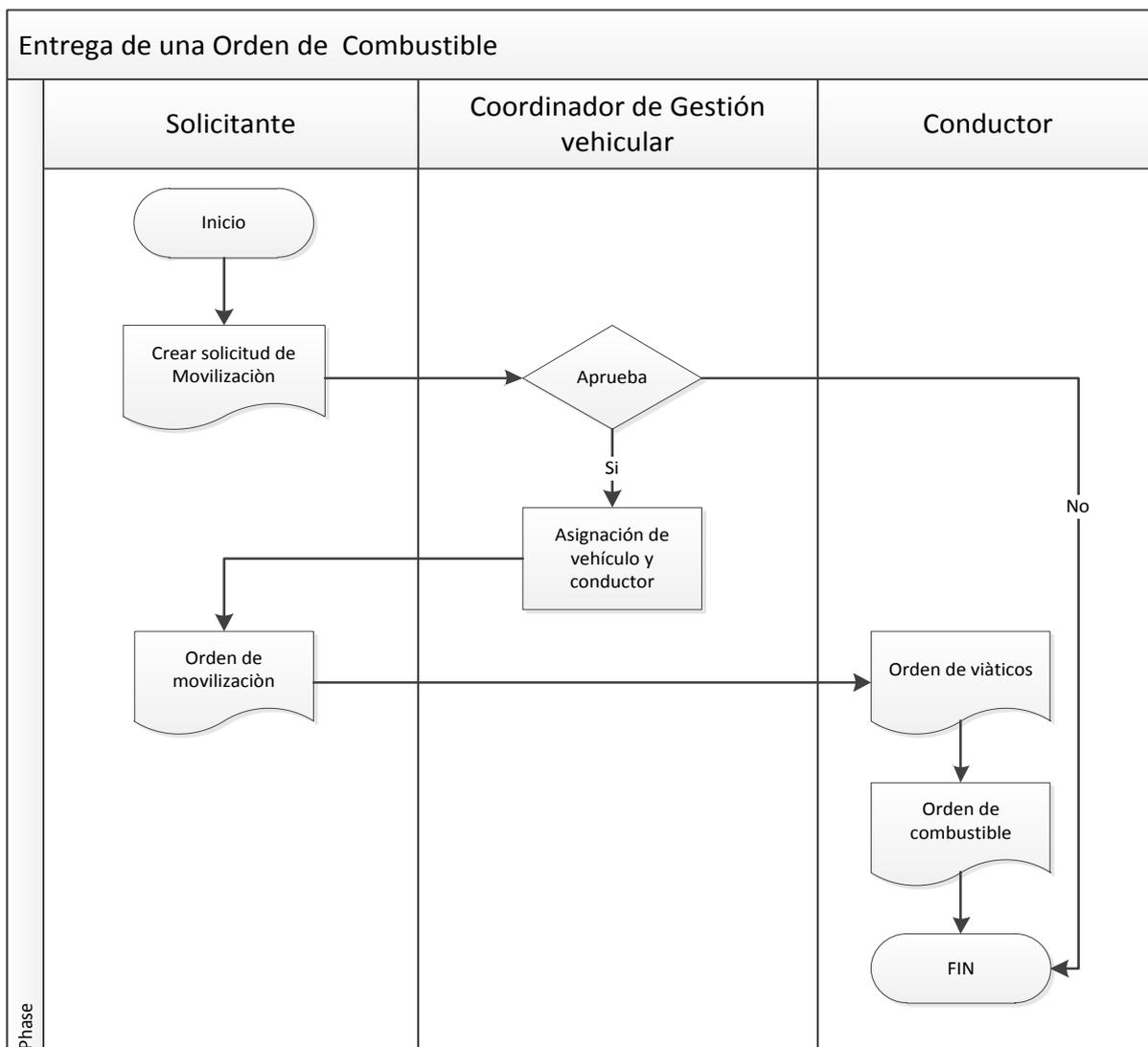


Figura 5. Flujo de proceso para la asignación de combustibles.  
Elaboración: (Ordóñez & Guaman, 2015)

### 2.3.4 Descripción del Proceso de para la Orden de Mantenimiento Correctivo

En la Tabla 10 se describe el proceso para emitir una orden de mantenimiento con los datos de entrada y salida que se generan a partir de una solicitud, los datos ingresados son características del vehículo, kilometraje, detalles de desperfectos, trabajo realizado y fecha de emisión.

Tabla 10. Descripción del proceso Orden de Mantenimiento Correctivo.

<b>ORDEN MANTENIMIENTO</b>					
PRN 04					
DESCRIPCIÓN					
El proceso de mantenimiento correctivo, parte de la solicitud de mantenimiento donde informe las novedades y defectos del vehículo. Para la emisión de la orden se incluye la lista de repuestos o lubricantes que se deben adquirir, que tipo de mantenimiento necesita, previamente chequeado por la coordinación de talleres.					
ENTRADAS					
<ul style="list-style-type: none"> <li>• LUGAR Y FECHA</li> <li>• UNIDAD SOLICITANTE</li> <li>• VEHÍCULO:                             <ul style="list-style-type: none"> <li>○ MARCA</li> <li>○ TIPO</li> <li>○ MODELO</li> <li>○ NÚMERO DEL MOTOR</li> <li>○ NÚMERO DE CHASIS</li> <li>○ PLACA</li> <li>○ AÑO</li> <li>○ NÚMERO DE GPL</li> </ul> </li> <li>• DETALLE DE DESPERFECTOS</li> </ul>					
ACTIVIDADES					
No	Nombre	Descripción	Roles participantes	Rol responsable	Resultado
1	Crear Solicitud	El conductor informa novedades y desperfectos del vehículo al coordinador de mantenimiento/ talleres y a su vez se registra	Solicitante	Coordinador de Mantenimiento	Solicitud de Mantenimiento
2	Verificación del estado del vehículo	El coordinar chequea el vehículo y decide si se aprueba o no la solicitud de mantenimiento, si se aprueba emite una orden de lubricantes y repuestos.	Solicitante	Coordinador de Mantenimiento	Firma de autorización.
3	Emitir Orden de Mantenimiento	La emisión de la orden de mantenimiento se especifica los datos registrados en la solicitud de mantenimiento.	Coordinador de Mantenimiento	Coordinador de Mantenimiento	Orden de Mantenimiento
SALIDAS					
<ul style="list-style-type: none"> <li style="width: 50%;">• NÚMERO DE SOLICITUD DE MANTENIMIENTO</li> <li style="width: 50%;">• FECHA DE ENTREGA</li> <li style="width: 50%;">• LUGAR Y FECHA</li> <li style="width: 50%;">• HORA DE ENTREGA</li> <li style="width: 50%;">• UNIDAD SOLICITANTE</li> <li style="width: 50%;">• NOMBRE DEL RESPONSABLE DEL TRABAJO</li> <li style="width: 50%;">• VEHÍCULO:                             <ul style="list-style-type: none"> <li>○ MARCA</li> <li>○ TIPO</li> <li>○ MODELO</li> <li>○ NÚMERO DEL MOTOR</li> <li>○ NÚMERO DE CHASIS</li> <li>○ PLACA</li> <li>○ AÑO</li> <li>○ NÚMERO DE GPL</li> </ul> </li> <li style="width: 50%;">• NOMBRE DEL CONDUCTOR</li> <li style="width: 50%;">• NOMBRE DEL JEFE DE MANTENIMIENTO</li> <li>• DETALLE DE DESPERFECTOS</li> <li>• SERVIDOR RESPONSABLE</li> <li>• FECHA DE RECEPCIÓN</li> <li>• TRABAJO REALIZADO</li> <li>• HORA DE RECEPCIÓN</li> </ul>					

Elaboración: (Ordóñez & Guaman, 2015)

En la Figura 6 se muestra el flujo de proceso para generar una orden de mantenimiento entre los actores involucrados se encuentra el conductor y el coordinador de talleres.

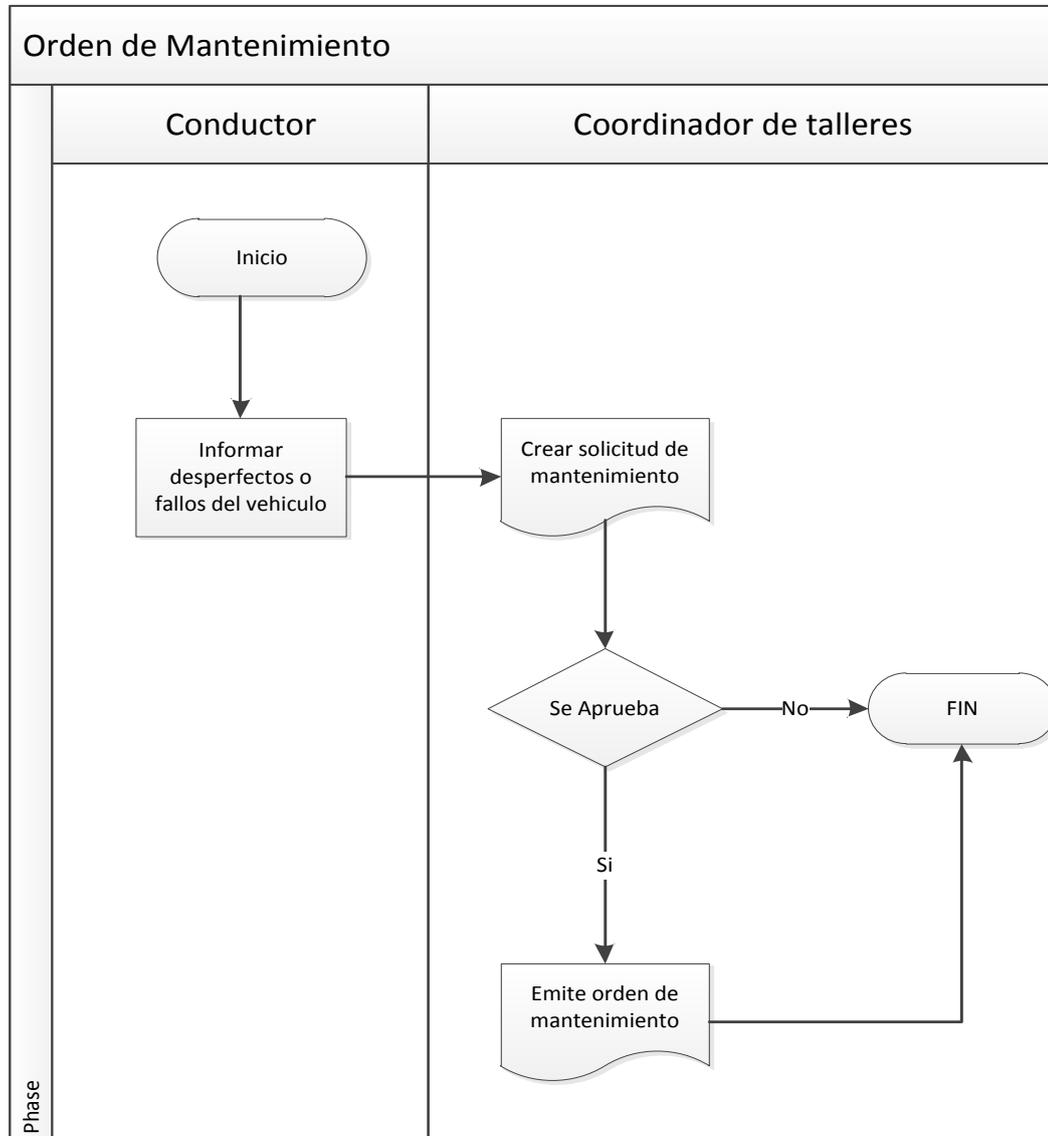


Figura 6. Flujo de proceso de la Orden de Mantenimiento Correctivo  
Elaboración: (Ordóñez & Guaman, 2015)

### 2.3.5 Descripción del proceso de Emisión de Lubricantes y Repuestos

En la Tabla 11 se describe el proceso para crear una orden de emisión de lubricantes y repuestos, esta se genera a partir de haber registrado una orden de mantenimiento.

Tabla 11. Descripción del proceso de Emisión de Lubricantes y Repuestos

<b>Lubricantes Y/O Repuestos</b>					
PRN 05					
DESCRIPCIÓN					
<p>El proceso permite crear la orden de lubricantes y repuestos, para emitir la orden primero debe aprobarse la orden de mantenimiento donde se incluye la cantidad, nombre o código de los repuestos o lubricantes necesarios para la reparación de los vehículos.</p>					
ENTRADAS					
<ul style="list-style-type: none"> <li>• FECHA DE LA ORDEN DE LUBRICANTES Y/O REPUESTOS</li> <li>• NOMBRE DEL DESPACHADOR DE REPUESTOS Y/O LUBRICANTES</li> <li>• NOMBRE DE LA PERSONA QUE RECIBE DICHOS REPUESTOS Y/O LUBRICANTES</li> <li>• CANTIDAD</li> </ul>					
ACTIVIDADES					
No.	Nombre	Descripción	Roles participantes	Rol responsable	Resultado
1	Crear Orden	Se registran los datos del despachador, los repuestos y lubricantes solicitados, la cantidad y concepto de los repuestos.	Administración de Mantenimiento	Jefe de Mantenimiento	Orden de Lubricantes y/o Repuestos.
2	Impresión de la Orden	Se agrega datos del vehículo y lugar del trabajo, el kilometraje y firmas de los responsables.	Administración de Mantenimiento	Jefe de Mantenimiento	Orden de Lubricantes y/o Repuestos.
SALIDAS					
<ul style="list-style-type: none"> <li>• FECHA DE LA ORDEN DE LUBRICANTES Y/O REPUESTOS</li> <li>• NOMBRE DEL DESPACHADOR DE REPUESTOS Y/O LUBRICANTES</li> <li>• NOMBRE DE LA PERSONA QUE RECIBE DICHOS REPUESTOS Y/O LUBRICANTES</li> <li>• CANTIDAD</li> <li>• CONCEPTO</li> <li>• NUMERO DE VEHICULO</li> <li>• LUGAR DE TRABAJO</li> <li>• KILOMETRAJE</li> <li>• NÚMERO DE PARTIDA</li> <li>• HORA DE TRABAJO</li> </ul>					

Elaboración: (Ordóñez & Guaman, 2015)

En la Figura 7 se describe el flujo de proceso para la entrega de un orden de repuestos. Entre los actores involucrados se encuentra el coordinador de mantenimiento y el jefe de bodega.

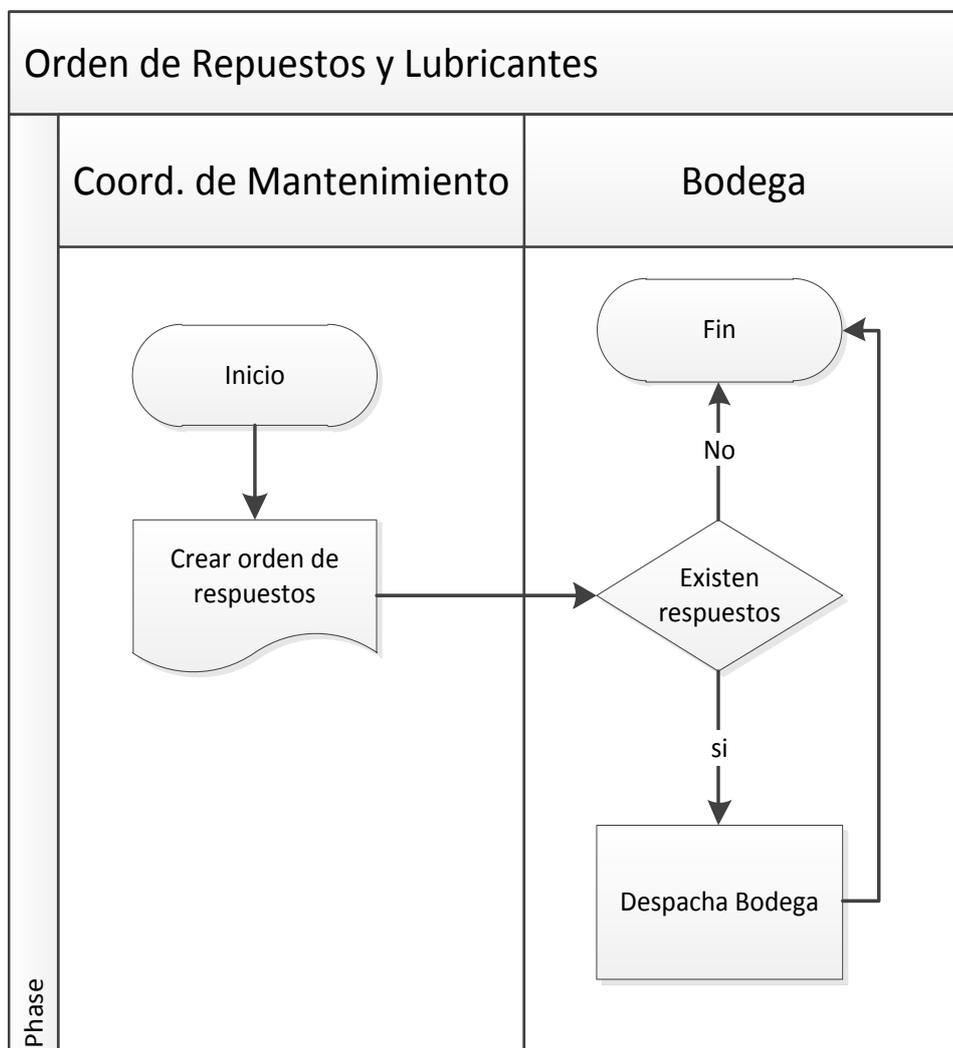


Figura 7. Flujo de proceso para emisión de Orden de Lubricantes y Repuestos. Elaboración: (Ordóñez & Guaman, 2015)

## 2.4 Problemática

Una vez detallados los procesos de la coordinación de gestión vehicular para la movilización y mantenimiento de los vehículos se procede a describir la problemática existente en el departamento para emisión de las órdenes de movilización, viáticos, combustibles, mantenimiento y los reportes exigidos por la Contraloría General del Estado.

En la Tabla 12 se describen los problemas existentes de a Coordinación de Gestión Vehicular del Gobierno Provincial de Loja.

Tabla 12. Problemas existentes dentro del GPL

Nº	PROBLEMA	DESCRIPCION DEL PROBLEMA
1	Generación y emisión de las órdenes de movilización incluyendo viáticos.	La emisión de las órdenes de movilización y viáticos se realiza de forma manual registrando los datos del solicitante en la plantilla de Excel, lo que conlleva un problema cuando se requiere emitir varias órdenes al día, ya que hacerlo manualmente ocupa demasiado tiempo y muchas de las veces los usuarios no se encuentran para dar la información correspondiente.
2	Generación y emisión de las órdenes de combustibles.	La generación de las órdenes de combustibles es igual que las anteriores descritas, se las emite de forma manual pero con un tiempo más de retraso ya que debe ser entregada para su revisión a otros departamentos y ser verificadas mediante llamadas telefónicas.
3	Generación y emisión de órdenes de mantenimiento.	Para emitir las ordenes de mantenimiento es necesario registrarlas de forma manual en un formato pre-impreso, y pasarlas a su aprobación a otros departamentos lo que conlleva más de una semana emitir dicha orden.
4	Generación y emisión de órdenes de repuestos.	La orden de repuestos se emite después de ser aprobada la orden de mantenimiento se procede a constatar el stock de los repuestos y lubricantes requeridos para el vehículo y luego se entrega la orden en un formato pre-impreso.
5	Exportar reportes requeridos por la Contraloría General del Estado.	Los reportes que requiere la Contraloría General del Estado son obligatorios ya que todas las entidades públicas deben entregarlos, los reportes se obtienen de plantillas Excel que previamente son llenados a manualmente después de crear una orden.
6	Asignación de galones de combustible para una ruta determinada.	El problema surge con el poco control en la asignación de combustibles, ya que se asigna dependiendo de lo que requiera el conductor.
7	Mantenimiento preventivo y correctivo a los vehículos.	El mantenimiento a los vehículos en cualquiera de las dos formas es de importancia ya que por ello el GPL determina dichos mantenimientos mediante la solicitud del conductor, sin tener en cuenta las anteriores reparaciones o cambios de repuestos, ni en qué tiempo se deben hacer los próximos.
8	Asignación vehículo/chofer.	El proceso de asignación de los conductores a los vehículos es de forma manual dependiendo de la disponibilidad de cualquiera de los recursos.
9	Alertas de caducidad de matrícula y SOAT vehicular.	Para determinar cuándo se debe renovar la matrícula vehicular mediante la alerta del conductor, el problema se encuentra en que muchas de las veces los conductores no informan o son removidos de los cargos, por lo que acarrea multas y retrasos con el canje de dichos documentos.

Elaboración: (Ordóñez & Guaman, 2015)

## 2.5 Planteamiento de la Solución

El presente proyecto de fin de titulación nace de la necesidad de automatizar los procesos de generación y emisión de las órdenes de movilización, viáticos, combustibles, mantenimiento, repuestos y la exportación de reportes listos para ser entregados a la Contraloría General del Estado, además del control de la asignación de combustibles y repuestos. Para esto se plantea una solución de software que consiste en el desarrollo e implementación de una aplicación web que permita el registro, verificación, emisión, exportación, asignación, alertas, control y administración de cuentas de usuario, de una forma segura y eficiente, por lo que los encargados del ingreso y emisión de dichas ordenes lo realizarían de manera automatizada, las asignaciones serán más rápidas de acuerdo a la disponibilidad de los recursos, la

automatización del control de combustibles, asignación de rutas dinámicamente, alertas necesarias para el canje de las matrículas vehicular y el manejo adecuado de los repuestos para los mantenimientos programados.

La posible solución contempla inicialmente el uso del patrón arquitectónico MVC (Modelo – Vista – Controlador), ya que permite disponer de una estructura ordenada y esquematizada para el acoplamiento de los módulos a integrar y componentes de un sistema informático, además de sus propiedades seguras para conectarse con los SGBD (Sistemas Gestores de Bases de Datos) y las relaciones entre ellos. De igual manera brinda mejores opciones para la seguridad de las aplicaciones a nivel de código y base de datos.

El presente proyecto contempla el uso de un nivel de seguridad aceptable, orden y reducción del código, evitar la redundancia de las conexiones, evitar el colapso de múltiples conexiones al mismo tiempo, autenticación segura, esquematización de la aplicación y la integración de los módulos que se describen a continuación:

- Administración: permite el registro de vehículos, conductores, tipo de licencias, tipo de combustibles, estaciones de servicios o gasolineras, y la asignación de vehículo/conductor, la creación de usuarios con sus respectivos roles, y la gestión de permisos para los usuarios de acuerdo al ejercicio de sus labores.
- Reportes: permite la visualización de información registrada y la búsqueda avanzada de los mismos, la exportación de los reportes de acuerdo a las búsquedas seleccionadas con los respectivos cálculos.

Los reportes integrados a este módulo son los siguientes: listado de vehículos, listado de conductores, control de mantenimiento, provisión de combustibles, matrícula, ordenes de movilización y novedades explícitamente solicitados por la Contraloría General del Estado.

- Gestión de Movilización: permite realizar solicitudes de movilización en el cual se escoge la ruta de forma dinámica, la aprobación o negación de las solicitudes, la emisión de la orden de movilización con los viáticos asignados, generar y emitir la orden de combustibles y el despacho de combustibles de acuerdo a la ruta que se solicitó.
- Talleres: permite realizar la solicitud de mantenimiento de acuerdo al cronograma programado de mantenimientos y el stock de los repuestos en bodega, visualizar el historial de reparaciones de los vehículos por cada uno, actualizar, borrar y crear nuevos mantenimientos de acuerdo a los kilometrajes o características de los vehículos.

## **2.6 Beneficios de la solución**

Con el desarrollo e implementación de la solución de software se beneficia directamente a la Coordinación de Gestión Vehicular en el que se encuentran los Departamentos de Gestión de Movilización y Talleres e indirectamente a bodega, ya que se podrá eliminar el tiempo empleado para el ingreso de datos de los conductores, la generación y emisión de órdenes y la exportación de reportes. Además se beneficia el GPL permitiéndose ahorrar recursos y automatizar los procesos de movilización y mantenimiento del parque automotor.

A continuación se presenta los beneficios de la solución de software:

- Minimizar el tiempo en el registro de datos de los empleados en las plantillas de Excel manualmente o reemplazando datos de anteriores registros.
- Reducir el tiempo utilizado en la impresión y entrega de las órdenes para movilizar los vehículos o dar mantenimiento de los mismos.
- Minimizar el tiempo de espera para la aprobación, negación o correcciones de las órdenes.
- Automatizar el proceso de asignación vehículo/chofer, eliminando el tiempo requerido para el proceso.
- Automatizar el control para un acertado 99.9% de la asignación del combustible para la movilización de los vehículos.
- Automatizar los mecanismos de alertas para la vigencia y caducidad de matrículas.
- Optimizar tiempos para generar reportes manualmente, de tal forma agilizar la exportación de los reportes con datos ya almacenados y cálculos específicos.
- Informatizar el proceso de solicitud y aprobación de la orden de movilización y viáticos, el proceso de emisión de la orden de combustibles, el proceso de solicitud y entrega de la orden de mantenimiento, y finalmente el proceso para la entrega de repuestos.
- Utilizar eficientemente los recursos.

## **2.7 Resultados Esperados**

A continuación se presenta los resultados esperados al momento de cumplir con los objetivos establecidos en el presente proyecto de tesis:

- Desarrollar e implementar una aplicación web que permita la gestión de la movilización, asignación, combustibles, y mantenimiento de los vehículos del parque automotor.
- Disponer de usabilidad de la aplicación web de tal manera que su ejecución sea una tarea sencilla y rápida de realizar.
- Gestión referenciada de rutas en el mapa de Google Maps, para determinar las rutas en la cual se obtendrá el kilometraje que recorrerá el vehículo.

- Disponibilidad de información oportuna para la exportación de reportes.
- Aplicación escalable y mantenible.
- Seguridad adecuada para el registro de información, peticiones, manejo de usuarios y claves que garanticen el normal funcionamiento de la aplicación web.
- Integridad en los datos almacenados y referenciados, extraíbles para las diferentes actividades.

**CAPITULO III**  
**DESARROLLO DEL SISTEMA DE GESTIÓN VEHICULAR**

### 3.1 Introducción

En esta sección se detalla la elaboración de las fases y artefactos de la metodología RUP para el desarrollo e implementación del sistema, como también la utilización de las tecnologías, herramientas, el patrón arquitectónico modelo-vista-controlador (MVC) y los procedimientos almacenados de la base de datos.

### 3.2 Metodología de desarrollo de software

El uso de la metodología RUP permite un desarrollo eficaz del sistema en cuanto a las fases de análisis y diseño.

### 3.3 Fases de la metodología

La metodología se caracteriza por ser un proceso iterativo e incremental, se incluyen artefactos para la documentación del software en cada una de las fases, las cuales se detallan a continuación en la Tabla 13:

Tabla 13. Fases de la metodología RUP y sus artefactos

FASE	HITO	ARTEFACTO
<b>Fase de Inicio</b>	En esta fase se identifica el alcance del proyecto, el tiempo establecido. Se desarrollarán los requisitos del producto, desde la perspectiva del usuario. Además de hacer un análisis comparativo de los requerimientos funcionales del actual sistema con el usuario.	<ul style="list-style-type: none"> <li>• Documento de visión</li> <li>• Documento de requerimientos</li> </ul>
<b>Fase de elaboración</b>	En esta fase se analizan los requisitos, se desarrollan los casos de uso y un prototipo de la arquitectura, mejorando la versión actual del sistema de gestión vehicular.	<ul style="list-style-type: none"> <li>• Documento de especificación de casos de Uso</li> <li>• Documento de arquitectura del sistema</li> </ul> <p><b>Vista l3gica</b>  <i>Diagrama de clases</i>  <i>Modelo E-R</i>  <b>Vista de Implementaci3n</b>  <i>Diagrama de secuencia</i>  <i>Diagrama de Despliegue</i></p>
<b>Fase de Construcci3n</b>	Se construye la soluci3n propuesta y se realizan las pruebas que garanticen todas las funcionalidades del sistema.	<ul style="list-style-type: none"> <li>• Desarrollo del software</li> </ul>
<b>Fase de Transici3n</b>	Se desarrolla el plan de pruebas en las que se incluyen pruebas unitarias, de rendimiento, funcionales, de sistema, usabilidad, seguridad y las de aceptaci3n del usuario.	<ul style="list-style-type: none"> <li>• Plan de pruebas</li> <li>• Manual de usuario</li> </ul>

Elaboraci3n: (Ord3n3z & Guaman, 2015)

### 3.4 Fase de Inicio del Sistema de gesti3n Vehicular

Se define el **documento de visi3n del proyecto** que tiene por objetivo la recolecci3n y an3lisis de las necesidades de los usuarios, y la definici3n de una perspectiva global de la soluci3n del software.

Se identifican los requerimientos funcionales y no funcionales del sistema detallados en el **documento de especificación de requerimientos**. En la fase se han implementado varias técnicas de recolección de datos tales como: entrevistas y la observación, que permiten entender y conocer las necesidades de los usuarios involucrados directamente e indirectamente con el proyecto, como se detalló en la segunda sección de la **definición del problema y estado actual**.

#### **3.4.1 Análisis del Estado Actual**

Los usuarios involucrados proporcionaron la aplicación inicial del CONGOPE y detallaron los procesos que se llevan a cabo, lo que ayudó a entender cómo se desarrolla cada una de las actividades dentro de la coordinación de gestión vehicular, se identificó los roles de los usuarios y las áreas involucradas.

Dentro de los aspectos que más se identificaron son los siguientes:

- Un empleado puede realizar una solicitud de movilización previa autorización de su delegado.
- Las órdenes de movilización son aprobadas por el departamento de movilización previa autorización de su máxima autoridad.
- La generación de reportes y formatos depende únicamente de la aprobación de la Contraloría General del Estado.
- Los reportes se generan a partir de la gestión de movilización y mantenimientos.

Luego de conocer a detalle los procesos actuales, se procede a realizar las entrevistas a los usuarios de la coordinación de gestión vehicular que se detallan en el **anexo A**.

#### **3.4.2 Entrevistas**

La coordinación de gestión vehicular proporcionó los formatos de las solicitudes y ordenes de movilización, viáticos, combustibles, mantenimiento y repuestos, además de las plantillas de los reportes, lo que permitió realizar la comparación entre la aplicación actual con la posible solución y determinar los módulos que se deben incluir.

Se investigó en los departamentos de movilización y talleres para obtener información relevante de los procesos y requerimientos para el sistema, las herramientas que utilizan, políticas y normas que rigen la movilidad y el mantenimiento de los vehículos. De tal forma se realizó el Documento de visión y el Documento de especificación de requerimientos.

### 3.4.3 Documento de Visión

El documento de visión consiste en el detalle del problema, la posible solución, análisis de las restricciones y la identificación de los participantes y roles de los usuarios para el sistema.

#### 3.4.3.1 Definición del Problema.

En la Tabla 14 se describe el problema actual y la posible solución para emitir las ordenes de movilización y mantenimiento.

Tabla 14. Definición del Problema

El problema de	<p>La emisión de las órdenes de movilización y de mantenimiento es de forma manual lo que representa pérdida de recursos como el tiempo para generar las órdenes.</p> <p>No se almacena la información para generar reportes para la Contraloría General del Estado.</p> <p>No existe el plan de mantenimiento preventivo para el cambio de las partes de los vehículos de forma oportuna para la toma de decisiones.</p>
afecta a	<p>Departamento de Gestión Vehicular, Departamento de Mantenimiento y Talleres, Empleados.</p>
El impacto asociado es	<p>No mantener orden en toda la documentación de los procesos de gestión de movilización y mantenimiento, por lo que evita desarrollar reportes.</p>
una adecuada solución sería	<p>Se desarrollara e implementara el Sistema Gestión Vehicular con los siguientes módulos: Gestión de Movilización, Talleres, Reportes y Administración, que nos permiten:</p> <ul style="list-style-type: none"> <li>• Disponer de la información en tiempo real y accesible desde cualquier lugar, para realizar un control y toma de decisiones acordes a la nueva regulación del control vehicular.</li> <li>• Automatizar el proceso actual del control de la flota de vehículos de la Institución.</li> </ul> <p>Mejorar los procesos de control de combustible, mantenimiento y reparación de vehículos y monitoreo de los mismos.</p>
Para	<p>Que los usuarios del departamento de movilización dispongan de la información en tiempo real para la asignación de un vehículo a una tarea.</p> <p>Los usuarios del departamento de talleres dispondrán de la información de cada vehículo, sus partes y repuestos, para determinar si se realiza un mantenimiento preventivo o correctivo.</p> <p>Los empleados podrán realizar una solicitud de movilización en cualquier momento y en el menor tiempo posible.</p>
Quienes	<p>Consumen la información y toman decisiones.</p>
El nombre del producto	<p>Sistema de Gestión Vehicular para el Gobierno Provincial de Loja</p>
Que	<p>Permite el registro de solicitudes y ordenes de movilización, mantenimiento, viáticos, combustibles, repuestos, la generación de reportes y la administración de la información.</p> <p>Valida la información ingresada y dan sugerencias.</p>
No como	<p>Registros de forma manual de las solicitudes y no llevan orden.</p>
Nuestro producto	<p>Permite registrar las solicitudes mediante la aplicación web.</p>

Elaboración: (Ordóñez & Guaman, 2015)

### 3.4.3.2 Roles de Usuario.

En la Tabla 15 se detallan los Stakeholders del Sistema de Gestión Vehicular con sus respectivos nombres, roles y la descripción del cargo.

Tabla 15. Roles de usuario

Nombre	Descripción	Stakeholder
<b>Jefa de la Coordinación de Gestión Vehicular</b>	Responsable de la Coordinación de Gestión Vehicular. Encargada de la aprobación de las solicitudes de movilización y mantenimiento.	Alba Barriga
<b>Administrador del Departamento de Movilización</b>	Responsable de emitir la orden de movilización, viáticos y combustible.	<b>Francisco Palacios</b>
<b>Mecánico del Departamento de Talleres</b>	Responsable de emitir la orden de mantenimiento, repuestos y lubricantes.	<b>Rubén Vicuña</b>
<b>Solicitante</b>	Solicitante de una orden de movilización y mantenimiento	<b>Empleados del Gobierno Provincial de Loja</b>

Elaboración: (Ordóñez & Guaman, 2015)

### 3.4.3.3 Participantes del Proyecto

En la Tabla 16 se describe los participantes directos del Sistema de Gestión Vehicular cuyas intervenciones influyen en el desarrollo e implementación del sistema.

Tabla 16. Participantes del proyecto

Nombre	Descripción	Responsabilidades
Ing. Pablo Vallejo	Representantes del Gobierno Provincial de Loja.	Seguimiento del desarrollo del proyecto.
Ing. Rubén Vicuña		Aprueban requisitos y funcionalidades

Elaboración: (Ordóñez & Guaman, 2015)

En el **anexo C** se encuentra más a detalle el **Documento de Visión**.

### 3.4.4 Especificación de Requerimientos

El documento de especificación de requerimientos tiene por objetivo definir los requerimientos funcionales y no funcionales que requiere el sistema. Los requerimientos funcionales describen la interacción con el sistema, los no funcionales se reflejan en la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

En la Tabla 17 se detallan los requerimientos funcionales del sistema y la prioridad de los mismos.

Tabla 17. Especificación de Requerimientos funcionales

Número	Requerimiento	Prioridad
RF1	Solicitud de movilización	5
RF2	Asignación Vehículo	5
RF3	Orden de Combustible	5
RF4	Despacho de Combustible	5
RF5	Solicitud de Mantenimiento	5
RF6	Historial vehículo Mantenimiento	5
RF7	Gestión del plan de mantenimiento preventivo	5
RF8	Gestión de Usuarios	5
RF9	Asignación vehículo Chofer	5
RF10	Gestión de Tipo Licencias	5
RF11	Gestión de combustible	5
RF12	Gestión de Gasolineras	5
RF13	Gestión de vehículos	5
RF14	Gestión de conductores	5
RF15	Reporte de listado de vehículos	5
RF16	Reporte de listado de conductores	5
RF17	Reporte de mantenimiento de vehículos	5
RF18	Reporte de provisión de combustibles	5
RF19	Reporte de matrícula	5
RF20	Reporte de movilizaciones diarias	5
RF21	Reporte de Novedades	5

Elaboración: (Ordóñez & Guaman, 2015)

La prioridad asignada a los requerimientos del Sistema de Gestión Vehicular están en base al desarrollo e implementación del mismo que parten de la tabla anterior donde se listan los requerimientos, por tal razón se ha dado el grado de prioridad de 5 a todos los requerimientos ya que su importancia en el sistema es muy alta.

Para más detalle de la **especificación de requerimientos** se puede visualizar en el **anexo D**.

#### **3.4.4.1 Perspectiva del Software**

El desarrollo del software tiene como finalidad satisfacer las necesidades de los usuarios involucrados y la problemática planteada. Para desarrollar la perspectiva de software se establecieron una serie de entregables de acuerdo a la metodología empleada (RUP) realizando cambios en cada iteración y dando cumplimiento de los requerimientos.

#### **3.4.4.2 Características del Software**

Las características del software son representadas en productos que en la Tabla 18 se detallan.

Tabla 18. Características del Software.

Producto	Concepto	Descripción	Código
<b>Gestión de Movilización</b>	Solicitud de movilización	Solicitud requerida por el usuario para movilizar un vehículo.	RF1
	Asignación Vehículo	Se asigna un vehículo disponible a un conductor.	RF
	Emitir Orden de Combustible.	La orden de combustibles consta con la cantidad de galones de combustible a despachar.	RF3
	Despacho de combustible	Guardar la cantidad real despachada de combustible.	RF4
	Orden de mantenimiento.	Esta orden al ser aprobada autoriza la reparación del vehículo.	RF5
<b>Gestión de Talleres</b>	Historial de mantenimiento	Resumen de los mantenimientos que se han realizado a los vehículos y permiten alertar a los conductores del próximo mantenimiento	RF6
	Plan de Mantenimiento preventivo	Guardar el plan de mantenimiento preventivo y que se aplicaran a los vehículos	RF7
<b>Gestión de la Administración</b>	Registro de vehículos.	Registrar y visualizar los datos de los vehículos.	RF13
	Registro de conductores.	Registrar y visualizar los datos de los conductores.	RF14
	Tipo de Licencia	Registrar licencias de los conductores.	RF10
	Gasolineras	Registrar las estaciones de servicios de combustibles.	RF12
	Combustibles	Registrar los tipos de combustibles.	RF11
	Usuarios	Registrar los usuarios del sistema y sus roles respectivos.	RF8
	Asignación vehículo/conductor.	Se asigna un vehículo disponible a un conductor.	RF9
<b>Gestión de Reportes</b>	Inventario de vehículos, accesorios y herramientas.	Reporte con los datos registrados de los vehículos y accesorios.	RF15
	Control de mantenimiento.	Reporte de los mantenimientos y reparaciones realizadas a los vehículos, los repuestos y lubricantes que fueron entregados, fecha en que se realizó el mantenimiento, motivo del mantenimiento o reparación.	RF17
	Control vigencia de la matrícula vehicular.	Reporte que visualiza y alerta al usuarios de la vigencia o caducidad de las matrículas vehiculares.	RF19
	Ordenes de movilización.	Reporte de las órdenes de movilización aprobadas con los datos como son: nombres de los conductores, nombres de los ocupantes, rutas, motivo y fecha de la solicitud.	RF20
	Partes de novedades.	Reportes de novedades de desperfectos de los vehículos.	RF21
	Provisión de combustibles.	Reporte de las órdenes de combustibles permitiendo visualizar subtotal y total por cada búsqueda.	RF18

Elaboración: (Ordóñez & Guaman, 2015)

### **3.5 Fase de Elaboración del Sistema de Gestión Vehicular**

En esta fase se define el **Documento de Especificación de Casos de Uso** que detallan de forma clara y concisa todas las funcionalidades del sistema. Además se define el **Documento de Arquitectura de Software** que tienen por objetivo integrar los requerimientos no funcionales del sistema con los atributos de calidad.

#### **3.5.1 Especificación de Casos de Uso**

Los usuarios de la aplicación web son la Coordinación de Gestión de Vehicular y los empleados del Gobierno Provincial de Loja donde ellos tendrán su propio perfil de usuario dentro de la aplicación.

En la Figura 8 se visualiza el diagrama de casos de uso para crear y emitir las ordenes de movilización y mantenimiento:

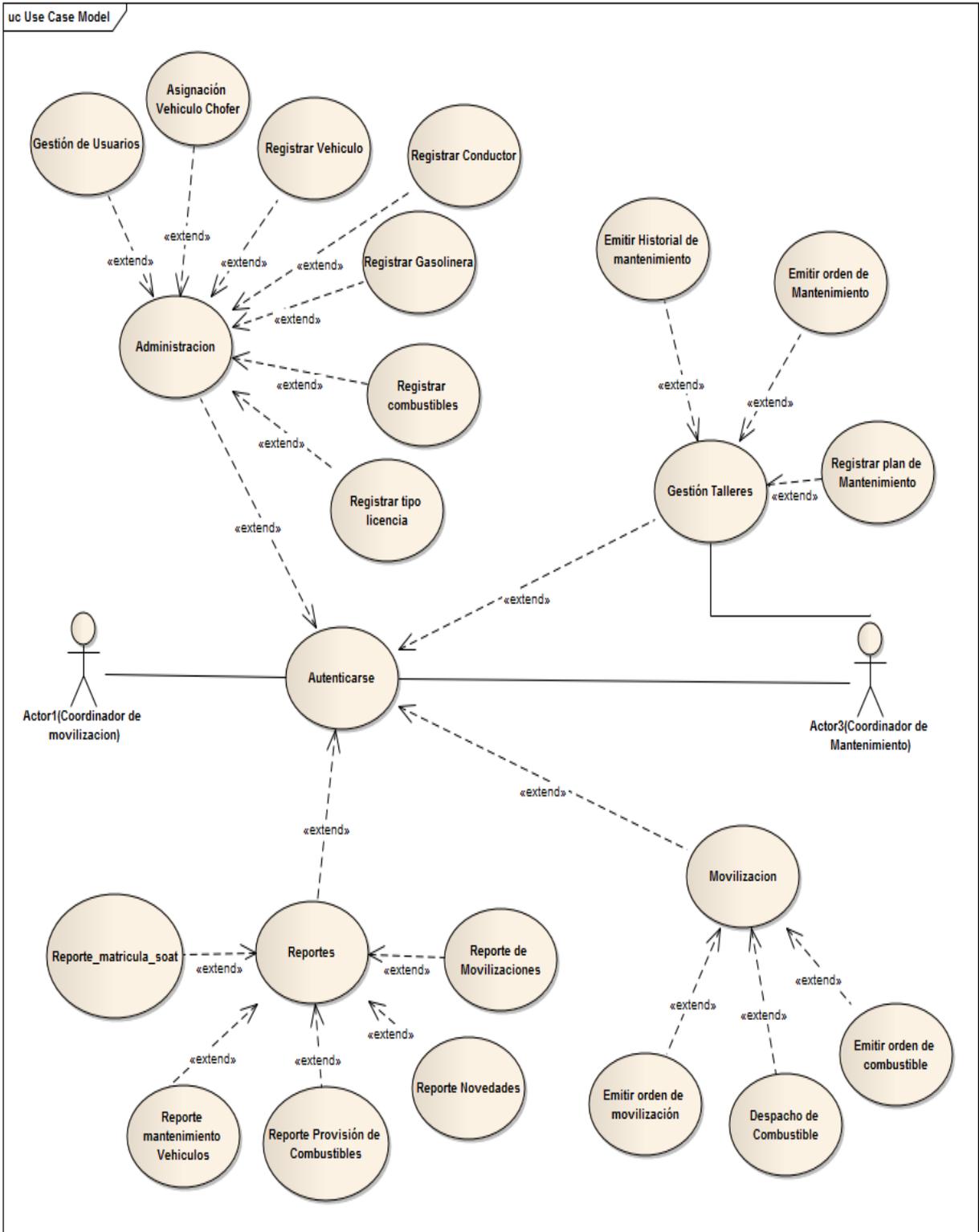


Figura 8 Diagrama de casos de uso  
 Elaboración: (Ordóñez & Guaman, 2015)

Para más detalle de los **casos de uso** se describen en el **anexo E**.

### 3.5.2 Arquitectura de Software

La arquitectura de software es la clave principal para estructurar, esquematizar y asegurar que el sistema cumpla con los niveles de estandarización, seguridad y calidad. Esta arquitectura permite dar cumplimiento al propósito de satisfacer los atributos de calidad (desempeño, seguridad y modificabilidad).

El patrón arquitectónico a utilizarse es el Modelo – Vista – Controlador que separa en 3 partes el código como son: los datos del sistema, la interfaz de usuario y la lógica del negocio. El patrón arquitectónico define la arquitectura 3 capas en: presentación, lógica de negocio y datos.

En la Figura 9 se visualiza el esquema de la arquitectura de software que representa el diseño del patrón arquitectónico, las tecnologías y seguridades a utilizarse.

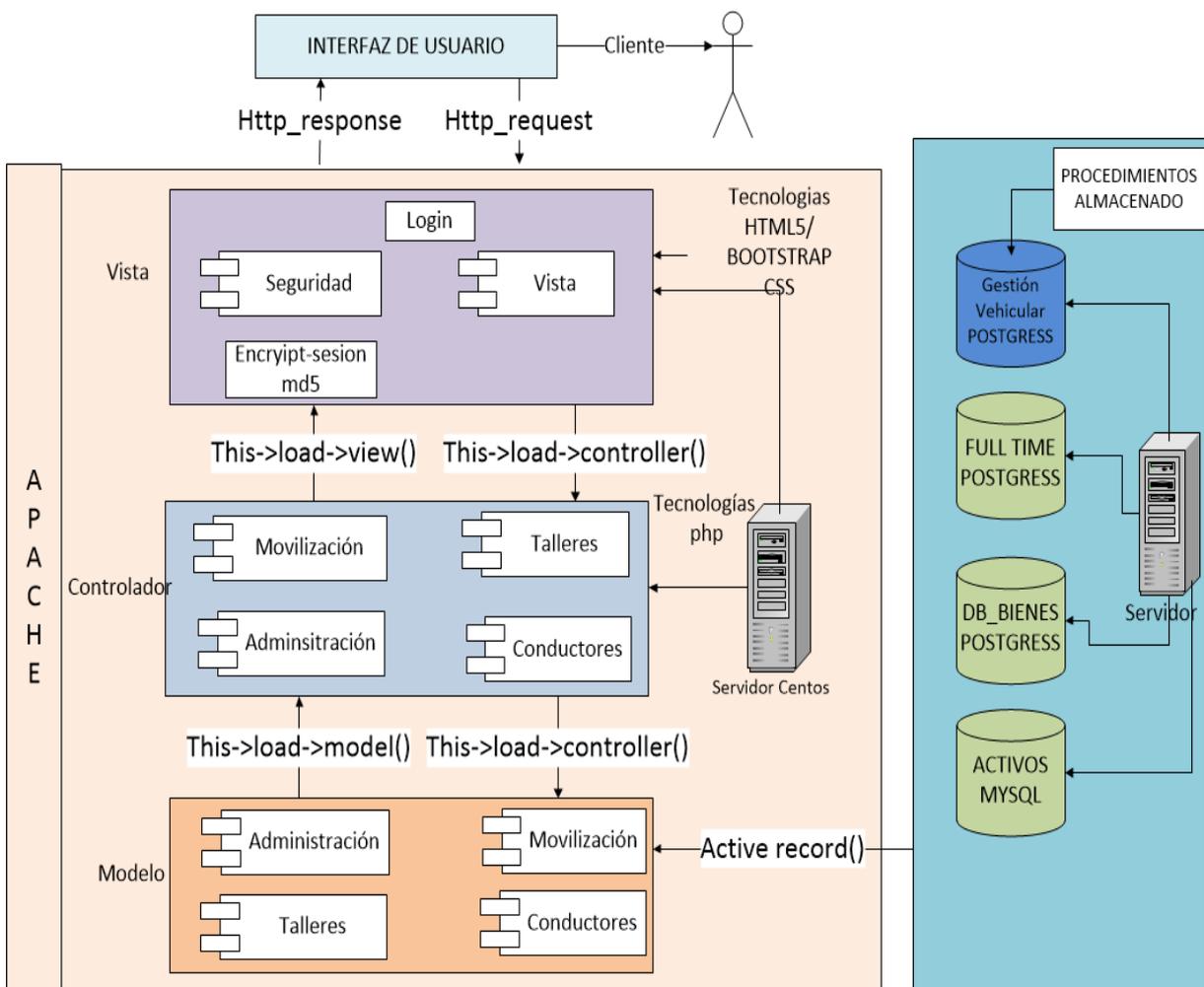


Figura 9. Arquitectura implementada  
Elaboración: (Ordóñez & Guaman, 2015)

La Figura 10 visualiza la integración del patrón arquitectónico en el framework de CodeIgniter.



Figura 10. Estructura del modelo, vista y controlador.  
Elaboración: (Ordóñez & Guaman, 2015)

Para más detalle a continuación se describe los componentes del patrón arquitectónico y tecnologías empleadas.

### **3.5.2.1 Descripción del Patrón Arquitectónico MVC.**

#### **3.5.2.1.1 Modelo**

El modelo representa la estructura de los datos con clases que contendrán funciones de insertar y actualizar la información desde las bases de datos.

CODEIGNITER para su distribución física en el directorio (models) incluye todos los scripts requeridos por el proyecto para trabajar sobre la base de datos en la Figura 11 se describe la estructura del modelo utilizado en los controladores:

```

ministracion_model.php (movile, fz3-14375899377130) - Sublime Text 2 (UNREGISTERED)
Project  Preferences  Help
administracion_model.php x  movilizacion_model.php x  administracion_model.php x
1 <?php if(! defined('BASEPATH')) exit ('No direct script access allowed');
2
3 class Administracion_model extends CI_Model{
4
5     function __construct(){
6         parent::__construct();
7         $this->load->database();
8         $this->db_b2 = $this->load->database('conexion_D',true);
9     }
10
11
12     function getIsvhiculosConductores($per_page, $desde){
13         if($query=$this->db->get("obtener_vehiculos_cohofer_list(".$per_page.", ".$desde.")as(cod_vehiculo integer, tipo chara
14         {
15             return $query->result();
16         }else{return false; }
17     }
18
19     function getChoferDisponible(){
20         if($query=$this->db->get("obtener_chofer_disponible() as (cod_chofer integer,nombres text);")
21         {
22             return $query;
23         }else{return false; }
24     }
25
26     function getVehiculo($cod_vehiculo){
27
28         $query = $this->db->get("obtener_unvehiculo_select(".$cod_vehiculo.");");
29         if($query->num_rows() > 0){
30             return $query;
31         }
32         else{ return false;}
33     }
34
35     function asignarChofer($data){
36
37         if($query=$this->db->query("select insertar_asig_chofer_vehiculo('$data[0]','$data[1]','$data[2]', $data[3], $data[4]
38         {
39             return $query;
40         }
41         else{return false; }
42     }
43
44     function update_asig_chofer_vehiculo($cod_vehiculo){
45         if($query=$this->db->query("select actualizar_vehiculos_chofer(".$cod_vehiculo.");")

```

Figura 11. Estructura del modelo  
Elaboración: (Ordóñez & Guaman, 2015)

En la Tabla 19 se detallan las clases del modelo del proyecto y las funciones para obtener, insertar, actualizar, y eliminar información.

Tabla 19. Clase y métodos del modelos

Clase	Método	Descripción
<b>Administración_model</b>	getIsvhiculosConductores	En esta clase modelo se especifica la administración de insertar, actualizar, listar y eliminar estos componentes:  Conductores Vehículos Gasolineras tipo Licencias Asignación de un vehículo a un chofer.
	getChoferDisponible	
	getVehiculo	
	asignarChofer	
	update_asig_chofer_vehiculo	
	update_asig_chofer_chofer	
	getAsignar_vehiculo	
	actu_asignar_chofer	
	delete_asig_chofer_vehiculo	
	getCombustibles	
	deleteCombustible	
	getActu_combustible	
	update_combustible_fin	
	getInsertarCombustibles	
	getGasolineras	
	getInsertarGasolinera	
	deleteGasolinera	
	getActu_Gasolinera	
	update_gasolinera_fin	
	getTipo_Licencia	
deleteTlicencia		
getTipo_licencia_id		
update_tipo_licencia_fin		

	getInsertarTipoLicencia getReg_Vehiculo get_select_combustibleRegVehiculo get_select_estado_actual get_select_veh_tipo getreg_insert_vehiculo get_select_placa_bd2 buscarDatosenBienesVehiculo1 buscarDatosenBienesVehiculo getInsertarFinRegVehiculo get_placa_bd2 deletereg_vehiculo getVehiculos getreg_vehiculo_id get_actualizar_vehiculo_id getConductores getMatriculaSoat get_novedades getVehiculos_matsoat	
<b>Conductores_model</b>	get_conductores_bd1 get_conductores get_conductor_bd1 get_conductor_bd2 get_nombres_apellidos buscarDatosConductores get_licencia get_insertar_conductor eliminar_conductor extraer_datos_chofer actualizar_conductor	La clase nos especifica el llamado a una base de datos externa al sistema, específicamente a los empleados conductores.
<b>Movilización_model</b>	buscarEmpleado agregarSolicitud getSolicitudes getOrdenesMovi getEstadoSolicitudes getProvincias getCantones getParroquias getMovilizacion actualizarMovilizacion getAllMovilizacion updateMovilizacion estadoMovilizacion getGasolineras getPartidasCombustible getChofer getVehiculos getCombustible agregarOrdenesCombustible getVehiculosDisponibles deleteSolicitud getOrdenesCombustible getListadoOrdenesCombustible getOrdenes_Combustible updateOrdenesCombustible obtenerCriteriosBusqueda_movilizacion getReporteBusqueda obtenerCriteriosBusqueda_combustible getReporteBusquedaCombustible	Esta clase modelo se implementa para el registro de solicitudes, de movilización, ordenes de combustible y reportes.
<b>Talleres_model</b>	buscarvehiculoPlaca buscarvehiculoNumvehiculo getPreventivo getBuscarClaseRepuesto buscarOperador agregar_mantenimiento agregar_respuesto	En esta clase se especifica el registro de las solicitudes de talleres, reportes y mantenimiento preventivo.

	getMantenimiento getRepuesto getHistorialVehiculo getControlMantenimiento agregar_mantenimiento_Preventivo getPlan_preven getPlan_preven_nombre deletePlan_preven update_plan_preven buscarkiloanterios getRepuestosUtilizados obtenerCriteriosBusqueda_mantenimiento getReporteBusqueda getReporteBusquedaCombustible	
<b>Usuario_model</b>	getUsuarios getAllUsuarios getMenus getSubmenus getRolPermiso insertUsuario getObtenerUsuarioRol obtener_usuario eliminarUsuario actualizarUsuario getActividadesUsuarios getbusquedaavanzada	En esta parte se crean los usuarios con los roles respectivos.

Elaboración: (Ordóñez & Guaman, 2015)

### 3.5.2.1.2 Vista

Representación visual de los datos en la aplicación web siendo responsable del uso de la información de la cual se dispone para producir la interfaz de usuario. Contiene todos los scripts de los formularios que se presentan para los usuarios finales, normalmente esta página no es una clase, sino más bien, permiten crear los archivos HTML generados con PHP, BOOTSTRAP, JavaScript, CSS.

En la Figura 12 se expone una vista creada para interactuar con el controlador, se puede observar la llamada del controlador para ejecutar las operación de guardado de los datos.

```

Project  Preferences  Help
administracion.php x  movilizacion.php x  administracion_model.php x  nuevoUsuario.php x
1 |<div class="panel panel-default">
2 |<div class="panel-heading">AGREGAR USUARIO</div>  Llama al controlador
3 |
4 |<div class="panel-body">
5 |<form id="nuevo" action="<?php echo base_url();>administracion/saveUsuario" method='post' onsubmit="return validate()">
6 |<table class='table'>
7 |<tr><td>
8 |<div class="form-group">
9 |<label for="inputEmail3" class="col-sm-2 control-label">Nombre</label>
10 |<input type="text" name="nombre" class="form-control" id="inputEmail3" placeholder="Ingrese su nombre y apellido" />
11 |</div>
12 |</td>
13 |<td>
14 |<div class="form-group">
15 |<label for="inputEmail3" class="col-sm-2 control-label">Usuario</label>
16 |<input type="text" name="usuario" class="form-control" id="inputEmail3" placeholder="Ingrese su usuario" required />
17 |</div>
18 |</td>
19 |<td>
20 |<div class="form-group">
21 |<label for="inputPassword3" class="col-sm-2 control-label">Password</label>
22 |<input type="password" name="password" class="form-control" id="inputPassword3" placeholder="Password" required />
23 |</div>
24 |</td>
25 |</tr>
26 |</table>
27 |<table>
28 |<tr><td>
29 |<div class="form-group"><?php echo $mensaje; ></div>
30 |</td>
31 |</tr>
32 |<tr><td><div class="form-group"><input type="submit" class="btn btn-default" value="Guardar" /></div></td></tr>
33 |</table><div id="mensaje" class="alert alert-success" style="display:none" role="alert"></div></td></tr>
34 |</table>
35 |</form>
36 |
37 |</div>
38 |</div>
39 |<script type="text/javascript">
40 |
41 |$(document).ready(function(){
42 |
43 |$("#nuevo").submit(function(){
44 |$.ajax({
45 |url: $(this).attr("action")

```

Figura 12. Estructura de la Vista  
Elaboración: (Ordóñez & Guaman, 2015)

La vista del módulo de administración del Sistema de Gestión Vehicular se describe a continuación en la Tabla 20.

Tabla 20. Vistas del módulo Administración

Vista	Descripción	Controlador y método relacionado
Usuarios	Vista de la administración de usuarios	administracion/ListaActividadesUsuarios
listado_vehiculo_chofer	Vista de la asignación de un vehículo a un chofer	administracion/listadoVehiculos_conductores
ListadoTipoLicencia	Vista de la administración del tipo de licencia.	administracion/listadoTipoLicencia
ListCombustibles	Vista de la administración de combustibles.	administracion/ListCombustibles administracion/ eliminar_combustible administracion/ actualizar_combustible administracion/ actu_combustible_fin administracion/ insertar_combustibles administracion/ insertarFinCombustibles
listadoGasolineras	Vista de la administración de las gasolineras.	administracion/listadoGasolineras administracion/insertar_gasolinera administracion/insertarFinGasolinera administracion/eliminar_gasolinera administracion/actualizar_gasolinera administracion/actu_gasolinera_fin
ListaVehiculosFinan	Vista del registro de vehículos.	administracion /insertar_reg_vehiculo administracion /buscarDatosenBienes administracion /buscarDatosenBienesVehiculo administracion /insertarFin_Reg_Vehiculo administracion /eliminar_reg_vehiculo administracion /actualizar_vehiculo
listar_conductores	Vista del registro de conductores.	administracion /insertar_Conductor administracion /buscarDatosConductoresFulltime administracion /insertar_conductor_final administracion /eliminar_Conductor administracion /actualizar_Conductor administracion /actualizar_conductor_final

Elaboración: (Ordóñez & Guaman, 2015)

La vista del módulo de reportes del Sistema de Gestión Vehicular se describe a continuación en la Tabla 21.

Tabla 21. Vistas del módulo Reportes

Vista	Descripción	Controlador y método relacionado
inventario_vehiculos	Vista de reportes de los vehículos.	administracion/reporte_vehiculos administracion/reportes_excel_vehiculos
reportes_conductores	Vista de los reportes de conductores que están de alta y baja	administracion/reporte_conductores administracion/reportes_excel_conductores
control_mantenimiento	Vista para generar el historial de mantenimiento de cada vehículo	Talleres/busquedaReporte talleres/reporte_excel_controlMantenimiento talleres/reporte_control_mantenimiento_view talleres/repuestos_vizualizar_reporte
provision_combustible	Vista para generar el total de combustibles asignados a cada vehículo.	Movilización/reporte_combustible_view Movilización/busqueda_reporte_combustible Movilizacion/reporte_combutible_Excel
matricula_soat	Vista para generar el reporte de vencimiento de matrícula vehicular.	Administracion/reporte_matriculaSoat Aministracion/Mat_Soat_Excel
reporte_movilizacion	Vista para generar los reportes de movilización periódicamente.	Movilización/reporte_movilizacion_view Movilización/busquedaReporte Movilizacion/reporte_excel_movilizacion
reporte_novedades	Vista para generar reportes de novedades de las movilizaciones correspondiente a cada vehículo	Administracion/reporte_novedades Administracion/reporte_excel_novedades

Elaboración: (Ordóñez & Guaman, 2015)

La vista del módulo de gestión de movilización del Sistema de Gestión Vehicular se describe a continuación en la Tabla 22.

Tabla 22. Vistas del módulo Gestión de Movilización

Vista	Descripción	Controlador y método relacionado
Solicitud	Vista de registro de solicitudes de movilización.	Movilización/solicitud Movilización/buscarEmpleado Movilizacion/save_solicitud Movilizacion/actualizarMovilizacion Movilizacion/aprobarSolicitud_Movilizacion
aprobacionMovilizacion	Vista que presentan todas las solicitudes de movilización ingresadas para proceder a la asignación de los vehículos.	Movilizacion/asignar_vehiculo movilizacion/asignacionVehiculo
aprobarOrdenesCombustible	Vista que presentan las solicitudes de movilización ingresadas y aprobadas para la asignación de combustibles.	movilizacion/asignarCombustible
listarOrdenesCombustible	Vista que presentan las solicitudes de combustibles asignadas para proceder al despacho del combustible.	movilizacion/despacharCombustible/

Elaboración: (Ordóñez & Guaman, 2015)

La vista del módulo de talleres del Sistema de Gestión Vehicular se describe a continuación en la Tabla 23.

Tabla 23. Vistas del módulo de Talleres

Vista	Descripción	Controlador y método relacionado
Solicitud	Vista que registra las ordenes de mantenimientos.	talleres/save_ordenesMantenimiento imprimir/salvo_conducto talleres/buscarSolicitudpLaca talleres/buscarSolicitudNumVehiculo talleres/solicitud_talleres_repuesto talleres/kilometrajeMantenimiento
list_historial_mantenimiento	Vista que presenta el ultimo mantenimiento por los vehículos.	talleres/historial_mantenimiento talleres/historial_mantenimiento_view
plan_preventivo	Vista que administra el plan de mantenimiento preventivo	Talleres/plan_preventivo talleres/agre_mante_preventivo talleres/eliminar_plan_man_preven talleres/actualizar_view_plan_preven

Elaboración: (Ordóñez & Guaman, 2015)

### 3.5.2.1.3 Controlador

Permite gestionar las peticiones de los usuarios con información necesaria donde el controlador actúa como intermediario de la vista y el modelo que se visualiza en la Figura 13.

```

1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3  class Administracion extends CI_Controller {
4
5      function __construct(){
6          parent::__construct();
7          $this->load->model('usuario_model');
8          $this->load->model('administracion_model');
9          $this->load->model('conductores_model');
10         $this->load->library('session');
11         $this->load->library('pagination');
12         $this->load->helper('url');
13     }
14
15     public function index(){
16
17         $this->load->view('headers');
18         $this->load->view('principal');
19     }
20     public function listadoUsuarios(){
21         $this->load->view('headers');
22         $this->load->view('principal');
23         $datos['usuarios']=$this->usuario_model->getAllUsuarios();
24         $this->load->view('administracion/usuarios', $datos);
25     }
26
27     public function agregarUsuario(){
28         $menu="";
29         $dato_menu = $this->usuario_model->getMenus();
30
31         foreach ($dato_menu->result() as $key) {
32             $menu.="<div class='row'><div class='col-sm-9'>";
33
34             $menu.="<b>". $key->nom_menu."</b><div class='row'>";
35             $dato_sub = $this->usuario_model->getSubmenus($key->id_menus);
36
37             foreach ($dato_sub->result() as $sub) {
38                 $menu.="<div class='col-md-6'><input type='checkbox' name='menu[]' value='".$sub->id_submenu.">".$sub->
39
40             }
41             $menu.="</div></div></div>";
42         }
43
44         $datos['menu_sub']= $menu ;
45         $this->load->view('administracion/nuevoUsuario', $datos);

```

Figura 13. Estructura del Controlador

Elaboración: (Ordóñez & Guaman, 2015)

En este componente se han implementado las siguientes clases y métodos que se detallan en la Tabla 24.

Tabla 24. Métodos de la clase

Clase	Métodos
<b>Administración</b>	Index listadoUsuarios agregarUsuario saveUsuario eliminarUsuario actu_usuario_view actualizar_usuario ListaActividadesUsuarios reporte_excel_actividadesusuarios obtenerActividades listadoVehiculos_conductores asignar_chofer view_asignarChofer visualizar_asignar_chofer actu_asignar_choferfin eliminar_asignar_chofer ListCombustibles eliminar_combustible actualizar_combustible actu_combustible_fin insertar_combustibles insertarFinCombustibles insertar_gasolinera insertarFinGasolinera eliminar_gasolinera actualizar_gasolinera actu_gasolinera_fin listadoTipoLicencia eliminar_Tlicencia actualizar_Tlicencia actu_tipo_licencia_fin insertar_Tlicencia insertarFinTipoLicencia listaVehiculosFinan insertar_reg_vehiculo buscarDatosenBienes buscarDatosenBienesVehiculo insertarFin_Reg_Vehiculo eliminar_reg_vehiculo actualizar_reg_vehiculo cargar_conductores insertar_Conductor buscarDatosConductoresFulltime insertar_conductor_final eliminar_Conductor actualizar_Conductor actualizar_conductor_final reporte_vehiculos reporte_condutores Mat_Soat_Excel reporte_matriculaSoat reporte_novedades reporte_excel_novedades reportes_excel_vehiculos reportes_excel_conductores
<b>Movilización</b>	Index Solicitud buscarEmpleado save_solicitud viewSolicitudMovi actualizarMovilizacion

---

	asignar_vehiculo enviarAprobar asignacionVehiculo asignarChoferDiferent aprobarSolicitud_Movilizacion asignarCombustible save_ordenesCombustible aprobarCombustible listarOrdenesCombustible despacharCombustible update_ordenesCombustible asignacionVehiculoChofer eliminar_solicitud reporte_movilizacion_view reporte_movilizacion_view_sesion busquedaReporte reporte_excel_movilizacion reporte_combustible_view busqueda_reporte_combustibleSesion busqueda_reporte_combustible reporte_combustible_Excel
<b>Talleres</b>	Index solicitudMantenimiento buscarSolicitudpLaca buscarSolicitudNumVehiculo save_ordenesMantenimiento historial_mantenimiento_view busquedaSesion_historial historial_mantenimiento plan_preventivo control_mantenimiento agre_mante_preventivo cargarModelo_vehiculo cargarVehiculo_modelo save_ordenesMantenimiento_Preventivo repuesto_preventivo eliminar_plan_man_preven actualizar_view_plan_preven actualizar_plan_preven repuestos_vizualizar_reporte reporte_control_mantenimiento_view busquedaReporteSesion busquedaReporte reporte_excel_controlMantenimiento
<b>Welcome</b>	Index com_login logout
<b>Imprimir</b>	Index salvo_conducto orden_combustible solicitud_mantenimiento nota_repuestos

Elaboración: (Ordóñez & Guaman, 2015)

Para más detalle acerca de la **Arquitectura del Software** se puede visualizar en el **anexo G**.

### 3.6 Fase de Construcción del Sistema de Gestión Vehicular

En esta fase presenta el diseño de la posible solución de software la misma que contiene los requerimientos funcionales y no funcionales del sistema.

CODEIGNITER utiliza formularios HTML generados con PHP los cuales son coordinados por el controlador para establecer la comunicación con el modelo. A continuación se detalla la estructura básica de las interfaces como también el patrón de diseño.

### **3.6.1 Diseño de interfaz de Usuario**

Para el diseño de la interfaz web se utilizó el lenguaje HTML5 diseñados con CSS como son:

- Logotipo de la empresa.
- Construcción de diseño a través de CSS.
- Uso del framework Bootstrap.
- Utilización de nuevos componentes de HTML5.

#### ***3.6.1.1 Diseño de la interfaz de ingreso al sistema***

En la Figura 14 se visualiza la interfaz para el ingreso o logeo del sistema validando usuario y contraseña.



Figura 14. Interfaz de ingreso al sistema  
Elaboración: (Ordóñez & Guaman, 2015)

#### ***3.6.1.2 Diseño de la Interfaz de la Pantalla Principal***

En la Figura 15 se visualiza la interfaz principal del sistema está carga cuando se autoriza el acceso a los usuarios.

Figura 15. Interfaz principal  
Elaboración: (Ordóñez & Guaman, 2015)

### 3.6.1.3 Diseño de los Reportes

En la Figura 16 se visualiza la interfaz para generar reportes con la información más relevante.

NRO DE ORDEN	FECHA	CONDUCTOR	TIPO DE VEHL.	NRO GPL	LUGAR DE LA COMISION	KILOMETRAJE	GALONES	VALOR
241	10/23/2015	JOFFRE ULISES FIERRO JARAMILLO	JEEP	73	LOJA-CATAMAYO-LOJA	2345678	10	21.8
244	10/23/2015	WILLAN RICARDO CUEVA JARAMILLO		108	nuevas dasdasds	1356789	10	10.37
245	10/23/2015	JOFFRE ULISES FIERRO JARAMILLO	NN	25	Asignar ruta por un dia	1234567892	5	5.185
247	10/23/2015	RUTBEL YOVANY AMBULUDI CALERO		26	sdsasds	3456789	5	7.4
248	10/23/2015	SEGUNDO VICENTE ORTEGA CAPA		51	se dirige por toda la ciudad de loja	500000	5	7.4
249	10/23/2015	SEGUNDO VICENTE ORTEGA CAPA		51	se dirige por toda la ciudad de loja va por todo lado de la ciudad de loja	2345678	5	7.4
250	10/23/2015	GALO AUGUSTO PIEDRA ORDOÑEZ		21	se dirige por toda la ciudad de loja	23456789	5	5.185
251	10/28/2015	SEGUNDO VICENTE ORTEGA CAPA		51	RECORRIDO POR LA CIUDAD DE LOJA	345000	5	7.4

Figura 16. Interfaz Reportes  
Elaboración: (Ordóñez & Guaman, 2015)

Para la presentación de los reportes se utilizaron las librerías de paginación y el plugin del **DateTables**.

## 3.7 Fase de Transición del Sistema de Gestión Vehicular

El objetivo de la fase es poner en producción el sistema para los usuarios o clientes finales, realizando capacitaciones y las configuraciones de instalación. En ésta fase se ha entregado el manual de usuario que se describe en el **anexo G** con el fin de brindar apoyo y guiar a los usuarios.

**CAPÍTULO IV**  
**PRUEBAS PARA EL SISTEMA DE GESTIÓN VEHICULAR**

## 4.1 Introducción

El presente capítulo describe el plan de pruebas de la metodología RUP donde se establecen técnicas, herramientas y actividades relacionadas a cada una de las pruebas realizadas para sistema, con este plan de pruebas se permite cumplir con los requerimientos funcionales y no funcionales del sistema, garantizar la calidad y el desempeño del software.

## 4.2 Tipos de Pruebas

En la siguiente tabla se presentan el tipo de pruebas que serán aplicadas al sistema en cada fase de la metodología RUP que se detallan en la Tabla 25.

Tabla 25. Tipo de pruebas aplicadas al Software

DEFINICIONES		
<b>Unitarias</b>	Permite verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.	<b>Elaboración</b>
<b>Sistema Seguridad</b>	<b>Sistema:</b> Estas pruebas buscan diferencias entre la solución desarrollada y los requerimientos con el fin de identificar errores que se puedan generar entre la especificación funcional y el diseño del sistema. <b>Seguridad:</b> Verifica el cumplimiento de las políticas de seguridad acordadas para el sistema.	
<b>Funcional Usabilidad</b>	<b>Funcional:</b> La prueba funcional es un proceso para procurar encontrar discrepancias entre el programa y la especificación funcional. <b>Usabilidad:</b> Esta prueba permite encontrar problemas de factores humanos, o usabilidad.	<b>Construcción</b>
<b>Aceptación</b>	Es la prueba final basada en las especificaciones del usuario o basada en el uso del programa.	<b>Transición</b>

Elaboración: (Ordóñez & Guaman, 2015)

## 4.3 Estrategia de las Pruebas

La estrategia de pruebas proporciona la descripción de los pasos que se debe llevar a cabo como parte del plan de pruebas, técnicas, herramientas, recursos, criterios de éxito y la culminación de las pruebas.

Las estrategias de pruebas del software tienen las siguientes características:

- Las pruebas comienzan a nivel de módulo trabajando hacia afuera y sobre la integración de todo el sistema.
- Según el momento son apropiadas técnicas de pruebas.
- La prueba se lleva a cabo por el responsable del desarrollo del software
- La prueba y depuración son actividades diferentes.

### 4.3.1 Técnicas de Especificación de las Pruebas.

Las estrategias del proceso del plan de pruebas se implementaran de acuerdo al siguiente esquema que se visualiza en la Figura 17:

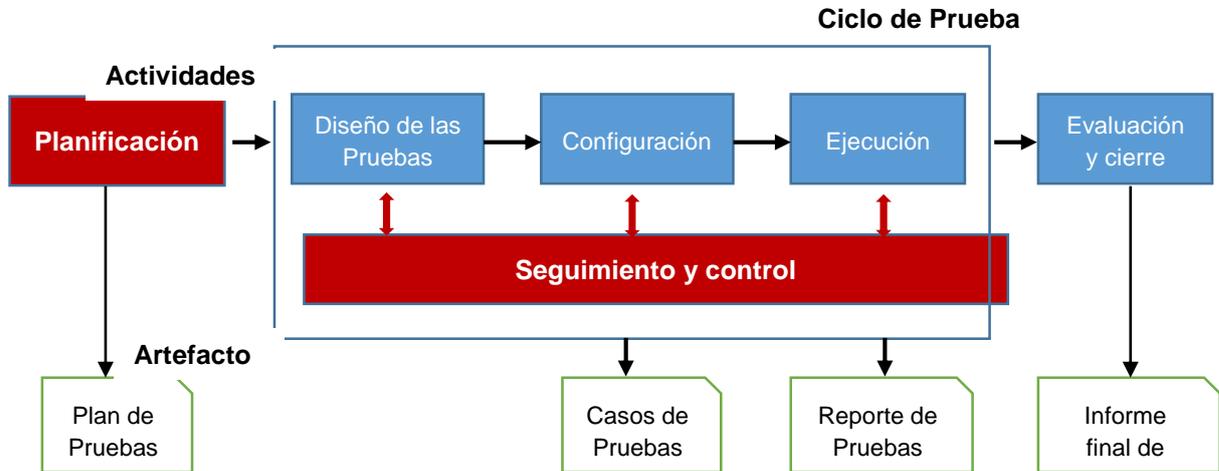


Figura 17. Ciclo del plan de pruebas  
Fuente: (Pressman R., 2005)

A continuación se describe las fases del ciclo del plan de pruebas usados para el Sistema de Gestión Vehicular:

#### 4.3.1.1 Planificación

Para el desarrollo del Sistema de Gestión Vehicular se debe considerar la ejecución de un plan de pruebas haciéndose necesario la planificación del mismo, teniendo en cuenta los siguientes parámetros:

- Se especifican las técnicas a utilizar
- Criterios de aceptación
- Recursos involucrados
- Uso de herramientas
- Criterios de aceptación

#### 4.2.1.2. Diseño de las pruebas

Para el diseño de las pruebas, se tendrán en cuenta aspectos que permitirán encontrar defectos en el periodo de desarrollo de software con la realización de las pruebas de verificación y validación de los datos se obtendrá los siguientes ítems:

**Alcance** de las pruebas estará dado por el marco del sistema:

- Procesos
- Descripción de casos de uso
- Vista lógica
- Vista de implementación
- Vista de despliegue
- Requerimientos no funcionales
- Prototipos del sistema

**Inventario de Pruebas** permitirán:

- Definir y asignar prioridades: Alta, media y Baja
- Se estima el tiempo en probar cada funcionalidad
- Evaluar aspectos técnicos del sistema.

**Resultado y ejecución de las pruebas:** Se resaltan los puntos de partida para la ejecución de las pruebas.

#### 4.3.1.2 Ejecución

Como se puede observar en la Figura 18 se visualiza un modelo de pruebas en V, este modelo se extiende a lo largo de todo el ciclo de vida del software. Mientras se realizan las fases de análisis, diseño, implementación y las pruebas.

Al llegar a la fase de pruebas se ejecuta la codificación de las pruebas unitarias hasta las pruebas funcionales

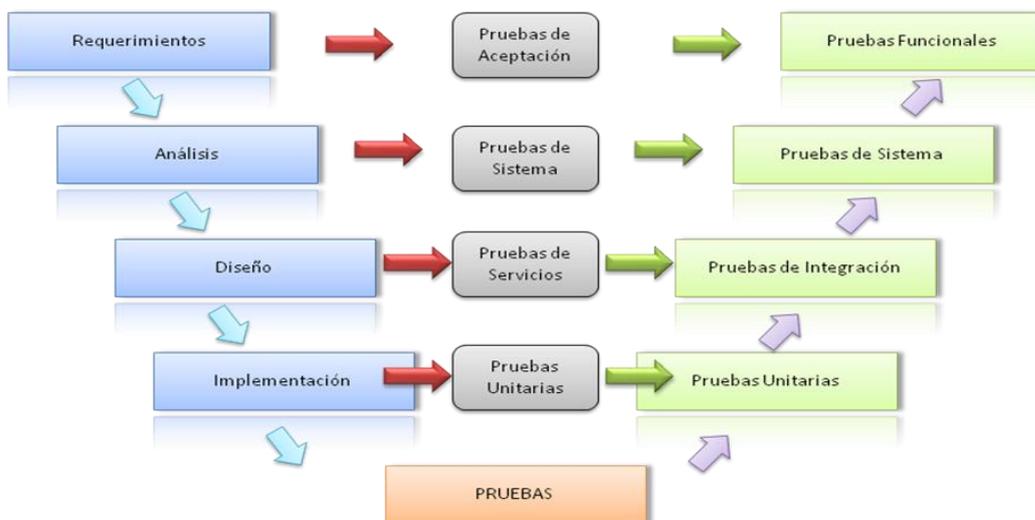


Figura 18. Modelo de Pruebas en V  
Fuente: (Federal Highway Administration, 2005)

#### 4.2.1.5 Evaluación y cierre

Para la evaluación y cierre de las pruebas se presentará el informe de pruebas donde se documentará el resultado de las diferentes pruebas ejecutadas al Sistema de Gestión Vehicular.

#### 4.4 Herramientas para Pruebas

Las herramientas que se utilizarán dependerán del tipo de pruebas que se realizan, se puede decir que por cada prueba es posible que se utilice una herramienta diferente como se detallan en la Tabla 26.

Tabla 26. Herramientas para Pruebas

HERRAMIENTA	CARACTERÍSTICAS	TIPO DE PRUEBA
<b>SimpleTest CodeInteger</b>	<p>Librería de CodeInteger para pruebas unitarias.</p> <p>Nos proporciona:</p> <ul style="list-style-type: none"> <li>• Escribir tests de una manera sencilla.</li> <li>• Ejecutarlos con facilidad.</li> <li>• Analiza los resultados.</li> </ul>	<p>Esta herramienta será utilizada para la ejecución de:</p> <ul style="list-style-type: none"> <li>• Pruebas unitarias.</li> </ul>
<b>WEBPAGE TEST W3C validator W3C Link Cheker</b>	<p>Ejecuta pruebas de velocidad al sitio por cada transacción.</p> <p>Servicio de validación de hojas de estilo en cascada, aplicación diseñada para encontrar errores comunes o topográficos o usos incorrectos de CSS.</p>	<p>Esta herramienta será utilizada para la ejecución de:</p> <ul style="list-style-type: none"> <li>• Pruebas Funcionales.</li> <li>• Pruebas de Sistema</li> <li>• Pruebas de Interfaz de Usuario</li> </ul>

Elaboración: (Ordóñez & Guaman, 2015)

#### 4.5 Configuración del ambiente de las pruebas

El ambiente y configuración de las pruebas será implementado tomando en cuenta la infraestructura tecnológica que posee el Gobierno Provincial de Loja que se describe en la Tabla 27.

Tabla 27. Ambiente y Configuración de las Pruebas

Equipo	SO	Características	Software instalado
<b>Servidor</b>	Centos 6.6	32 bits(i386)	Apache 5.5 Base de datos postgres 8.20
<b>Computador</b>	Windows 7	Procesador: Intel(R) Core(TM) i5-337U CPU @1.80 Ghz Memoria instalada: 2GB Tipo de sistema: Sistema operativo de 32bits	

Elaboración: (Ordóñez & Guaman, 2015)

## 4.6 Ejecución y Resultados de las Pruebas

El tipo de pruebas que se describieron en la lista anterior se contemplan en las fases de la metodología de desarrollo de software, las que tienen como propósito:

- Descubrir errores.
- Identificar y resolver falencias funcionales de la aplicación.
- Cumplir con los requerimientos del cliente.
- Evaluar la calidad del software

### 4.6.1 Pruebas Unitarias

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado como se detalla en la Tabla 28.

Tabla 28. Ejecución de Pruebas Unitarias

Objetivo de la Prueba:	Validar las piezas individuales del software como una unidad independiente.
Estrategia	<ul style="list-style-type: none"> <li>• Se efectúan para los servicios del negocio y para la lógica del sistema en la capa web que tengan complejidad alta.</li> <li>• Generar casos de pruebas necesarios que permitan identificar:</li> <li>• Que al menos cada sentencia o instrucción del programa se ejecute al menos una vez correctamente.</li> <li>• Que cada condición tenga por lo menos una vez un resultado positivo y/o negativo.</li> <li>• Que cada bucle del sistema se pueda probar considerando: ignorar el bucle, pasar una vez y pasar n veces.</li> </ul>
Herramienta requeridas:	<ul style="list-style-type: none"> <li>• SimpleTest en CodeInteger</li> </ul>
Observaciones	<p>La prueba se realizará por Módulo entendiéndose por tal:</p> <ul style="list-style-type: none"> <li>• Bloque básico de programa</li> <li>• Implementa funciones independientes y simples</li> <li>• Puede probarse por separado.</li> </ul>

Elaboración: (Ordóñez & Guaman, 2015)

En la tabla 29 se presentan los resultados obtenidos de las pruebas unitarias aplicadas al software. Se evaluo por cada actividad de cada modulo.

Tabla 29.Resultados de las Pruebas Unitarias

	Errores	Aciertos
<b>Gestión de movilización</b>	0	4
<b>Talleres</b>	0	3
<b>Consulta de Reportes</b>	0	5
<b>Administración</b>	0	7
<b>Total</b>	0	19
<b>Total %</b>	0%	100%

Elaboración: (Ordóñez & Guaman, 2015)

Las métricas que se evaluaron son errores y aciertos.

- Errores: las funcionalidades que muestran errores.
- Aciertos: Las funcionalidades de los componentes funcionaron correctamente.

Los resultados de las pruebas unitarias visuales se muestran en la Figura 19.

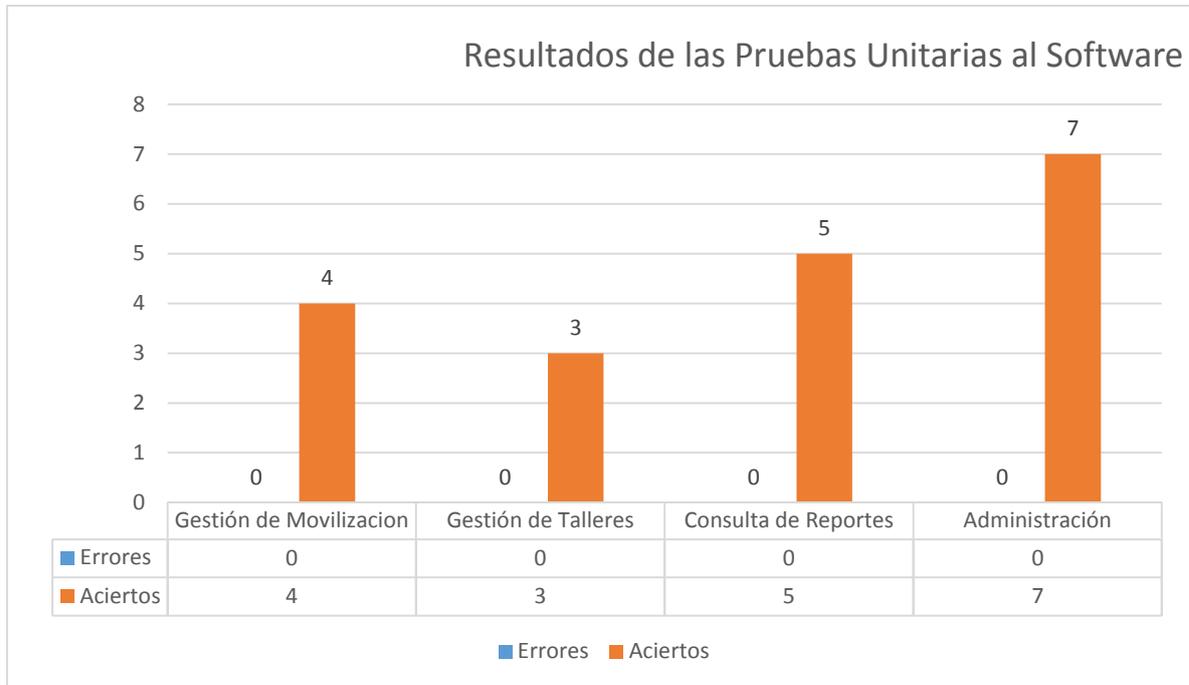


Figura 19. Resultado de las Pruebas Unitarias.  
Elaboración: (Ordóñez & Guaman, 2015)

Se puede concluir que los resultados obtenidos en los casos de prueba dieron el 100% de acierto lo que quiere decir que el sistema esta funcionando correctamente en cada una de sus funcionalidades.

Para más información sobre las Pruebas Unitarias ejecutadas en la solución del software se puede dirigir al **anexo H**.

#### 4.6.2 Pruebas del sistema

Las pruebas de sistema buscan diferencias entre la solución desarrollada y los requerimientos, enfocándose en la identificación de los errores que se puedan generar entre la especificación funcional y el diseño del sistema, así como, el negocio como objeto de la aplicación que se describen en la Tabla 30.

Tabla 30. Pruebas del sistema ejecutadas.

<b>Objetivo de la Prueba:</b>	<ul style="list-style-type: none"> <li>Validar aquellos volúmenes de datos máximos (por lo general las transacciones o informes) que pueden ser completados dentro de un período específico en el tiempo y con un nivel de concurrencia dado (carga, concurrencia y desempeño).</li> <li>Validar los requerimientos no funcionales de cada proyecto.</li> </ul>
<b>Estrategia:</b>	<ul style="list-style-type: none"> <li>Realizar Set de Pruebas a partir de los Requerimientos no funcionales.</li> <li>Realizar pruebas de rendimiento básico. Consiste en probar la aplicación simulando la carga esperada en el entorno de producción.</li> <li>Realizar las pruebas de concurrencia: verificar el comportamiento de la aplicación en condiciones de sobrecarga de usuarios, que supone permitirá identificar potenciales problemas de rendimiento o cuellos de botella antes de su pase a producción.</li> <li>Realizar pruebas de requerimientos no funcionales: consiste en probar la aplicación con cada uno de los requerimientos no funcionales establecidos en el proyecto.</li> <li>Identificar posibles cuellos de botella o problemas de rendimiento.</li> <li>Realizar pruebas de carga: altos volúmenes de información.</li> </ul>
<b>Herramienta requeridas:</b>	<ul style="list-style-type: none"> <li>WEBPAGE TEST</li> <li>W3c validator</li> <li>W3c Link Cheker</li> </ul>
<b>Observaciones:</b>	<p>La elección de la herramienta de prueba será a discreción del grupo de pruebas y su elección dependerá de la prueba que se va a realizar.</p> <p>El conjunto de pruebas que se abarcaran son las siguiente:</p> <ul style="list-style-type: none"> <li>Prueba de Estrés</li> <li>Prueba de Interfaz de usuario</li> <li>Prueba de rendimiento</li> <li>Prueba de seguridad</li> </ul>

Elaboración: (Ordóñez & Guaman, 2015)

Para más información sobre los resultados de las pruebas del sistema se puede dirigir al **anexo H**.

### 4.6.3 Pruebas de Interfaz de Usuario

La funcionalidad principal de las pruebas de interfaz de usuario consisten en hacer revisiones precisas de la forma en la que se despliegan las páginas del sitio y si cumplen con los estándares mínimos definidos por la normativa.

En la Tabla 31 se muestra en resultado obtenido de las pruebas de interfaz de usuario aplicadas al software.

Tabla 31. Pruebas de Interfaz de Usuario

	Descripción	Errores	Aciertos
<b>Verificación de Contenidos</b>	Verificación de detección y enlaces rotos.	0%	100%
<b>Verificación de Estándares</b>	Validación del código HTML.	24%	76%
<b>Verificación de Interfaces</b>	Diagramación vs. Browsers El contenido se los puede visualizar correctame en los siguientes navegadores. (Google Chrome, Microsoft Internet Explorer, Opera y Mozilla Firefox.)	0%	100%
<b>Total %</b>			

Elaboración: (Ordóñez & Guaman, 2015)

Las métricas que se evaluaron son:

- Errores: muestran el porcentaje que dio el test aplicado a la pruebas de interfaz de usuario
- Aciertos: si el test dio correctamente 100%.

En la siguiente Figura 20 se visualiza gráficamente los resultados de las pruebas de interfaz de usuario.

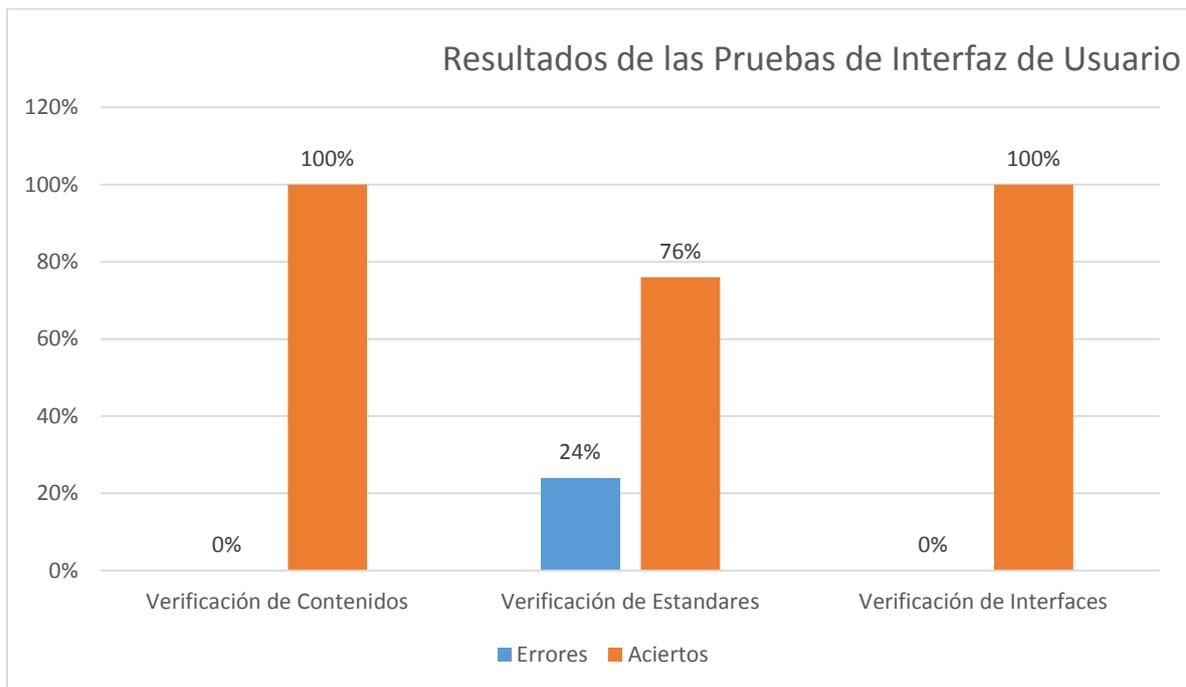


Figura 20. Resultados de las pruebas de interfaz de usuario.  
Elaboración: (Ordóñez & Guaman, 2015)

Se puede concluir que al aplicar el test para realizar las pruebas de interfaz de usuario dos de ellas dieron como resultado 100% y el otro supero más del 75% por lo que se concluye que el software está funcionando correctamente la interfaz de usuario.

#### 4.6.4 Pruebas de Rendimiento

El objetivo de las pruebas de rendimiento son determinar si el usuario está satisfecho con la velocidad con se ejecuta el software.

A continuación en la Tabla 32 se muestra los tiempos en los que se ejecutaron cada requerimiento del software.

Tabla 32. Pruebas de Rendimiento

Módulo	Acción	Tiempo
<b>Gestión de Movilización</b>	Solicitud de Movilización	0.2010s
	Asginación Vehículo	0.1050s
	Orden de combustible	0.2000s
	Despacho de combustible	0.1396s
<b>Gestión de Talleres</b>	Solicitud de mantenimiento	0.2030s
	Historial de Mantenimiento	0.3100s
	Plan de mantenimiento preventivo	0.1030s
<b>Consulta de Reportes</b>	Control de mantenimiento	0.3160s
	Provisión de combustibles	0.6900s
	Matricula y soat	0.2110s
	Ordenes de Movilización	0.6900s
	Novedades	0.1930s
<b>Administración</b>	Registro de usuarios	0.0653 s
	Asignación vehículo chofer	0.1793 s
	Registro Licencia	0.2000 s
	Registro combustible	0.1590s
	Registro Gasolinera	0.1900s
	Registro vehículo	0.3170s
	Registro conductor	0.1860s

Elaboración: (Ordóñez & Guaman, 2015)

A continuación se detallan los promedios obtenidos en el resultado de las pruebas de rendimiento. Las métricas que se evaluaron son las siguientes:

- Tiempo mínimo: 0.0653s es el promedio mínimo de una solicitud.
- Tiempo medio: 0.3170s es el promedio medio de una solicitud.
- Tiempo máximo: 0.6900s es el promedio máximo de una solicitud.

Se puede concluir de las pruebas de rendimiento dependen bastante de la veracidad de internet que posee la Institución y con la cantidad de los datos que se manejen, al realizar el test nos da como resultado tiempos mínimos de ejecución que fueron 0.0653s y tiempos máximos de ejecución que fueron 0.6900s.

#### 4.6.5 Pruebas de Seguridad

El objetivo de las pruebas de seguridad es constatar y validar las políticas de seguridad propuestas e integradas durante el transcurso del sistema, que a continuación se describen:

##### 4.6.5.1 Asistente de Seguridad de CODEIGNITER:

El asistente de seguridad integrado en los proyectos de CODEIGNITER permite implementar funciones relacionadas a la seguridad en el código PHP.

Para cargar el Asistente de Seguridad se incluye la siguiente línea de código:

```
$this->load->helper('security');
```

Las siguientes funciones se encuentran disponibles para el manejo de seguridades en el código PHP:

- `xss_clean()`: Permite dar filtrado a los ataques XSS (Cross Site Script).

```
xss_clean($ str [, $ is_image = FALSO]);
```

- `dohash()`: Permite encriptar contraseñas con MD5.

```
$cadena=dohash($cadena,'md5');
```

- `strip_image_tags()`: Elimina las etiquetas de las imágenes de una cadena obteniendo como resultado la URL de la imagen como texto plano.

```
$cadena=strip_image_tags($cadena);
```

- `encode_php_tags()`: Convierte etiquetas PHP a entidades, si usa la función `xss_clean()` se ejecutará automáticamente la función.

```
$cadena = encode_php_tags($cadena);
```

#### 4.6.1.2 Procedimientos Almacenados POSTGRESQL:

Una de las utilidades más eficaces que los procedimientos almacenados brindan es en relación a la seguridad e integridad de las bases de datos, se implementa los procedimientos almacenados para todas las operaciones comunes del sistema ya que proporciona un entorno seguro y consistente y pueden asegurar que cada operación se ejecuta apropiadamente. Al integrar en el sistema los procedimientos almacenados nos brinda seguridad en cuanto a las aplicaciones y los usuarios que no obtendrían ningún acceso directo a las tablas de la base de datos, sólo pueden ejecutar algunos procedimientos almacenados autorizados para el rol del sistema. Para más información sobre los resultados de las pruebas funcionales se puede visualizar en el **anexo H**.

#### 4.6.2 Pruebas Funcionales

Las pruebas funcionales son un proceso para procurar encontrar discrepancias entre el software desarrollado y la especificación funcional. La prueba funcional normalmente es una actividad de caja negra. Esta prueba permite validar:

- Los procesos y reglas de negocio establecidas,
- Que se cumplan los requerimientos funcionales establecidos

En esta prueba se validan los casos de uso que fueron aprobados por el cliente, y a partir de ellos se diseñan y ejecutan los set de pruebas correspondientes. Se deben elaborar los casos de pruebas necesarios que permitan asegurar el funcionamiento de todos los flujos normales y alternos de dichos casos de uso como se detalla en la Tabla 33.

Tabla 33. Pruebas Funcionales ejecutadas

Objetivo de la Prueba: Se asegura el trabajo apropiado de los requisitos funcionales, Incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

Estrategia :	<ul style="list-style-type: none"> <li>Validación y ejecución de Set de Pruebas y escenarios definidos, teniendo en cuenta flujo normal y flujos alternativos, usando datos validos e inválidos para verificar lo siguiente:             <ul style="list-style-type: none"> <li>Los resultados esperados ocurren cuando se usan datos válidos.</li> <li>Se despliegan mensajes de error cuando se usan datos inválidos.</li> <li>Cada regla de negocio es propiamente aplicada.</li> <li>Realizar set de pruebas de los requerimientos mínimos para el adecuado funcionamiento de la aplicación</li> </ul> </li> </ul>
Herramientas Requeridas	<ul style="list-style-type: none"> <li>Formato de casos de prueba funcionales</li> </ul>
Observaciones:	Se utiliza la plantilla de Casos de Pruebas.

Elaboración: (Ordóñez & Guaman, 2015)

A continuación se muestra el resultado de las pruebas funcionales validadas por el usuario con las métricas utilizadas en la Tabla 34.

Tabla 34. Resultado de las Pruebas Funcionales.

Módulo	Acción	Aceptada	Rechazada
<b>Gestión de Movilización</b>	Solicitud de Movilizacion	1	
	Asginación Vehiculo	1	
	Orden de combustible	1	
	Despacho de combutible	1	
			1
<b>Gestión de Talleres</b>	Solicitud de mantenimiento	1	
	Historial de Mantenimiento	1	
	Plan de mantenimiento preventivo	1	
			1
<b>Consulta de Reportes</b>	Control de mantenimiento	1	
	Provisión de combustibles	1	
	Matricula y soat	1	
	Ordenes de Movilizacion	1	
	Novedades	1	
<b>Administración</b>	Registro de usuarios	1	
	Asignacion vehiculo chofer	1	
	Registro Licencia	1	
	Registro combustible	1	
	Registro Gasolinera	1	
	Registro vehiculo	1	
	Registro conductor	1	

Elaboración: (Ordóñez & Guaman, 2015)

A continuación se vizualiza los resultados de las pruebas funcionales en la Figura 21.



Figura 21. Resultado de las Pruebas Funcionales  
Elaboración: (Ordóñez & Guaman, 2015)

Para más información sobre los resultados las pruebas funcionales que se visualizan **anexo H.**

**CAPITULO V**  
**CONCLUSIONES, RECOMENDACIONES**

## CONCLUSIONES

Al dar por terminado el proyecto se concluye con los siguientes ítems que detallan el alcance esperado para cada objetivo planteado:

- Se automatizó un área crítica de la Coordinación de Gestión Vehicular del GPL, al cambiar los registros de archivos manuales por registro en Base de Datos y un Sistema Informático que permite manipular la información de una forma más efectiva y poder controlar el parque automotor del Gobierno Provincial de Loja.
- La implementación del Sistema de Gestión Vehicular permite el control de consumo de combustible de cada vehículo, obteniendo reportes diarios, semanales y mensuales.
- El Sistema de Gestión Vehicular permite programar los mantenimientos y reparaciones de los vehículos de acuerdo al kilometraje recorrido, para el control en la asignación de repuestos y lubricantes.
- La integración de la metodología de desarrollo RUP permite dar cumplimiento con la entrega de los artefactos determinados para cada fase de desarrollo del sistema, lo que da como resultado una aplicación escalable.
- La implementación de la arquitectura de software permite cumplir con los requerimientos no funcionales del sistema como son: estabilidad, funcionalidad, seguridad y de rendimiento.
- La validación con el plan de pruebas permite determinar la calidad y el desempeño de cada uno de los requerimientos funcionales y no funcionales del sistema.
- La aplicación permite a los usuarios tener un control diario de la asignación de combustible para cada vehículo generando reportes diarios, semanales y mensuales.
- El sistema permite a los usuarios tener control del mantenimiento y repuestos por cada vehículo.

## RECOMENDACIONES

Las recomendaciones de la solución del software se detallan a continuación:

- Es muy importante mantener actualizado el registro de conductores y vehículos ya que esta información es utilizada en la mayoría de los procesos que se realizan en el sistema.
- Mantener los flujos de los procesos levantados para el desarrollo del sistema, ya que como entidad pública el GPL, sus procesos no varían en gran porcentaje, para mantener el correcto funcionamiento de la aplicación.
- En caso de futuras actualizaciones de Software se recomienda seguir utilizando el patrón arquitectónico MVC, permitiendo tener código estructurado y así garantizar un sistema flexible y escalable de fácil mantenimiento.
- Para llegar al éxito total del desarrollo de software se recomienda realizar un análisis correctivo y detallado en la fase de Inicio donde se abarquen todas las necesidades del cliente y estar en iteración constante con los mismos. Ya que de esto dependerá que al final el software cumpla con todos los requerimientos requeridos.

## BIBLIOGRAFÍA

- Acosta, O. (3 de Junio de 2008). *Cuadro Comparativo de Metodologías de Desarrollo*. Obtenido de Temas de programación . Net Microsoft: <http://oacosta.blogspot.com/2008/06/cuadro-comparativo-de-metodologias-de.html>
- Alegsa, L. (2010). Definición de Arquitectura de software. *Diccionario de Informática y Tecnología*.
- Alvarez, M. Á. (2009). *Entender el flujo de aplicación de codeigniter*. Obtenido de Entender el flujo de aplicación de codeigniter: <http://www.desarrolloweb.com/articulos/flujo-aplicacion-codeigniter.html>
- Arcaya, L. (2011). *Sistema de Información Cliente/Servidor con Tecnología Web para los procesos de matrículas y trámites de certificación de la escuela Nacional de Estadística e Informática del INEI - Tacna - 2011*. Tacna.
- Ayala, W. J. (11 de Abril de 2011). Patrones de Arquitectura. *Patrones de Arquitectura*.
- Bourque, P., & Fairley, R. (2014). SWEBOOK V3.0.
- Buschmann. (1996). *Oriented software Architecture, A System of Patterns*.
- Camacho, E., Cardeso, F., & Nuñez, G. (2004). *Arquitecturas de Software Guías de Estudio*.
- Centro de Desarrollo Territorial Holguín, U. (2015). *EcuRed*. Obtenido de EcuRed: [http://www.ecured.cu/index.php/Sistema\\_Gestor\\_de\\_Base\\_de\\_Datos](http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos)
- Departamento de Lenguajes y Sistemas Informaticos, D. (2002). *Productos para desarrollar aplicaciones web: ASP, CFM, JSP y PHP*.
- Duarte, M. P., & Pérez, I. M. (2012). *Programación en PHP a través de ejemplos*.
- Echeverría, Á. M. (2002). *Manual Práctico de HTML*.
- EcuRed. (2015). *EcuRed Conocimientos con todos y para todos*. Obtenido de [http://www.ecured.cu/index.php/Patr%C3%B3n\\_Modelo\\_Vista\\_Controlador](http://www.ecured.cu/index.php/Patr%C3%B3n_Modelo_Vista_Controlador)
- Gracían, S. D. (2006). *Flujo de Trabajo del Análisis y Diseño del RUP*. Obtenido de <http://repository.uaeh.edu.mx/bitstream/bitstream/handle/123456789/11150/Flujo%20de%20trabajo%20del%20an%C3%A1lisis%20y%20dise%C3%B1o%20de%20Rup.pdf?sequence=1>
- Isaias Carrillo. (2008). Metodología de desarrollo de Software. *Metodología de desarrollo de Software, 7*.
- Isaías, C., Rodrigo, P., & Aureliano, R. (2008). Metodología de Desarrollo de Software. *Metodología de Desarrollo de Software*.
- Laboratorio Nacional de Calidad del Software de INTECO, I. (2009). *Ingeniería del Software: metodologías y ciclos de vida*. Obtenido de [https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&uact=8&ved=0ahUKEwiYxZ3ShtTJAhWBC5AKHZ2BBU4QFgg\\_MAU&url=https%3A%2F%2Fwww.incibe.es%2Ffile%2FN85W1ZWFHifRgUc\\_oY8\\_Xg&usq=AFQjCNENnI5-oTppq3s99afB4Lisiq4tJ3w&sig2=LdWfa5QwaEXEYfIDLAhP](https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&uact=8&ved=0ahUKEwiYxZ3ShtTJAhWBC5AKHZ2BBU4QFgg_MAU&url=https%3A%2F%2Fwww.incibe.es%2Ffile%2FN85W1ZWFHifRgUc_oY8_Xg&usq=AFQjCNENnI5-oTppq3s99afB4Lisiq4tJ3w&sig2=LdWfa5QwaEXEYfIDLAhP)
- Laya, J. (2014). *Metodología Rup*. Obtenido de <http://www.jlaya.com/metodologia-rup-c-juanlaya/>
- Ledesma, R. (2008). Introducción al Bootstrap. *Tutorials in Quantitative Tutorials for Psychology, 51-60*.
- Len Bass, P. C. (2003). *Software Architecture in Practice*. Addison-Wesley Professional.
- Letelier, P., & Penadés, C. (Abril de 2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.
- Loira, R. d. (2006). *Ingeniería del Software. Séptima edición*. Madrid España.
- Luna, E., & Freire, L. (2014). Sistema de Información y Control de Usuarios (SIGGER) para el centro gerontológico municipal "DR. ARSENIÓ DE LA TORRE MERCADILLO".
- Martín, P. R. (2001). *HTML*.
- Martínez, R. (2009). *PostgreSQL-es*. Obtenido de PostgreSQL-es: <http://www.postgresql.org.es/node/297?page=3>
- Mestras, J. P. (2013). Bootstrap 3.0. *Aplicaciones Web/Sistemas Web*.

- Metzner, C., & Niño, N. (2012). El Proceso de Desarrollo RUP-GDIS. *Universidad Central de Venezuela*.
- Microsoft. (2014). *Ejecutar un Procedimiento Almacenado*. Obtenido de Ejecutar un Procedimiento Almacenado: <https://msdn.microsoft.com/es-es/library/ms189915%28v=sql.120%29.aspx>
- Mitaritonna, A. D. (2010). *Tesis de Maestría en Ingeniería en Sistemas de Información*. Obtenido de [http://www.caminandoutopias.org.ar/accesible/Tesis\\_ADM\\_v1.0\\_final.pdf](http://www.caminandoutopias.org.ar/accesible/Tesis_ADM_v1.0_final.pdf)
- Navarrete, T. (s.f.). *El lenguaje JavaScript (Introducción a la programación)*.
- Ordóñez, Y., & Guaman, A. (2015). Proceso de Negocio. Loja, Loja, Ecuador.
- Peinado, F. (2010). *PHP Introducción y sintaxis*.
- Peinado, F. (2014). *Tecnologías WEB*.
- Pérez E., J. (2015). *Introducción a CSS*.
- Pérez Eguíluz, J. (2008). *Introducción a JavaScript*.
- Pressman, R. S. (2005). *Ingeniería de Software*.
- Romano, J. M. (2015). *Desarrollo de sitios web con PHP y MySQL*.
- Rozanski, N., & Woods, E. (2012). *Software Systems Architecture*.
- Sánchez, J. (2003). *JavaScript manual de referencia*.
- Sánchez, J. (2013). CSS. En J. Sánchez, *Lenguajes de marcas y sistemas de gestión de información*.
- Sarria, F. A. (2006). Sistemas de Gestión de Bases de Datos y SIG. En F. A. Sarria, *Sistemas de Gestión de Bases de Datos y SIG*.
- Sibaja, J. O. (2015). *Cuadro comparativo entre metodologías de desarrollo de aplicaciones web*. Obtenido de academia.edu.
- Sotware, R. (2002). Rational Unified Process.
- Suárez, J. d. (27 de Noviembre de 2008). *Arquitectura de software: Arquitectura de software: Universidad Autónoma de Baja California Facultad de Ingeniería*. (2004). Obtenido de Servidor YAQUI: <http://yaqui.mx.l.uabc.mx/~molguin/as/RUP.htm>
- Universidad de Alicante, U. (2015). *Universidad de Alicante*. Obtenido de <http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Universidad Técnica Federico Santa María, U. (2014). *UTFSM CSRG*. Obtenido de UTFSM CSRG: [http://csrg.inf.utfsm.cl/~jfuentes/\\_build/html/index.html](http://csrg.inf.utfsm.cl/~jfuentes/_build/html/index.html)
- Zelaya, C. E. (2001). *Manual Básico de HTML*.

## **ANEXOS**

## ANEXO A: ENTREVISTAS

Entrevista al departamento de Movilización		ENTREVISTA N° 002 15-01-2015
Solicitado por:	Andrea Guamán Yonder Ordóñez	
Asistentes:	Francisco Palacios Alba Barriga Andrea Guamán Yonder Ordóñez	
<b>Entrevista</b>		
Notas	LA PRESENTE ENTREVISTA FUE DISEÑADA Y PLANTEADA PARA EL LEVANTAMIENTO DE INFORMACIÓN QUE AYUDE A LA ESPECIFICACIÓN DE REQUERIMIENTOS.	
Tema	Detalle	
Proceso del sistema actual y el usuario.	Actualmente no se utiliza el sistema porque no posee algunas funcionalidades que son necesarias para los usuarios, la aplicación no es adaptable al proceso que se lleva por parte de la dirección de gestión vehicular.	
Proceso para emitir la orden de movilización.	Actualmente se lleva el proceso manualmente que parte inicialmente cuando un empleado solicita se le autorice un vehículo para un trabajo en particular, entonces la solicitud registra los datos del solicitante y vehículo, pasa a ser entregada a la directora del departamento de gestión vehicular, cuando la persona encargada lo aprueba la solicitud pasa al administrador del departamento para que emita la orden de combustible y viáticos para el vehículo y chofer designado. También es necesario disponer de dos tipos de solicitudes para la movilización, una en días laborables y la otra para días no laborables.	
Asignación de Vehículos	El Proceso se lleva manualmente, el proceso es parte de la orden de movilización, ya que permite asignar un chofer a un vehículo, o al mismo vehículo asignarle otro chofer, conociendo el estado de disponibilidad tanto del vehículo como del chofer. Además se necesitan saber la fecha de emisión de matrícula vehicular y SOAT para poder determinar la fecha de expiración dichos documentos y se alerte al administrador para que proceda a realizar el cambio de los mismos.	
Reporte para la Contraloría General del Estado.	<p>Actualmente registran la información en documentos Excel y realizan la impresión de los mismos como reportes, los reportes que necesitan son los siguientes:</p> <ul style="list-style-type: none"> <li>○ Inventario de vehículos, accesorios y herramientas.</li> <li>○ Control de mantenimiento.</li> <li>○ Control vigencia de la matrícula vehicular y del seguro obligatorio de accidentes de tránsito, SOAT.</li> <li>○ Ordenes de movilización.</li> <li>○ Informes diarios de movilización de cada vehículo, que incluya kilometraje que marca el odómetro.</li> <li>○ Partes de novedades y accidentes.</li> <li>○ Control de lubricantes, combustible y lubricantes.</li> <li>○ Registro de entrada y salida de vehículos.</li> <li>○ Libro novedades.</li> <li>○ Actas de entrega de recepción de vehículos.</li> </ul>	
Entrevista al departamento de Mantenimiento Talleres		ENTREVISTA N° 002 15-01-2015
Solicitado por:	Andrea Guamán Yonder Ordóñez	

Asistentes:	Rubén Vicuña Andrea Guamán Yonder Ordóñez
<b>Entrevista</b>	
Notas	LA PRESENTE ENTREVISTA FUE DISEÑADA Y PLANTEADA PARA EL LEVANTAMIENTO DE INFORMACIÓN QUE AYUDE A LA ESPECIFICACIÓN DE REQUERIMIENTOS.
Tema	Detalle
Proceso del sistema actual y el usuario.	Actualmente no se utiliza el sistema porque no posee algunas funcionalidades que son necesarias para los usuarios, la aplicación no es adaptable al proceso que se lleva por parte de la dirección de gestión vehicular.
Proceso para emitir la orden de mantenimiento correctivo.	<p>Actualmente se lleva el proceso manualmente que parte inicialmente cuando un conductor solicita el mantenimiento correctivo del vehículo que tiene a su cargo, por consiguiente el realiza la solicitud con sus datos y los del vehículo, lo entrega al jefe de mantenimiento.</p> <p>El jefe de mantenimiento se encarga de las revisiones del vehículo y es quien decide si se aprueba. En caso de ser aprobada la solicitud se procede a entregar la orden de repuestos/lubricantes necesarios que fueron identificados. Esta orden se emite a bodega, aquí se realiza el chequeo de la existencia de los repuestos o lubricantes, en caso de que no existan se informa al jefe de mantenimiento para que realice la adquisición fuera rigiéndose en la ley de régimen especial, que se aplica a las compras de bienes a entidades públicas del país.</p>
Mantenimiento preventivo.	<p>Es un proceso manual, en el cuál el jefe de mantenimiento realiza chequeos a todos los vehículos rigiéndose en el kilometraje para realizar el cambio de partes a los vehículos. El jefe de mantenimiento realiza estos cambios a los vehículos de acuerdo a una tabla de kilometraje vehicular que va de 5 en 5, 10 en 10, 30 en 30 y de 60 en 60.</p> <p>Se necesita que el sistema envíe alertas continuas al jefe de mantenimiento para que proceda a realizar el mantenimiento preventivo a los vehículos, de acuerdo a la serie de la tabla y del último mantenimiento realizado a los vehículos.</p>
Emisión de la orden de repuestos/lubricantes	<p>El proceso se lleva manualmente, después de aprobarse la solicitud de mantenimiento, se emite la orden en el cual se detalla los códigos de los repuestos en bodega, la cantidad necesaria, esta orden es enviada a bodega para su despacho y debe quedar otra de constancia para la contraloría.</p> <p>El sistema actual debería mostrar la información de estos repuestos para que pueda despachar los repuestos o lubricantes, sin tener que movilizarse a bodega a verificar.</p>

## **ANEXO B: POLITICAS**

El Gobierno Provincial de Loja como entidad pública se rige al ***Acuerdo N° 005 de la Contraloría General del Estado para el control de los vehículos del sector público y las entidades de derecho privado que disponen de recursos públicos.***

### **Art. 5. Movilización de los vehículos oficiales y excepciones.-**

Ningún vehículo oficial podrá circular sin la respectiva orden de movilización y con justificación expresa de la necesidad institucional.

### **Art. 6. - Solicitud de orden de movilización.-**

El titular de la unidad administrativa o su delegado, autorizarán el desplazamiento del servidor en días y horas laborables; para tal propósito, la orden de movilización se tramitará en formatos propios de la entidad.

La autorización previa al desplazamiento de los servidores fuera de la jornada ordinaria de trabajo, en días feriados y/o fines de semana o que implique el pago de viáticos y/o subsistencias, será otorgada por la máxima autoridad o su delegado; en tal caso, la orden de movilización se tramitará únicamente mediante el aplicativo cgeMovilización, publicado en la página web de la Contraloría General del Estado.

### **Art. 7. - Registros y estadísticas.-**

La unidad encargada de la administración de los vehículos, para fines de control y mantenimiento, deberá llevar los siguientes registros:

- a. Inventario de vehículos, accesorios y herramientas.
- b. Control de mantenimiento.
- c. Control de vigencia de la matrícula vehicular y del Seguro Obligatorio de Accidentes de Tránsito, SOAT
- d. Órdenes de movilización.
- e. Informes diarios de movilización de cada vehículo, que incluya el kilometraje que marca el odómetro.
- f. Partes de novedades y accidentes.
- g. Control de lubricantes, combustibles y repuestos.
- h. Órdenes de provisión de combustible y lubricantes.
- i. Registro de entrada y salida de vehículos.
- j. Libro de novedades.
- k. Actas de entrega recepción de vehículos.

#### **Art. 8.- Custodia del vehículo.-**

El conductor asignado es responsable de la custodia del vehículo durante el tiempo que dure la comisión. Las llaves del automotor permanecerán en su poder.

Cuando los vehículos se destinen a comisión es fuera de las horas de la jornada ordinaria de trabajo, días feriados y/o fines de semana, o cuando implique el pago de viáticos y/o subsistencias, la responsabilidad por el cuidado, protección y mantenimiento básico del mismo, corresponderá al conductor. Si las labores a cumplirse sobrepasan el plazo de treinta días, con los justificativos correspondientes, se asignará el vehículo mediante acta de entrega recepción, suscrita por el encargado de la unidad de transportes de la entidad y el conductor autorizado.

El conductor autorizado es responsable de verificar la vigencia de la matrícula vehicular y del Seguro Obligatorio de Accidentes de Tránsito, SOAT, las condiciones del vehículo y que sus partes y accesorios se encuentren completos al momento de recibirlo, de lo cual dejará constancia en el formulario correspondiente de entrada y salida de vehículos, debiendo entregarlo en igual forma.

Al término de la jornada de trabajo o de la comisión, los vehículos deben guardarse en los patios de la institución o en los garajes autorizados, cuya ubicación será informada por escrito y con antelación a la Contraloría General del Estado.

En los días y horas no laborables, el guardia de turno o el servidor encargado, registrará la hora de entrada o salida de los vehículos, con la firma del conductor y no permitirá la salida de los automotores sin la correspondiente orden de movilización.

#### **Art. 9.- Mantenimiento preventivo y correctivo.-**

El mantenimiento y la reparación de los vehículos, debe efectuarse en los talleres de la institución, en caso de haberlos. El mantenimiento preventivo, periódico y programado es responsabilidad del encargado o responsable de la unidad de transportes y del conductor.

Para las acciones de mantenimiento y/o reparación se utilizarán formularios preestablecidos, en los que deben constar los datos de la última revisión o reparación y el aviso de la fecha en que debe efectuarse el siguiente control. El responsable de la unidad de transportes y el conductor del vehículo entregado mediante acta de entrega recepción, son corresponsables del mantenimiento, custodia y uso del mismo

Es obligación del conductor, revisar diariamente el vehículo asignado, observando los niveles de fluidos, presión y estado de neumáticos, así como los accesorios y el aseo interior y exterior del vehículo. Para atender el cambio de fluidos, aceites o lubricantes, o de repuestos, se utilizarán formularios específicos elaborados por la propia entidad.

Los mecánicos de la institución, en caso de haberlos, realizarán trabajos ordinarios de reparación de partes mecánicas, eléctricas y electrónicas que se encuentren en mal estado y las sustituciones necesarias para evitar la paralización del vehículo.

Los vehículos pueden repararse con profesionales mecánicos particulares, bajo las previsiones de la Ley Orgánica del Sistema Nacional de Contratación Pública, únicamente en los siguientes casos:

1. Por falta de personal especializado en la entidad.
2. Insuficiencia de equipos herramientas y/o accesorios.
3. Convenios de garantía de uso con la firma o casa en la que se adquirió el automotor.

**Art. 10.- Abastecimiento de combustible y lubricantes.-**

El servidor responsable de la unidad de transportes, debe establecer el control de consumo del combustible, con referencia hecha al rendimiento medio de kilómetros por litro, de acuerdo a cada tipo de vehículo, modelo, entre otros.

Para el abastecimiento de combustible, se utilizarán formularios con formato específico para asegurar el debido control.

**ANEXO C: DOCUMENTO DE VISIÓN**

***Sistema de Gestión Vehicular***  
***Documento de Visión***

## Información del Documento

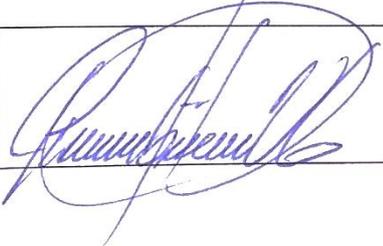
TÍTULO: Documento de Visión del Sistema  
 SUBTÍTULO: Sistema de Gestión Vehicular  
 VERSIÓN: 1.2  
 ARCHIVO: certificadox  
 AUTOR: Andrea Guamán  
 Yonder Ordóñez  
 ESTADO: Final

## Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	2015-01-01	Andrea Guamán- Yonder Ordóñez	Creación del Documento.
1.1	2015-07-06	Andrea Guamán- Yonder Ordóñez	Corrección del documento de Visión.
1.2	2015-09-10	Andrea Guamán- Yonder Ordóñez	Documento Final

## Firmas y Aprobaciones

REVISADO POR:	Ing. Pablo Vallejo Coordinador De Infraestructura Tecnología	
FECHA:	[2015-09-08] Firma:	

REVISADO POR:	Ing. Rubén Vicuña Coordinador de Gestión Vehicular	
FECHA:	[2015-09-08] Firma:	



## **1. Introducción**

### **1.1 Propósito**

El propósito de este documento es recoger, analizar y definir todas las necesidades, y las características de implementación al software de gestión vehicular. Este documento también se centra en la funcionalidad requerida de los usuarios. Esta funcionalidad se basa principalmente en la administración del parque automotor, gestión de movilización y mantenimiento. Los detalles de cada requerimiento del sistema se centra el en documento de ERS.

### **1.2 Alcance**

El sistema de Gestión vehicular permitirá, a los usuarios encargados controlar todo lo referente la gestión de movilización (orden de movilización, ordenes de combustible, administración de repuestos, de vehículos, conductores, ordenes de mantenimiento, etc.). También permitirá a cualquier empleado gestionar su solicitud de movilización.

### **1.3 Definiciones, Acrónimos, y Abreviaciones**

GPL: Gobierno Provincial de Loja

ERS: Especificación de Requerimiento de Software

## **2. Posicionamiento**

### **2.1 Oportunidad de Negocio**

El sistema permitirá poder llevar un

- Control logístico
  - Análisis de la información de los vehículos.
  - Análisis de la información de los conductores.
  - Cumplimiento de horarios y asignaciones de trabajo, contratos y rutas.
  - Reducción del mantenimiento de la flota.
  
- Reducción de costos
  - Aumento de la vida útil del vehículo.
  - Mayor tiempo entre los mantenimientos.

## 2.2 Sentencia que define el problema

El problema de	<p>La emisión de las órdenes de movilización y de mantenimiento es de forma manual lo que representa pérdida de grandes tiempos en generar cada una de las órdenes.</p> <p>No se almacena la información para generar reportes para la Contraloría General del Estado.</p> <p>No existe el plan de mantenimiento preventivo para el cambio de las partes de los vehículos de forma oportuna para la toma de decisiones.</p>
afecta a	<p>Departamento de Gestión vehicular,</p> <p>Departamento de mantenimiento y talleres,</p> <p>Empleados.</p>
El impacto asociado es	No hay un orden de toda la documentación de los proceso de gestión de movilización y mantenimiento evitando generar reportes.
una adecuada solución sería	<p>Se desarrollara e implementa el Sistema Gestión Vehicular con los siguientes módulos: Gestión de Movilización, Talleres, Reportes y Administración, que nos permiten:</p> <ul style="list-style-type: none"> <li>• Disponer de la información en tiempo real y accesible desde cualquier lugar, para realizar un control y toma de decisiones acordes a la nueva regulación del control vehicular.</li> <li>• Automatizar el proceso actual del control de la flota de vehículos de la Institución.</li> </ul> <p>Mejorar los procesos de control de combustible, mantenimiento y reparación de vehículos y monitoreo de los mismos.</p>
Para	<p>Los usuarios del departamento de movilización dispondrán de la información en tiempo real para la asignación de un vehículo a una tarea.</p> <p>Los usuarios del departamento de talleres dispondrán de la información de cada vehículo, sus partes y repuestos, para determinar si se realiza un mantenimiento preventivo o correctivo.</p> <p>Los empleados podrán realizar una solicitud de movilización en cualquier momento y en el menor tiempo posible.</p>
Quienes	Consumen la información y toman las decisiones.
El nombre del producto	Sistema de Gestión Vehicular para el Gobierno Provincial de Loja.
Que	<p>Permite el registro de solicitudes y ordenes de movilización, mantenimiento, viáticos, combustibles, repuestos, la generación de reportes y la administración de la información.</p> <p>Validar la información ingresada y dar sugerencia.</p>
No como	Registran de forma manual todas las solicitudes y no hay un orden
Nuestro producto	Permite registrar las solicitudes mediante la aplicación web.

## 2.3 Sentencia que define la posición del producto.

Para	<p>Los usuarios del departamento de movilización que dispondrán de la información en tiempo real para la asignación de un vehículo a una tarea.</p> <p>Los usuarios del departamento de talleres que dispondrá de la información del cada vehículo, sus partes y repuestos, para poder asignar un mantenimiento preventivo y correctivo.</p> <p>Los empleados que podrán realizar una solicitud de movilización a cualquier momento y en el menor tiempo posible.</p>
Quienes	Son lo que consumen la información y toman las respectivas decisiones.
El nombre del producto	Sistema de gestión vehicular
Que	<p>Permite el registro de solicitudes, ordenes de mantenimiento, control de combustibles y lubricantes, y reportes.</p> <p>Validar la información ingresada y dar sugerencia.</p>

No como	Registran en hoja de papel todas las solicitudes y no tener de forma ordenada y accesible.
Nuestro producto	Permitirá registrar mediante una aplicación web que almacenará en una base de datos.

### 3 Descripción de Stakeholders (Participantes en el Proyecto) y Usuarios

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto.

#### 3.1 Resumen de Stakeholders

Nombre	Descripción	Responsabilidades
Jairo Silva Pablo Vallejo	Representantes globales del Gobierno Provincial de Loja.	El stakeholder realiza: Seguimiento del desarrollo del proyecto. Aprueba requisitos y funcionalidades

#### 3.2 Resumen de Usuarios

Nombre	Descripción	Stakeholder
Jefa de la Coordinación de Gestión Vehicular	Responsable de la coordinación de Gestión Vehicular. Encargada de la aprobación de las solicitudes de movilización y mantenimiento.	Rubén Vicuña
Administrador del departamento de Movilización	Responsable de emitir la orden de movilización, viáticos y combustible.	<b>Francisco Palacios</b>
Mecánico del departamento de Talleres	Responsable de emitir la orden de mantenimiento, repuestos y lubricantes.	<b>Rubén Vicuña</b>
Solicitante	Solicitante de una orden de movilización y mantenimiento	<b>Empleados del Gobierno Provincial de Loja</b>

#### 3.3 Entorno de usuario

- Los usuarios ingresaran al sitio virtual utilizando su usuario y contraseña, tras este paso entrarán a la parte de la aplicación diseñada para cada uno según su rol.
- Reportes generados con la aplicación Microsoft Excel.
- Las impresiones de las ordenas de movilización, viáticos, combustible y mantenimiento emitidos en PDF.

### 3.4 Perfil de los Stakeholders.

#### 3.4.1 Representante del área técnica y sistemas de información

Representante	Ings. Pablo Vallejo y Jairo Silva
Descripción	Representantes del departamento de Tecnología.
Tipo	Expertos en sistemas.
Responsabilidades	Encargado de mostrar las necesidades de cada usuario del sistema. Además, lleva a cabo un seguimiento del desarrollo del proyecto y aprobación de los requisitos y funcionalidades del sistema
Grado de participación	Revisión de requerimientos, estructura del sistema

### 3.5 Perfiles de Usuario

#### 3.5.1 Jefa del departamento de gestión vehicular.

Representante	Alba Barriga
Descripción	Encargada de la aprobación de las solicitudes de movilización y mantenimiento.
Tipo	Usuario del sistema
Responsabilidades	Aprueba o desaprueba las solicitudes de movilización y mantenimiento entrantes en el sistema.
Grado de participación	Revisar funcionales del sistema.

#### 3.5.2 Administrador del departamento de Movilización

Representante	Francisco Palacios
Descripción	Responsable de emitir la orden de movilización, viáticos y combustible.
Tipo	Usuario del sistema
Responsabilidades	Emite desde el sistema las órdenes mediante PDF, con los datos correctos de los solicitantes, el motivo de la solicitud y los montos para viáticos y combustibles.
Grado de participación	Revisa que las órdenes generadas sean correctas.

#### 3.5.3 Mecánico del departamento de Gestión Vehicular

Representante	Rubén Vicuña
Descripción	Responsable de emitir la orden de mantenimiento, repuestos y lubricantes.
Tipo	Usuario del sistema
Responsabilidades	Emite desde el sistema las órdenes mediante PDF, con los datos correctos de los solicitantes, el motivo de la solicitud, el detalle de los repuestos y lubricantes solicitados, y la asignación del taller para el mantenimiento del vehículo.
Grado de participación	Revisa que las órdenes generadas sean correctas.

### 3.5.4 Solicitante

Representante	Empleados del Gobierno Provincial de Loja
Descripción	Solicitantes de las órdenes de movilización.
Tipo	Usuarios finales.
Responsabilidades	Quienes solicitan mediante su director del departamento la movilización de un vehículo.
Grado de participación	Ingresar al sistema y realizar la solicitud de movilización.

## 4 Descripción Global de Producto

### 4.1 Perspectiva del producto

La implementación de los módulos al sistema actual de gestión vehicular permitirá de forma ágil de emisión de órdenes de movilización y mantenimiento. Así, también generan reportes de los ingresados que son requeridos por la Contraloría General del Estado. Las áreas a involucrarse son la Gestión Vehicular e Inventarios.

### 4.2 Resumen de características

A continuación se mostrará un listado con los beneficios que obtendrá el cliente a partir del producto:

Beneficio del cliente	Características que lo apoyan
Mayor agilidad para realizar una solicitud de mantenimiento por parte de los empleados.	Aplicación web que podrán acceder los empleados.
Gestión automatizada de las órdenes de movilización.	Sistema de Optimización aplicación web.
Generación de reportes para la Contraloría General del Estado.	Almacenamiento de grandes cantidades de datos en la web.
Emisión de la orden de combustible.	Mediante la solicitud de movilización aprobada se puede emitir con los mismos datos la orden.
Emisión de la orden de viáticos.	Mediante la solicitud de movilización aprobada se puede emitir con los mismos datos la orden.
Asignación de vehículos.	En el menor tiempo posible asignar un conductor y acompañantes a un vehículo conociendo su disponibilidad.
Control de mantenimiento	Disminuir el mantenimiento de los vehículos cuando sea innecesario.

### 4.7 Restricciones

- La aplicación web se implementara a partir de la aplicación actualmente emita Software Congope.
- La aplicación web no implementara dispositivos GPS.
- La aplicación web se realizará en un entorno de programación PHP y servidor de base de datos Postgress.

#### **4.8 Procedencia y Prioridad**

- Ingreso de datos
- Verificación de Datos
- Generación de reportes

#### **4.9 Estándares Aplicables**

- Se aplicara la metodología de desarrollo RUP, para las fases y documentación del proyecto.
- Se aplicarán estándares de programación para la solución del software.
- Para determinar la calidad del proyecto será en función de factibilidad del usuario.

#### **4.10 Conclusiones y recomendaciones**

- La extracción de información se realiza con la ayuda de plantillas diseñadas en base a la metodología RUP.
- Las operaciones CRU son aplicables a los usuarios.
- Los reportes se elaboraran en base a las preguntas y resoluciones con los usuarios
- Toda la información generada se almacenara en una base de datos relacional

## **ANEXO D: DOCUMENTO DE REQUERIMIENTOS**

### **Sistema de Gestión Vehicular** *Especificación de Requerimientos del Software*

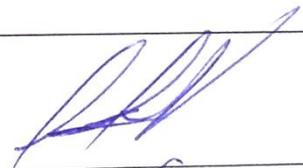
## Información del Documento

TÍTULO: Documento de especificación de requerimientos del Sistema de Gestión Vehicular  
SUBTÍTULO: Sistema de Gestión Vehicular  
VERSIÓN: 1.2  
ARCHIVO: certificadox  
AUTOR: Andrea Guamán  
Yonder Ordóñez  
ESTADO: Final

## Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	2015-01-01	Andrea Guamán- Yonder Ordóñez	Creación del Documento.
1.1	2015-07-06	Andrea Guamán- Yonder Ordóñez	Corrección del documento de Requerimientos.
1.2	2015-09-10	Andrea Guamán- Yonder Ordóñez	Documento Final

## Firmas y Aprobaciones

REVISADO POR:	Ing. Pablo Vallejo Coordinador De Infraestructura Tecnología	Firma: 
FECHA:	[2015-09-08]	

REVISADO POR:	Ing. Rubén Vicuña Coordinador de Gestión Vehicular	Firma: 
FECHA:	[2015-09-08]	



## 1. Introducción

Este documento es una Especificación de Requisitos de Software (ERS) del Sistema de Gestión Vehicular del (GPL), y contiene el resultado de la especificación de requerimientos funcionales y no funcionales de la aplicación web.

### 1.1 Propósito

El propósito del documento a desarrollar es especificar de manera clara y precisa las funcionalidades y restricciones del software del sistema de gestión vehicular (movilización y mantenimiento).

### 1.2 Ámbito

El Software a desarrollar se estará ejecutando dentro del GPL, agilizando el proceso de aprobación de las ordenes de movilización, combustible, viáticos, mantenimiento, repuestos y lubricantes, y la generación de reportes, etc. El administrador, jefe del departamento de gestión vehicular, administrador del departamento de movilización, mecánico del departamento de gestión vehicular y empleados del GPL estarán utilizando el sistema, es decir, todos los requerimientos que se especifiquen en este documento afectará a éstas entidades.

### 1.3 Definiciones, Acrónimos y Abreviaciones

**Usuario:** aquél que usa algo o que usa ordinariamente algo.

**Cliente:** permite hacer mención a la persona que accede a un producto o servicio a partir de un pago.

**Base de datos:** es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite.

**Portal web:** es una actividad que consiste en la planificación, diseño e implementación de sitios web y páginas web.

**GPL:** Gobierno Provincial de Loja, entidad pública para quien se desarrolla el sistema.

### 1.4 Referencias

- *Racional Unified Process RUP*
- Documento de Visión del Sistema
- Glosario

## 1.5 Resumen Ejecutivo

Este documento consta de tres secciones. Esta primera sección presenta una visión general de la ERS. La segunda sección contiene una descripción general del sistema a implementar y la tercera presenta la descripción detallada de todas las funcionalidades que tendrá el sistema.

## 2 Descripción General

En esta sección se presenta una descripción a alto nivel del sistema. Se presentará las características de los usuarios, las suposiciones y dependencias del sistema.

### 2.1 Perspectiva del Producto

#### 2.1.1. Interfaces de Usuario

La interfaz de usuario contará con formularios, menús, tablas y botones que construirán la aplicación que podrá visualizarse en los navegadores.

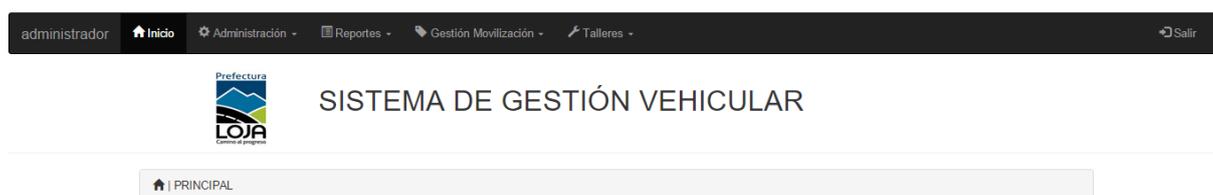


Figura 22. Interfaz Sistema de Gestión Vehicular  
Elaboración: (Ordóñez & Guaman, 2015)

#### 2.1.2. Interfaz del software

Puede ejecutarse en cualquier sistema operativo de Windows XP o superior.

### 2.2 Funciones del Producto

El sistema tendrá funciones tales como:

- Vehículos/Conductores: Listado de vehículos, listado de conductores, asignación vehículo/conductor.
- Gestión de Movilización: Orden de movilización, despacho de vehículos, administración de combustibles y reportes.
- Talleres: Orden de mantenimiento, despacho, administración de repuestos, reportes.

- Administración: Creación de usuarios, administración de combustibles, gasolineras, vehículos, conductores.

### 2.3 Características de los usuarios.

Tipo de Usuario	Administrador
Habilidades	Conocimiento en ingeniería de software
Actividades	Controlar todas las actividades del sistema

Tipo de Usuario	Cliente
Habilidades	Conocimiento sobre el desarrollo del departamento de Movilización.
Actividades	Emitir y Generas reportes e ingreso de los datos

### 2.4 Restricciones del diseño.

- El servidor deberá responderá a consultas simultaneas
- El sistema constara con la estructura tres capas

### 2.5 Supuestos y dependencias

#### Suposiciones.

- Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado dicho documento. Cualquier petición de cambios en la especificación debe ser aprobada por todas las partes y gestionada por el grupo.
- Se asume que los equipos de los usuarios cumplen con los requisitos mínimos necesarios para su producción.

#### Dependencias.

- Se hará la implementación en el mismo software entregado por el Congoje el mismo que esta desarrollo en PHP y con motor de base de datos Postgres.

## 5 Requisitos Específicos

### 5.4 Requisitos Funcionales

Gestión de Movilización	RF01	Ingresar Solicitud de movilización.
	RF02	Asignación de Vehículo Chofer
	RF03	Emitir orden de Combustible
	RF04	Despacho de Combustible
	RF05	Emitir orden de Mantenimiento
Gestión de Talleres	RF06	Emitir historial de plan de mantenimiento
	RF07	Registrar plan de mantenimiento preventivo
	RF08	Gestión de usuarios, registrar, eliminar y editar.

Gestión de Administración	RF09	Asignar vehículo chofer
	RF10	Gestión de licencias registrar, eliminar y editar
	RF11	Gestión de combustibles, registrar, eliminar y editar
	RF12	Gestión de gasolineras, registrar, eliminar y editar,
	RF13	Gestión de vehículos, registrar, eliminar y editar.
	RF14	Gestión de conductores, registrar, eliminar y editar.
	Gestión de Reportes	RF15
RF16		Reporte de conductores
RF17		Reporte de mantenimiento de vehículos
RF18		Reporte de provisión de combustibles
RF19		Reporte de matrícula y soat
RF20		Reporte de movilizaciones diarias
RF21		Reporte de novedades

Número del Requisito	RF01
Nombre del Requisito	Solicitud de Movilización
Tipo de Requisito	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Característica del Requisito	Permite el ingreso de una solicitud mediante los campos requeridos
Descripción del Requisito	Comprende el ingreso de datos como cedula donde se validara los datos del empleado que se encuentre ya registrado. Se ingresara la ruta a través de un mapa de donde se identificaran los kilometrajes recorridos, se ingresa el número y nombres de los ocupantes, se registrara fecha salida y fecha de llega, hora de salida y hora de llegada.
Prioridad del Requisito	<input checked="" type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Número del Requisito	RF02
Nombre del Requisito	Asignación de Vehículo
Tipo de Requisito	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Característica del Requisito	Permite asignar un vehículo disponible para una solicitud de movilización.
Descripción del Requisito	Se asigna un vehículo el sistema desplegara al usuario los vehículos disponibles para la fecha. Y así mismo desplegara el conductor asignado para ese vehículo.
Prioridad del Requisito	<input checked="" type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Número del Requisito	RF03
Nombre del Requisito	Generación de Orden de combustible.
Tipo de Requisito	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Característica del Requisito	Permite asignar una orden de combustible al usuario
Descripción del Requisito	Se deberá asignar el valor de combustible de acuerdo al kilometraje recorrido. Fecha de asignación, gasolinera, tipo de orden. Se debe guardar la solicitud e imprimirla.
Prioridad del Requisito	<input checked="" type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Número del Requisito	RF04
Nombre del Requisito	Despacho de combustible
Tipo de Requisito	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Característica del Requisito	Permite actualizar el valor de combustible asignado al vehículo
Descripción del Requisito	Se actualizara el valor del combustible asignado.

Prioridad del Requisito	(X) Alta/Esencial	(X) Media/Deseado	( ) Baja/Opcional
-------------------------	-------------------	-------------------	-------------------

Número del Requisito	RF05
Nombre del Requisito	Emitir orden de mantenimiento
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Permite registrar un orden de mantenimiento.
<b>Descripción del Requisito</b>	<b>Se registran datos de vehículo tipo de mantenimiento o reparación, fecha, descripción del mantenimiento, trabajo, realizado, asignación de repuestos. Se debe permitir imprimir la orden de mantenimiento y realizar la orden de repuestos.</b>
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF06
Nombre del Requisito	Emitir historial de plan de mantenimiento.
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Generar un reporte de vehículos que están en mantenimiento.
Descripción del Requisito	Visualizar el próximo mantenimiento de los vehículos y mostrar los mantenimientos que se han hecho.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF07
Nombre del Requisito	Registrar plan de mantenimiento correctivo
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Guardar plan de mantenimiento preventivo
Descripción del Requisito	Se registran datos de nombre del mantenimiento, tipos de vehículos a los que se aplica y kilometraje y tipo de vehículo.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF08
Nombre del Requisito	Gestión de usuarios.
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Administración de los usuarios registrados en el sistema
Descripción del Requisito	Permitirá la creación, actualización y eliminación de los usuarios registrados los datos ingresados serán usuarios, nombres y apellidos y rol de los menús que tendrán acceso al sistema
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF09
Nombre del Requisito	Asignación vehículo chofer.
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Administración para asignar un vehículo a un conductor.
Descripción del Requisito	Permitirá la creación de una asignación de un conductor a un vehículo previamente registrados. Esta asignación consistirá de guardar formulario que constara los registros de fecha, acta, revisión, detalles y acta.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF10
Nombre del Requisito	Registro de licencias tipo

Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Administración del tipo de licencias en el sistema
Descripción del Requisito	Crear, actualizar y eliminar el tipo de licencia, los campos a llenar nombre de la licencia.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF10
Nombre del Requisito	Registrar combustibles
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Administración de los combustibles
Descripción del Requisito	Insertar, eliminar y actualizar los combustibles que se guardan en el sistema.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF11
Nombre del Requisito	Registrar Gasolineras
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Administración de gasolineras
Descripción del Requisito	Insertar, eliminar y actualizar las gasolineras que se guardan en el sistema.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF12
Nombre del Requisito	Registrar vehículos
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Administración de vehículos
Descripción del Requisito	Insertar, eliminar y actualizar los vehículos que se guardan en el sistema.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF13
Nombre del Requisito	Registrar vehículos
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Administración de vehículos
Descripción del Requisito	Insertar, eliminar y actualizar los vehículos que se guardan en el sistema.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF14
Nombre del Requisito	Registrar conductores
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Administración de conductores
Descripción del Requisito	Insertar, eliminar y actualizar los conductores que se guardan en el sistema.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF15
Nombre del Requisito	Reporte de listado de vehículos

Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Generación de reporte en Excel de vehículos
Descripción del Requisito	Mostrar datos de vehículos características que tienes q ser exportados en Excel.
Prioridad del Requisito	(X) Alta/Eencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF16
Nombre del Requisito	Reporte de listado de conductores
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Generación de reporte en Excel de conductores
Descripción del Requisito	Mostrar datos del conductor características que tienes q ser exportados en Excel.
Prioridad del Requisito	(X) Alta/Eencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF17
Nombre del Requisito	Reporte de mantenimiento de vehículos
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Generación de reporte en Excel de mantenimiento
Descripción del Requisito	Mostrar datos de mantenimiento de vehículos realizados por fechas diarios, mensuales, semanales
Prioridad del Requisito	(X) Alta/Eencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF18
Nombre del Requisito	Reporte provisión de combustibles.
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Generación de reporte en Excel provisión de combustibles.
Descripción del Requisito	Sacar un dato total del combustible gastado en las órdenes de movilización.
Prioridad del Requisito	(X) Alta/Eencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF19
Nombre del Requisito	Reporte de matrícula y soat
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Generación de reporte en Excel de matrícula y soat.
Descripción del Requisito	Sacar un reporte de vencimiento de matrícula y soat de los vehículos.
Prioridad del Requisito	(X) Alta/Eencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF20
Nombre del Requisito	Reporte de movilizaciones
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Generación de reporte en Excel de movilizaciones.
Descripción del Requisito	Sacar un reporte de movilizaciones, diarias, semanales y mensuales con una búsqueda avanzada de choferes, vehículos y fechas.
Prioridad del Requisito	(X) Alta/Eencial (X) Media/Deseado ( ) Baja/Opcional

Número del Requisito	RF21
Nombre del Requisito	Reporte de novedades
Tipo de Requisito	( X )Requisito ( ) Restricción
Característica del Requisito	Generación de reporte en Excel de novedades.
Descripción del Requisito	Sacar un reporte de novedades existentes en cada movilización diaria. Con una búsqueda avanzada de fechas, conductor y vehiculos.
Prioridad del Requisito	(X) Alta/Esencial (X) Media/Deseado ( ) Baja/Opcional

## 5.5 Requisitos Suplementarios.

### 3.2.1 Requerimientos no Funcionales

Número	Requerimiento	Descripción	Prioridad
RNF1	Funcionalidad	Garantizar que los usuarios tengan una eficiencia en rapidez y ejecución de un 95% de la aplicación	5
RNF2	Usabilidad	El sistema debe ser dinámico e interactivo con el usuario.	5
RNF3	Reusabilidad	El sistema es adaptable y modificable a nuevas versiones para nuevos cambios según se requieran.	3
RNF4	Portabilidad	Capacidad del sistema de ejecutarse en diferentes plataformas o arquitecturas con mínimas modificaciones.	3
RNF5	Soportabilidad	El sistema permite de manera fácil su soporte o mantenimiento, ya que es construido y desarrollo de manera sencilla.	5
RNF6	Seguridad	Se puede consultar y actualizar la información simultáneamente.	5

## **ANEXO E: ESPECIFICACION DE CASOS DE USO**

### **Sistema de Gestión Vehicular** *Especificación de Casos de Uso.*

## Información del Documento

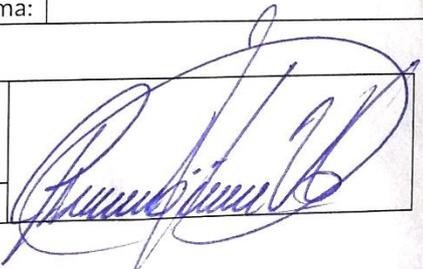
TÍTULO: Documento de casos de uso  
SUBTÍTULO: Sistema de Gestión Vehicular  
VERSIÓN: 1.2  
ARCHIVO: certificadox  
AUTOR: Andrea Guamán  
Yonder Ordóñez  
ESTADO: Final

## Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	2015-01-01	Andrea Guamán- Yonder Ordóñez	Creación del Documento.
1.1	2015-07-06	Andrea Guamán- Yonder Ordóñez	Corrección del documento de casos de uso.
1.2	2015-09-10	Andrea Guamán- Yonder Ordóñez	Documento Final

## Firmas y Aprobaciones

REVISADO POR:	Ing. Pablo Vallejo Coordinador De Infraestructura Tecnología	
FECHA:	[2015-09-08]	Firma:

REVISADO POR:	Ing. Rubén Vícuña Coordinador de Gestión Vehicular	
FECHA:	[2015-09-08]	Firma:



<b>ID</b>	<b>UC-01</b>		
Nombre	Gestión de Movilización		
Tipo	Implementación Obligatoria		
Prioridad	Alta		
Actores involucrados	Administrador		
<b>DESCRIPCION DEL CASO DE USO</b>			
En este caso de uso se especifica la gestión para emitir una solicitud de movilización, emitir una orden de combustible, despacho de combustible y asignar un vehículo a una solicitud de movilización.			
<b>PRECONDICIONES</b>		<b>POSTCONDICIONES</b>	
<ul style="list-style-type: none"> <li>• Usuario ha hecho el login correctamente al sistema.</li> </ul>			
<b>FLUJO NORMAL DE EVENTOS</b>			
<b>No.</b>	<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1	El usuario se autentica en el sistema		
2			El sistema presenta las opciones a) Gestión de Movilización b) Gestión de Talleres c) Reportes d) Gestión de administración
3	El administrador selección gestión de movilización		
4			Presenta una lista de opciones a) Solicitud de movilización b) Asignación vehículo c) Emitir orden de combustible d) Despacho de combustible
5	Cerrar sesión		
6			Fin caso de uso.
<b>CAMINOS ALTERNATIVOS</b>			
<b>No. actividad en flujo normal</b>	<b>No.</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
Se presenta la alternativa	Se integra al flujo normal		
	SI	1 Opción solicitud de movilización	
Solicitud De Movilización		2	Visualizar los datos del de la solitud a llenar
		3	Ingresar todos los campos obligatorios
		4	Validar los datos, y guardar si los datos son correctos se procede a asignar vehículo. Si no se vuelve a editar la solicitud
	SI	1	Opción asignar vehículos
Asignar Vehículo chofer		2	El sistema presentara las solicitudes guardadas.
		3	Se deberá escoger la solicitud
		4	El sistema presentar los vehículos disponibles para la fecha
		5	Debe seleccionar un vehículo Se guarda la orden de movilización y el sistema desplegara el formulario a imprimir.
	SI	1	Opción orden de combustible
Emitir orden de Combustible			

	El sistema desplegara las solicitudes que has sido aprobada.
El usuario selecciona solicitud	
	El sistema desplegara el formulario con los datos a llenar
Llenar el formulario	
	El sistema valida los datos, si son correctos el sistema emitirá la orden a imprimirse.

#### PUNTOS DE EXTENSIÓN

- El usuario podrá desplegar el menú si cumple con su rol de administrador

#### CRITERIOS DE ACEPTACION

- Solicitud enviada y guardada.

#### REQUERIMIENTOS NO FUNCIONALES ASOCIADOS

- Funcionalidad
- Seguridad

#### Interfaz Grafica

#### SEGUIMIENTO AL CASO DE USO

Fecha	Descripción	Gestor
15-01-2015	Primera versión preliminar del caso de uso.	Andrea Guamán-Yonder Ordoñez

**ID**

**UC-02**

Nombre	Gestión de talleres
Tipo	Implementación Obligatoria
Prioridad	Alta
Actores involucrados	Administrador

#### DESCRIPCION DEL CASO DE USO

En este caso de uso se especifica la gestión para emitir una orden de solicitud de mantenimiento, gestión de plan de mantenimiento preventivo, historial de mantenimiento vehículos.

#### PRECONDICIONES

- Usuario ha hecho el login correctamente al sistema.

#### POSTCONDICIONES

#### FLUJO NORMAL DE EVENTOS

No.	Acción del actor	Respuesta del sistema
1	El usuario se autentica en el sistema	
2		El sistema presenta las opciones 4.10.1.2Gestión de Movilización 4.10.1.3Gestión de Talleres 4.10.1.4Reportes 4.10.1.5Gestión de administración
3	El administrador selección gestión de talleres	
4		Presenta una lista de opciones a. Solicitud de mantenimiento. b. Historial de mantenimiento c. Plan de mantenimiento Preventivo
5	Cerrar sesión	
6		Fin caso de uso.

#### CAMINOS ALTERNATIVOS

No. actividad en flujo normal	No.	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo		

Solicitud de mantenimiento	normal	1	Opción solicitud de mantenimiento	
	SI	2		El sistema desplegara el formulario a llenar
		3	Ingresar todos los campos obligatorios	
		4		Validar los datos, y guardar si los datos son correctos se procede a asignar vehículo. Si no se vuelve a editar la solicitud
		5	Imprimir solicitud de mantenimiento	
		6		El sistema desplegara el formulario a imprimir.

**PUNTOS DE EXTENSIÓN**

- El usuario podrá desplegar el menú si cumple con su rol de administrador

**CRITERIOS DE ACEPTACION**

- Solicitud enviada y guardada.

**REQUERIMIENTOS NO FUNCIONALES ASOCIADOS**

- Funcionalidad
- Seguridad

**Interfaz Grafica**

**SEGUIMIENTO AL CASO DE USO**

Fecha	Descripción	Gestor
15-01-2015	Primera versión preliminar del caso de uso.	Andrea Guamán-Yonder Ordoñez

**ID UC-03**

Nombre	Gestión de Reportes
Tipo	Implementación Obligatoria
Prioridad	Alta
Actores involucrados	Administrador

**DESCRIPCION DEL CASO DE USO**

En este caso de uso se especifica la gestión para la generación de reportes en Excel para el usuario.

**PRECONDICIONES**

- Usuario ha hecho el login correctamente al sistema.

**POSTCONDICIONES**

**FLUJO NORMAL DE EVENTOS**

No.	Acción del actor	Respuesta del sistema
1	El usuario se autentica en el sistema	
2		El sistema presenta las opciones a) Gestión de Movilización b) Gestión de Talleres c) Reportes d) Gestión de administración
3	El administrador selecciona la opción Reportes	
4		Presenta una lista de opciones a) Listado de vehículos b) Listados de conductores. c) Control de mantenimiento d) Provisión de combustibles e) Matricula y soat

		f) Ordenes de movilización	
		g) Novedades	
5	Cerrar sesión		
6		Fin caso de uso.	
<b>CAMINOS ALTERNATIVOS</b>			
No. actividad en flujo normal	No	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal	.	
	SI	1 Opción solicitud de mantenimiento	
		2	El sistema desplegara el formulario a llenar . a) Fecha desde b) Fecha hasta c) Tipo d) Vehículo e) Conductor
		3 Seleccionar los campos de búsqueda	
Control de mantenimiento		4	El sistema desplegara el reporte de acuerdo a los campos ingresados
		5 Exportar a Excel	
		6	El sistema presentara el reporte descargado en Excel.
Provisión de combustibles	SI	1 Opción provisión de combustibles.	
		2	El sistema desplegara el formulario a llenar. a) Fecha desde b) Fecha hasta c) Vehículo d) Conductor
		3 Seleccionar los campos de búsqueda	
		4	El sistema desplegara el reporte de acuerdo a los campos ingresados
		5 Exportar a Excel	
		6	El sistema presentara el reporte descargado en Excel.
<b>PUNTOS DE EXTENSIÓN</b>			
• El usuario podrá desplegar el menú si cumple con su rol de administrador			
<b>CRITERIOS DE ACEPTACION</b>			
• Solicitud enviada y guardada.			
<b>REQUERIMIENTOS NO FUNCIONALES ASOCIADOS</b>			
• Funcionalidad			
• Seguridad			
<b>Interfaz Grafica</b>			
<b>SEGUIMIENTO AL CASO DE USO</b>			
Fecha	Descripción	Gestor	
15-01-2015	Primera versión preliminar del caso de uso.	Andrea Guamán-Yonder Ordoñez	

<b>ID</b>	<b>UC-04</b>		
Nombre	Gestión de Administración		
Tipo	Implementación Obligatoria		
Prioridad	Alta		
Actores involucrados	Administrador		
<b>DESCRIPCION DEL CASO DE USO</b>			
En este caso de uso se especifica la gestión para la administración de usuarios, gasolineras, combustibles, licencias, conductores y vehículos.			
<b>PRECONDICIONES</b>		<b>POSTCONDICIONES</b>	
<ul style="list-style-type: none"> <li>• Usuario ha hecho el login correctamente al sistema.</li> </ul>			
<b>FLUJO NORMAL DE EVENTOS</b>			
<b>No.</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1	El usuario se autentica en el sistema		
2		El sistema presenta las opciones <ul style="list-style-type: none"> <li>a) Gestión de Movilización</li> <li>b) Gestión de Talleres</li> <li>c) Reportes</li> <li>d) Administración</li> </ul>	
3	El administrador selecciona la opción Administración		
4		Presenta una lista de opciones <ul style="list-style-type: none"> <li>a) Usuarios</li> <li>b) Asignación, vehículo/chofer</li> <li>c) Tipo licencia</li> <li>d) Combustible</li> <li>e) Gasolinera</li> <li>f) Registrar Vehículo</li> <li>g) Registrar conductor.</li> </ul>	
5	Cerrar sesión		
6		Fin caso de uso.	
<b>CAMINOS ALTERNATIVOS</b>			
<b>No. actividad en flujo normal</b>	<b>No</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Se presenta la alternativa</b>	<b>Se integra al flujo normal</b>	.	.
	SI	1 Opción usuario	
		2	El sistema desplegara la lista de usuarios registrados en el sistema
		3 Seleccionar añadir usuario	
		4	El sistema desplegara el formulario con los datos. <ul style="list-style-type: none"> <li>a. Nombres y apellidos</li> <li>b. Nombre de usuario</li> <li>c. Contraseña</li> <li>d. Roles</li> </ul>
Registrar usuarios		5 El usuario seleccionara el acceso a los menús que tendrá acceso.	
		6	El sistema validara los datos deben por lo menos seleccionar un menú al que tendrá acceso.

Asignación Vehículo/Chofer	SI	1	Asignación vehículo Chofer.	
		2		El sistema desplegara la lista de los vehículos que están asignados y no están.
		3	Asignar chofer	
		4		El sistema desplegara el formulario a llenar  a) Nombre Chofer b) Fecha c) Acta de entrega d) Detalle e) Revisado y confirmado
		5	Guardan	
		6		El sistema validara los datos ingresados. Si los datos son ingresados correctamente se volverá paso 1.

**PUNTOS DE EXTENSIÓN**

- El usuario podrá desplegar el menú si cumple con su rol de administrador

**CRITERIOS DE ACEPTACION**

- Solicitud enviada y guardada.

**REQUERIMIENTOS NO FUNCIONALES ASOCIADOS**

- Funcionalidad
- Seguridad

**Interfaz Grafica**

**SEGUIMIENTO AL CASO DE USO**

<b>Fecha</b>	<b>Descripción</b>	<b>Gestor</b>
15-01- 2015	Primera versión preliminar del caso de uso.	Andrea Guamán-Yonder Ordoñez

## Diagrama de casos de uso

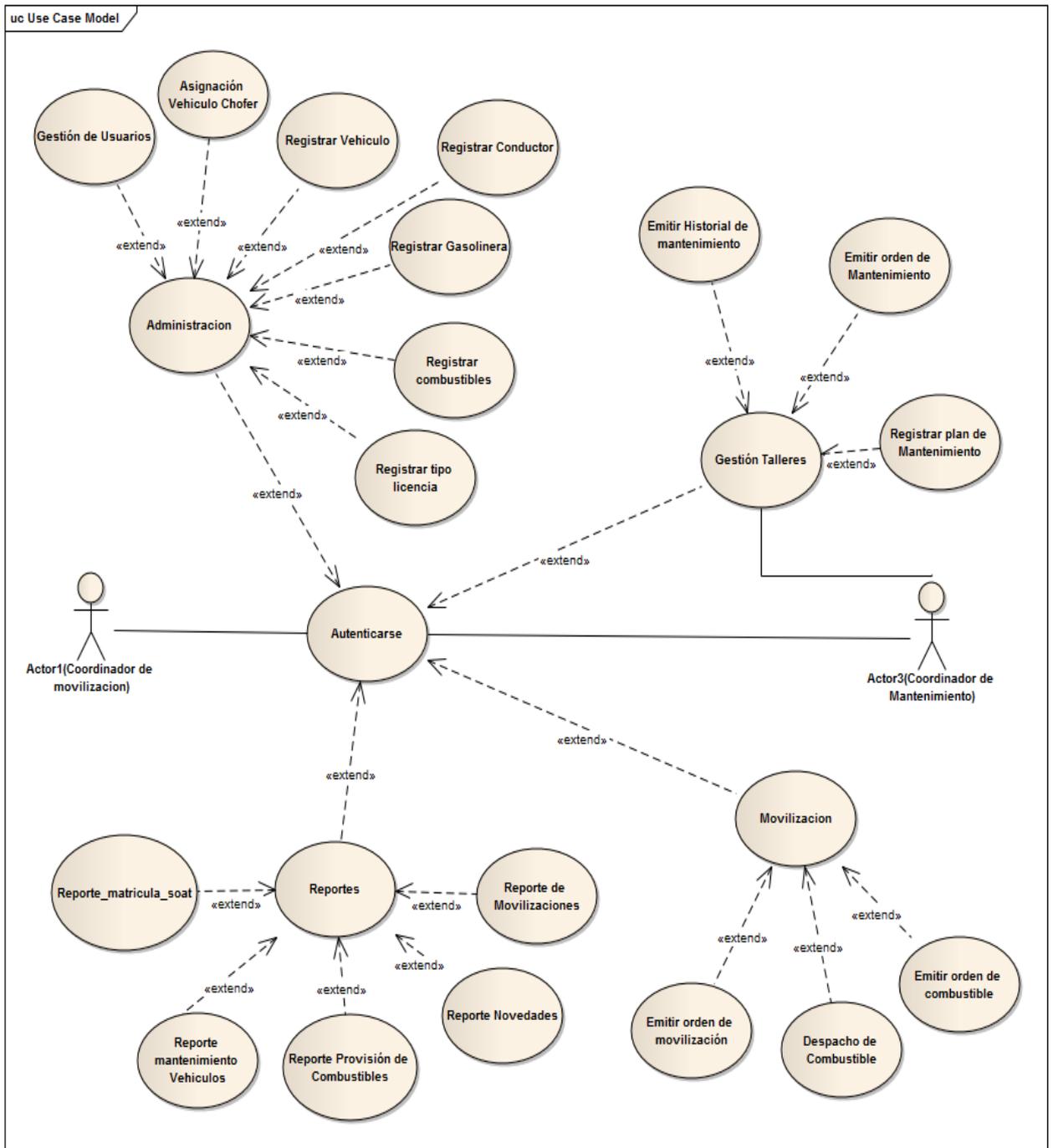


Figura 23. Diagrama de Casos de Uso. Sistema de Gestión Vehicular.  
Elaboración: (Ordóñez & Guaman, 2015)

**ANEXO F: DOCUMENTO DE ARQUITECTURA DE SOFTWARE**

**Sistema de Gestión Vehicular**  
*Documento de Arquitectura de Software*

## Información del Documento

TÍTULO: Documento de Arquitectura del Software  
SUBTÍTULO: Sistema de Gestión Vehicular  
VERSIÓN: 1.2  
ARCHIVO: certificadoox  
AUTOR: Andrea Guamán  
Yonder Ordóñez  
ESTADO: Final

## Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	2015-01-01	Andrea Guamán- Yonder Ordóñez	Creación del Documento.
1.1	2015-07-06	Andrea Guamán- Yonder Ordóñez	Corrección del documento de Arquitectura del software.
1.2	2015-09-10	Andrea Guamán- Yonder Ordóñez	Documento Final

## Firmas y Aprobaciones

REVISADO POR:	Ing. Pablo Vallejo Coordinador De Infraestructura Tecnología	
FECHA:	[2015-09-08]	

REVISADO POR:	Ing. Rubén Vicuña Coordinador de Gestión Vehicular	
FECHA:	[2015-09-08]	



## 1. Introducción

Este documento es una especificación de la Arquitectura de Software propuesta para la solución donde se especifica vistas para representar diversos aspectos del sistema.

### 1.1. Propósito

El propósito del documento es proporcionar una descripción técnica de la arquitectura del sistema, haciendo uso de las diferentes vistas para representar diversos aspectos del sistema.

### 1.3 Alcance

El presente documento detalla el tipo de arquitectura a utilizarse (MVC) Modelo Vista Controlador. Detallando cada paquete del sistema.

### 1.4 Definiciones, Acrónimos y Abreviaciones

**Actor:** Persona que interactúa con el sistema.

**UML:** Modelado de Diagramas

## 2. Representación de la Arquitectura

La arquitectura a utilizarse es el Modelo Vista Controlador es quien separa los datos del sistema, la interfaz de usuario, y la lógica de control en tres componente que a continuación se detallan.

- **Modelo:** Es una representación de la información con la cual la aplicación opera, e implementa la lógica de negocio, de tal forma es el responsable de la obtención de los datos transformándolos en datos significativos para la aplicación.
- **Vista:** Es una representación visual de los datos en la aplicación web, responsable del uso de la información de la cual se dispone para producir la interfaz de usuario.
- **Controlador:** Permite gestionar los accesos o peticiones de los usuarios el cual responde con la información solicitada.

La arquitectura se basara en el modelo 4+1 vistas que contendrá las vistas Lógica, implementación, procesos y despliegue.

A continuación se detalla cada una de las vistas con los artefactos que corresponden:

#### **Vista Lógica:**

- **Área:** Requerimientos funcionales que describen las solución del software, y se detallan los casos de uso.
- **Artefactos:** Diagrama de clases, modelo Entidad-Relación

#### **Vista Implementación:**

- **Área:** componentes de software que describen los módulos.
- **Artefactos:** Diagrama de secuencia, diagrama de estado y de colaboración.

#### **Vista de proceso:**

- **Área:** Requerimientos no funcionales.
- **Artefactos:** Diagrama de actividad.

#### **Vista de despliegue**

- **Área:** Mapeo y ubicación de los componentes.
- **Artefactos:** Diagrama de despliegue.

### **3. Metas Restricciones de las Arquitectura.**

Una de las metas principales de la arquitectura es mostrar los aspectos fundamentales que influirán en la etapa de desarrollo, además de cumplir con los atributos de calidad del sistema como: Rendimiento, Seguridad, Disponibilidad, Escalabilidad y Flexibilidad.

#### **3.1 Metas.**

- Implementar el componente Modelo del patrón arquitectónico MVC en servidores físicos independientes ya que permiten mejor flexibilidad en el acceso a datos.
- Desarrollar e implementar procedimientos almacenados a lado del servidor de base de datos postgresql, que luego serán invocadas desde el servidor PHP.
- Implementar el framework de desarrollo codeInteger que permite la conexión con la base de datos a través de su clase active record.

#### **3.2 Restricciones.**

- Los usuarios que podrán acceder al sistema son el departamento de Coordinación de Movilización.
- Las base de datos con la cual se trabajara será postgres en su versión 8.4.20
- Las configuraciones del servidor php será de acuerdo a la infraestructura hardware y software que posee el departamento de infraestructura tecnología del Gobierno Provincial de Loja.

### 3.3 Vista Lógica

**3.3.1 Diagrama de clases:** El diagrama de clases servirá para visualizar las relaciones entre las clases estas se asocian mediante sus atributos y métodos.

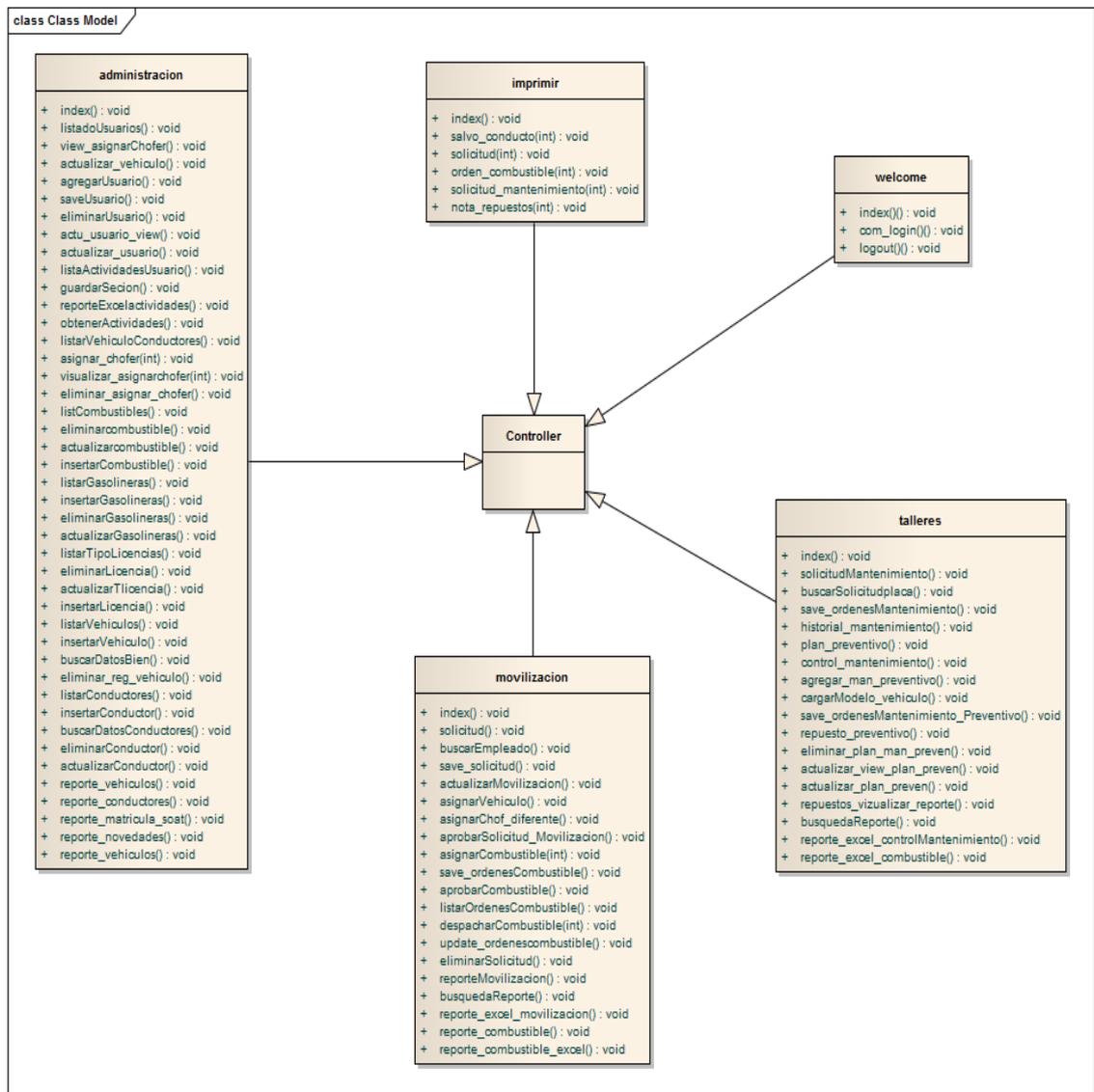


Figura 24. Diagrama de Clases  
Elaboración: (Ordóñez & Guaman, 2015)

**3.3.2 Modelo Entidad-Relación:** El modelo entidad relación permite detallar el modelo de datos de la solución del software a través de objetos con atributos y propiedades.

kilometraje	
nu_comprobante	BIGINT
cod_asig	BIGINT
cod_gasolinera	BIGINT
fecha	DATE
hij_partida	NUMERIC
hij_legada	NUMERIC
hij_recomido	NUMERIC
cantidad_galones	INTEGER
valor_entregado	NUMERIC
gasolinera_provincia	CHARACTER VARYING(30)
verificacion_gasolinera	CHARACTER(1)
cargado	CHARACTER(1)
destino	CHARACTER VARYING(100)
precio_comb	NUMERIC
cod_partida	CHARACTER VARYING(15)
tipo_combustible	CHARACTER VARYING(15)
cargado_yuelta	CHARACTER(1)
porcen_via	NUMERIC
porcen_cargado	NUMERIC
cod_ciudad	BIGINT
cod_canton	BIGINT
cod_parrquia	BIGINT
porcen_ciudad	NUMERIC
porcen_canton	NUMERIC
porcen_parrquia	NUMERIC
galones_entregados	NUMERIC
pesado	CHARACTER(1)
total	NUMERIC
eliminado	CHARACTER(1)
fec_inicio	DATE
fec_fin	DATE
total_horas	INTEGER
asignado_comp	CHARACTER(1)

vehiculos	
cod_esta_act	BIGINT
cod_gasolina	BIGINT
placa	CHARACTER VARYING(30)
factura	CHARACTER VARYING(30)
casa	CHARACTER VARYING(30)
pais	CHARACTER VARYING(30)
marca	CHARACTER VARYING(50)
modelo	CHARACTER VARYING(80)
motor	CHARACTER VARYING(80)
chasis	CHARACTER VARYING(80)
color	CHARACTER VARYING(20)
tonelaje	DOUBLE PRECISION
num_vehiculo	CHARACTER VARYING(6)
serie	CHARACTER VARYING(200)
ano_fabricacion	CHARACTER VARYING(5)
precio_adq	NUMERIC
condicion_real	CHARACTER VARYING(10)
nro_accidentes	INTEGER
rendimiento_galon	NUMERIC
asignado_chof	CHARACTER(1)
cod_vehiculo2	CHARACTER VARYING(100)
cod_vehiculo	INTEGER
imagen_vehiculo	CHARACTER VARYING(200)
imagen_vehiculo2	CHARACTER VARYING(200)
novedades	CHARACTER VARYING(200)
notificaciones_talleres	CHARACTER VARYING(200)
fecha_mat	CHARACTER VARYING(200)
fecha_sost	CHARACTER VARYING(200)
tipo	CHARACTER VARYING(100)
fecha_adquisicion	CHARACTER VARYING
vehiculo_tipo	INTEGER

movilizacion	
cod_movilizacion	INTEGER
solicitante	CHARACTER VARYING(200)
fecha_emision	CHARACTER VARYING(20)
fecha_salida	CHARACTER VARYING(20)
fecha_legada	CHARACTER VARYING(20)
motivo	TEXT
aprobado	CHARACTER(1)
asignado	CHARACTER(1)
destino	CHARACTER VARYING(200)
ocupantes	INTEGER
fecha_autorizacion	CHARACTER VARYING(20)
combu_asig	INTEGER
hora_salida	TIME(8) WITHOUT TIME ZONE
hora_retomo	TIME(8) WITHOUT TIME ZONE
responsable	CHARACTER VARYING(40)
estado	CHARACTER(1)
cod_vehiculo	INTEGER
cod_chofer	INTEGER
obs_ch_dif	CHARACTER VARYING(100)
integrantes	TEXT
num_mov1	INTEGER
negar_razon	CHARACTER VARYING(200)
departamento	CHARACTER VARYING(100)
kilometraje_recomido	INTEGER

mantenimiento_vehiculos	
cod_mantenimiento	INTEGER
fecha	CHARACTER VARYING(20)
detalle	TEXT
aprobado	CHARACTER(1)
asignado	CHARACTER(1)
taller	CHARACTER VARYING(20)
motivo	CHARACTER VARYING(100)
senior	CHARACTER VARYING(200)
despachado	CHARACTER(1)
num_factura	CHARACTER VARYING(30)
valor_factura	DOUBLE PRECISION
cod_chofer	INTEGER
cod_vehiculo	INTEGER
pedido_despachado	INTEGER
numero	INTEGER
trabajo_realizado	TEXT
kilometraje	INTEGER
id_man_preven	INTEGER

chofer	
cod_jcencia	BIGINT
nombres	TEXT
telefono	CHARACTER VARYING(10)
cedula	CHARACTER VARYING(10)
relacio_laboral	TEXT
num_accidentes	INTEGER
asignado_veh	CHARACTER(1)
cod_chofer	INTEGER
imagen_chofer	CHARACTER VARYING(200)
denominacion_cargo	CHARACTER VARYING(200)
novedades	CHARACTER VARYING(200)
bancoun	CHARACTER VARYING(100)
tipocuanta	CHARACTER VARYING(100)
numcuanta	CHARACTER VARYING(100)
correo	TEXT
departamento	TEXT

ordenes_combustible	
fecha	CHARACTER VARYING(50)
cantidad_asignada	DOUBLE PRECISION
cantidad_despachada	DOUBLE PRECISION
despachada	CHARACTER(1)
cod_gasolina	INTEGER
cod_gasolinera	INTEGER
kilometraje_recomido	CHARACTER VARYING
cod_movilizacion	INTEGER
cod_chofer	INTEGER
cod_vehiculo	INTEGER
km_salida	NUMERIC
km_retorno	NUMERIC
hora_salida	TIME(8) WITH TIME ZONE
hora_retomo	TIME(8) WITH TIME ZONE
cuenta_combus	CHARACTER VARYING(50)
cod_orden	INTEGER

vehiculos_pesados	
lugar_trabajo	TEXT
num_comp	BIGINT
cod_gasolinera	BIGINT
fecha_inicio	DATE
fecha_termino	DATE
num_horas	INTEGER
total_diesel	NUMERIC
total_super	NUMERIC
total_extra	NUMERIC
gasolinera_provincial	TEXT
observaciones	TEXT
verificacion_gasolinera	CHARACTER(1)
precio_comb	NUMERIC
cod_orden	BIGINT

taller_particular	
cod_go	INTEGER
numero	CHARACTER VARYING(30)
fecha	CHARACTER VARYING(30)
nombre	CHARACTER VARYING(200)
motivo	CHARACTER VARYING(100)
caracteristicas	CHARACTER VARYING(300)
marca	CHARACTER VARYING(50)
disco	CHARACTER VARYING(30)
cod_chofer	CHARACTER VARYING(30)
detalle	TEXT

factura_gasolinera	
fecha_emision	DATE
cod_fac	BIGINT
cod_gasolinera	BIGINT
cantidad_total	NUMERIC
total_consumido	NUMERIC
total_extra	NUMERIC
total_super	NUMERIC
total_diesel	NUMERIC
num_factura	CHARACTER VARYING(15)

asig_chofer_vehiculo	
fecha_asig	DATE
fecha_termino	DATE
descripcion	TEXT
estado	CHARACTER(1)
cod_asig	INTEGER
cod_chofer	INTEGER
cod_vehiculo	INTEGER
acta_entrega	CHARACTER VARYING(200)
revison_asig	CHARACTER(1)

historial_ordenes	
cod_hist_ord	INTEGER
cod_mantenimiento	INTEGER
cantidad	DOUBLE PRECISION
unidad	CHARACTER VARYING(20)
marca	CHARACTER VARYING(50)
cod_repuesto	INTEGER
repuesto	TEXT

tbl_audit	
nombre_tabla	CHARACTER VARYING(45)
operacion	CHARACTER(1)
valor_anterior	TEXT
valor_nuevo	TEXT
fecha	DATE
usuario	CHARACTER VARYING(45)
id_audit	INTEGER

repuesto	
cod_hist_veh	BIGINT
nombre_rep	CHARACTER VARYING(80)
cantidad	INTEGER
cod_repuesto	INTEGER
precio_unidad	CHARACTER VARYING(20)
cod_repuesto2	CHARACTER VARYING(20)

historial_vehiculo	
cod_hist_veh	BIGINT
cod_vehiculo	BIGINT
fecha	DATE
kilometro	BIGINT
origen	CHARACTER VARYING(15)
detalle	CHARACTER VARYING(80)

mantenimiento_preventivo	
id_man_preven	INTEGER
descripcion	CHARACTER VARYING(100)
cod_tipo_vehiculo	INTEGER
kilometraje1	INTEGER
cod_gasolina	INTEGER

usuarios	
id_usuario	INTEGER
nombre_usu	CHARACTER VARYING(100)
login_usu	CHARACTER VARYING(30)
passwd	CHARACTER VARYING(100)
tip_veh	INTEGER

parroquias	
cod_parroquia	INTEGER
cod_canton	INTEGER
nombre	CHARACTER VARYING(120)
kilometros	INTEGER

canton	
cod_canton	INTEGER
cod_provincia	INTEGER
nombre	CHARACTER VARYING(120)
kilometros	INTEGER

submenus	
id_submenu	INTEGER
id_menu	INTEGER
nom_sub	CHARACTER VARYING(100)
url	CHARACTER VARYING

tipo_vehiculo	
cod_tipo	BIGINT
nombre_tipo	CHARACTER VARYING(30)
deso_tipo	CHARACTER VARYING(80)
id_veh_tipo	INTEGER

departamento	
cod_departamento	INTEGER
nombre	CHARACTER VARYING(200)
director	CHARACTER VARYING(200)

partidas	
cuenta	CHARACTER VARYING(60)
descripcion	CHARACTER VARYING(150)
anio	INTEGER

combustible	
cod_gasolina	BIGINT
nombre_gasolina	TEXT
precio	NUMERIC

rol_permiso	
id_submenu	INTEGER
id_usuario	INTEGER
cod_rol_permiso	INTEGER

gasolinera	
cod_gasolinera	BIGINT
nombre_gasolinera	CHARACTER VARYING(80)

provincia	
cod_provincia	INTEGER
nombre	CHARACTER VARYING(120)

menus	
id_menus	INTEGER
nom_menu	CHARACTER VARYING(100)

veh_tipo	
id_veh_tipo	INTEGER
tipo	CHARACTER VARYING(40)

licencia_tipo	
cod_jcencia	BIGINT
nombre_licencia	TEXT

estado_actual	
cod_esta_act	BIGINT
descripcion	TEXT

Figura 25. Diagrama Entidad-Relación  
Elaboración: (Ordóñez & Guaman, 2015)

### 3.4 Vista de Implementación

#### 3.4.1 Diagramas de Secuencia.

##### a) Gestión de Movilización.

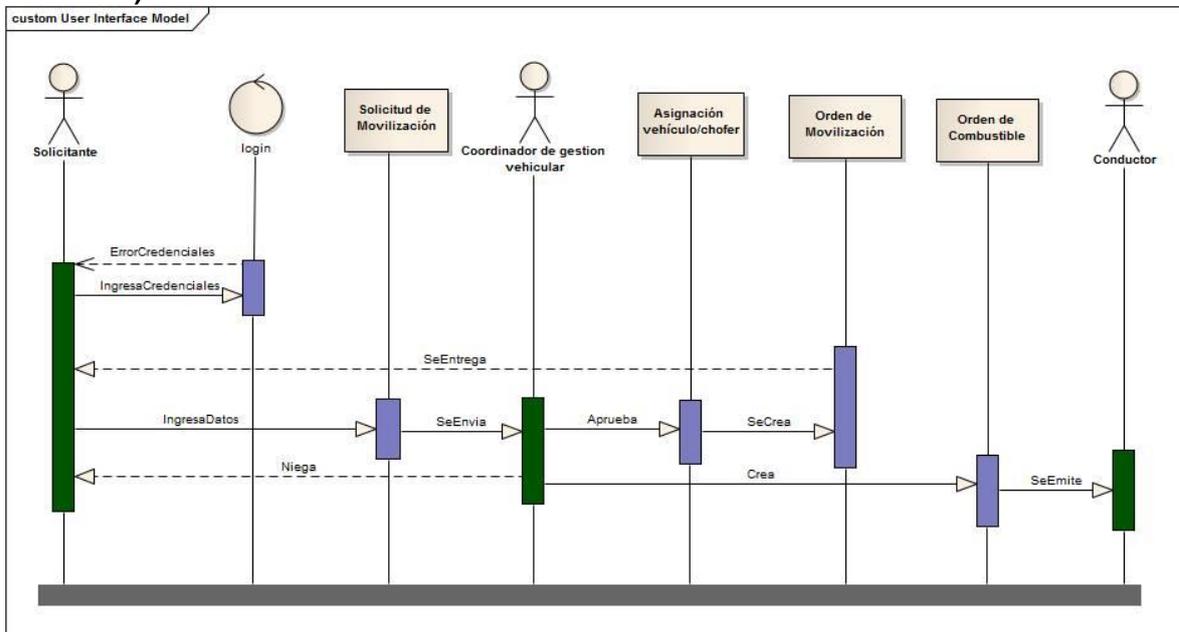


Figura 26. Diagrama de Secuencia del Módulo Movilización  
Elaboración: (Ordóñez & Guaman, 2015).

##### b) Gestión de Talleres

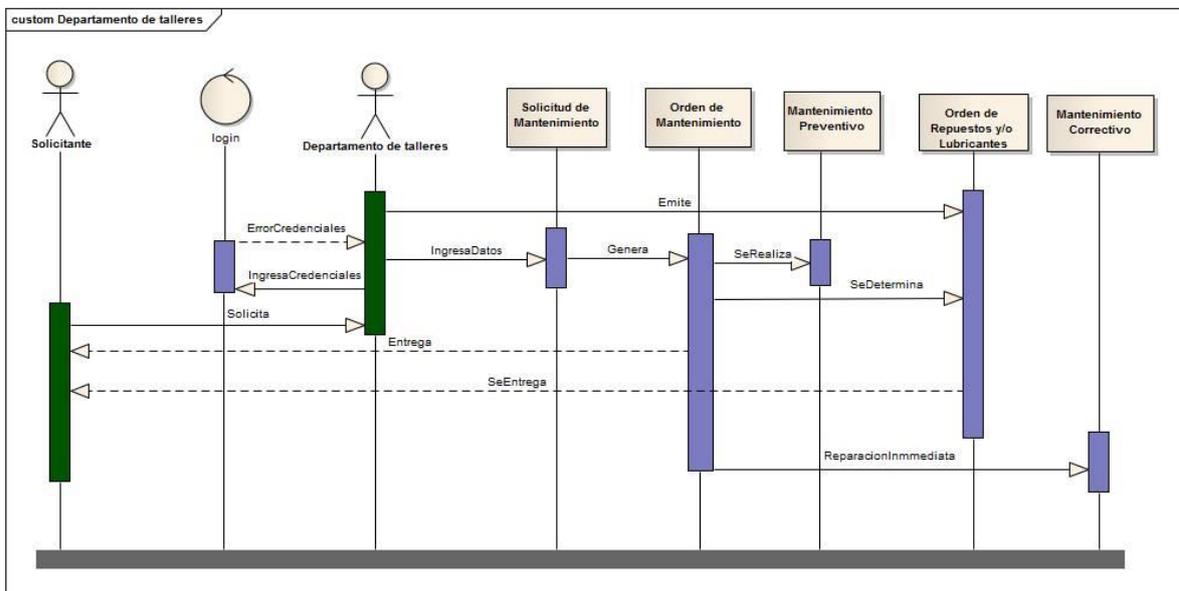


Figura 27 Diagrama de Secuencia del Módulo Talleres.  
Elaboración: (Ordóñez & Guaman, 2015)

### 3.5 Vista de Despliegue

#### 3.6.1 Vista de Despliegue

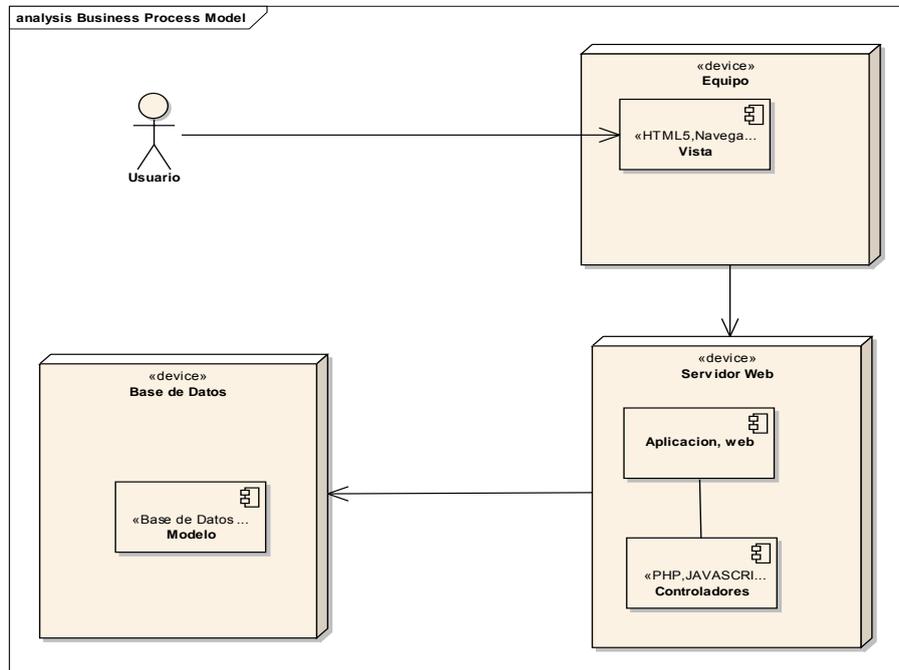


Figura 28. Diagrama de Despliegue  
Elaboración: (Ordóñez & Guaman, 2015)

## **ANEXO G: PLAN DE PRUEBAS**

### ***SISTEMA DE GESTIÓN VEHÍCULAR*** *Plan de Pruebas*

Versión [0.1]

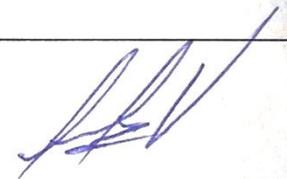
## Información del Documento

TÍTULO: Documento de Plan de Pruebas  
SUBTÍTULO: Sistema de Gestión Vehicular  
VERSIÓN: 1.2  
ARCHIVO: certificadox  
AUTOR: Andrea Guamán  
Yonder Ordóñez  
ESTADO: Final

## Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	2015-01-01	Andrea Guamán- Yonder Ordóñez	Creación del Documento.
1.1	2015-07-06	Andrea Guamán- Yonder Ordóñez	Corrección del documento de Plan de Pruebas
1.2	2015-09-10	Andrea Guamán- Yonder Ordóñez	Documento Final

## Firmas y Aprobaciones

REVISADO POR:	Ing. Pablo Vallejo Coordinador De Infraestructura Tecnología	
FECHA:	[2015-09-08]	

REVISADO POR:	Ing. Rubén Vicuña Coordinador de Gestión Vehicular	
FECHA:	[2015-09-08]	



## 1. Introducción

El plan de pruebas está basado en la metodología de Rational Unified Process (RUP), que tiene como propósito establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación de cada una de las pruebas que permiten garantizar el cumplimiento de los requerimientos planteados en el marco del desarrollo del sistema.

### 1.1 Proposito

El presente documento tiene como propósito establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación del plan de pruebas, en el que se incluye responsabilidades de cada una de las tareas, los recursos y los prerequisites que deben ser considerados en cada una de las pruebas.

### 1.2 Alcance

La metodología de pruebas y este documento de plan de pruebas permiten evaluar aspectos como: la lógica estructural, la seguridad, la interconexión, el soporte conceptual, las herramientas de apoyo y sobretodo la independencia de aspectos técnicos del desarrollo de la solución de software, tales como: la plataforma tecnológica o la arquitectura de la solución a probar, a continuación se describen las diferentes pruebas al ser aplicadas.

TIPO DE PRUEBA	DEFINICIONES	FASE DE RUP
<b>UNITARIAS</b>	<b>Unitarias:</b> Permite verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.	<b>ELABORACIÓN</b>
<b>SISTEMA CARGA RENDIMIENTO INTERFAZ DE USUARIOS SEGURIDAD</b>	<p><b>Sistema:</b> Estas pruebas buscan diferencias entre la solución desarrollada y los requerimientos, con el fin de identificar errores que se puedan generar entre la especificación funcional y el diseño del sistema.</p> <p><b>Carga:</b> Valida aquellos volúmenes de datos máximos especificados en los requerimientos no Funcionales</p> <p><b>Rendimiento:</b> Permite validar si la aplicación cumple los criterios de tiempos de respuesta establecidos.</p> <p><b>Interfaz de usuario:</b> Permite verificar que la navegación a través de los elementos que se están probando, reflejen las funciones del negocio y los requerimientos funcionales.</p> <p><b>Seguridad:</b> Verifica el cumplimiento de las políticas de seguridad acordadas para el sistema.</p>	<b>CONSTRUCCIÓN</b>
<b>FUNCIONAL USABILIDAD</b>	<p><b>Funcional:</b> La prueba funcional es un proceso para procurar encontrar discrepancias entre el programa y la especificación funcional.</p> <p><b>Usabilidad:</b> Esta prueba permite encontrar problemas de factores humanos, o usabilidad.</p>	
<b>ACEPTACIÓN</b>	Es la prueba final basada en las especificaciones del usuario o basada en el uso del programa por el usuario final luego de un periodo de tiempo.	
	En esta prueba se valida que el sistema mantenga su correcta funcionalidad debido a la incorporación de un ajuste, corrección o nuevo requerimiento. Es una prueba funcional y	<b>TRANSICIÓN</b>

**REGRESIÓN**

técnica que valida que el sistema siga funcionando perfectamente después de que las correcciones sean aplicadas.

### 1.3 Definiciones, acrónimos y abreviaturas

- El plan de prueba: describe todos los métodos que se utilizarán para verificar que el software satisface la especificación del producto y las necesidades del cliente.
- Casos de prueba: lista los ítems específicos que serán probados y describe los pasos detallados que serán seguidos para verificar el software.
- Reporte de pruebas: describen los problemas encontrados al ejecutar los casos de prueba.
- Herramientas de pruebas y automatización: documentación de las herramientas empleadas en el proceso de pruebas.
- Métricas, estadísticas y resúmenes: indican como ha sido el progreso del proceso de prueba.

### 1.4 Referencia

- Documento de Especificación de Requerimientos.
- Rational Unified Process (RUP).

## 2. Herramientas para pruebas.

Las herramientas que se usen dependen del tipo de prueba que se realiza, es decir que por cada tipo de prueba es posible que se utilice una herramienta diferente.

HERRAMIENTA	CARACTERÍSTICAS	TIPO DE PRUEBA
<b>SimpleTest</b> <b>CodeInteger</b>	Librería de CodeInteger para pruebas unitarias. Nos proporciona: <ul style="list-style-type: none"> <li>• Escribir tests de una manera sencilla.</li> <li>• Ejecutarlos con facilidad.</li> <li>• Analiza los resultados.</li> </ul>	Esta herramienta será utilizada para la ejecución de: <ul style="list-style-type: none"> <li>• Pruebas unitarias.</li> </ul>
<b>WEBPAGE TEST</b> <b>W3C validator</b> <b>W3C Link Cheker</b>	Ejecuta pruebas de velocidad al sitio por cada transacción.  Servicio de validación de hojas de estilo en cascada, aplicación diseñada para encontrar errores comunes o topográficos o usos incorrectos de CSS.	Esta herramienta será utilizada para la ejecución de: <ul style="list-style-type: none"> <li>• Pruebas Funcionales.</li> <li>• Pruebas de Sistema</li> <li>• Pruebas de Interfaz de Usuario.</li> </ul>

### 3. Configuración del ambiente de pruebas.

El ambiente y configuración de las pruebas será implementado tomando en cuenta la infraestructura tecnológica que posee el gobierno provincial de Loja.

Equipo	SO	Características	Software instalado
<b>Servidor</b>	Centos 6.6	32 bits(i386)	Apache 5.5 Base de datos postgres 8.20
<b>Computador</b>	Windows 7	Procesador: Intel(R) Core(TM) i5-337U CPU @1.80 Ghz  Memoria instalada: 2GB  Tipo de sistema: Sistema operativo de 32bits	Navegador Mozilla

### 4. Ejecución de las pruebas y resultado de las pruebas

El tipo de pruebas que se describieron en la lista anterior se contemplan cada una de las fases de la metodología de desarrollo de software, las que tienen como propósito

- Descubrir errores.
- Identificar y resolver falencias funcionales de la aplicación.
- Cumplir con los requerimientos del cliente.
- Evaluar la calidad del software

#### 4.1 Pruebas Unitarias

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.

Objetivo de la Prueba:	Validar las piezas individuales del software como una unidad independiente.
Estrategia	<ul style="list-style-type: none"> <li>• Se efectúan para los servicios del negocio y para la lógica del sistema en la capa Web que tengan complejidad alta.</li> <li>• Generar casos de pruebas necesarios que permitan identificar: <ul style="list-style-type: none"> <li>• Que al menos cada sentencia o instrucción del programa se ejecute al menos una vez correctamente.</li> <li>• Que cada condición tenga por lo menos una vez un resultado positivo y/o negativo.</li> <li>• Que cada bucle del sistema se pueda probar considerando: - ignorar el bucle, pasar una vez, pasar n veces.</li> </ul> </li> </ul>
Herramienta requeridas:	<ul style="list-style-type: none"> <li>• Simple test CodeInteger</li> </ul>
Observaciones	<p>La prueba se realizará por Módulo entendiéndose por tal:</p> <ul style="list-style-type: none"> <li>• Bloque básico de programa</li> <li>• Implementa función independiente y simple</li> <li>• Puede probarse por separado.</li> </ul>

#### 4.1.1 Prueba Unitaria para la Solicitud de Movilización

**Objetivo:** Con la siguiente imagen muestra el código que nos permite evaluar la funcionalidad del módulo de Solicitud de Movilización en cuanto a los tipos de datos que se envían y se reciben.

**Codificación realizada en la siguiente prueba:**

```
public function prueba_unit_cas01(){
    $this->load->library('unit_test');
    $actual= date("d/m/Y");
    $usuario=$this->session->userdata('username');
    $tests = Array(
        ((String) $this->input->post('nombres'))." ".((String) $this->input->post('apellidos')),
        $actual,
        $this->input->post('fecha_salida'),
        $this->input->post('fecha_llegada'),
        (String) strtoupper($this->input->post('motivo')),
        (String) strtoupper($this->input->post('ruta')),
        (integer) $this->input->post('ocupantes'),
        (String) strtoupper($this->input->post('integrantes')),
        (integer) $this->input->post('kilometraje'),
        (String) $this->input->post('departamento'),
        $this->input->post('hora_salida'),
        $this->input->post('hora_retorno'));

    foreach ($tests as $test) {
        $esperado=$test;
        $this->unit->run($test,$esperado,"". $test."");
    }
    echo $this->unit->report();
}
```

Figura 29. Codificación de la prueba Unitaria Solicitud de Movilización.  
Elaboración: (Ordóñez & Guaman, 2015)

**Resultados:** A continuación se presenta la imagen de resultados obtenidos tras ejecutar el código anterior el cual nos determina su cada tipo de dato enviado y recibido son los deseados o fallan.

A continuación detallamos los atributos visualizados con el Simple Test:

- Test Name: Es el nombre del resultado en este caso los datos que estamos enviando.
- Test Datatype: Tipo de dato enviado.
- Expected Datatype: Tipo de dato esperado.
- Result: Se aprueba o se desaprueba los datos enviados (Passed en tono de color verde y Failed en tono de color rojo).
- File Name: El archivo en el que se esta ejecutando el Simple Test.
- Line Number: La línea de comando donde se presenta el reporte, en otras palabras la pantalla de resultados del Simple Test.
- Notes: Notas de fallos.

Como primer paso debemos crear y guardar una solicitud de Movilización para obtener el Simple Test

## Solicitud de Movilización

[Nueva Solicitud](#)
[Solicitudes en Elaboración](#)
[Estado Solicitudes](#)

Solicitante:	1102768031	Departamento:	COORD DE INFRAESTRUCTURA TECNOLOGICA
Nombres:	JAIRO HERMES	Apellidos:	SILVA AGUILAR
Fecha Salida:	09/02/2015	Fecha Retorno:	09/03/2015
Hora Salida:	9:00	Hora Retorno:	12:00
Actividades a realizarse:	Trabajo de prueba		
Ruta:	SARAGURO GUALEL		
Número de ocupantes:	1		
Nombres de ocupantes:	YAGUANA ERAS RIVER GUSTAVO		

[Guardar](#)

Figura 30. Formulario Solicitud de Movilización  
Elaboración: (Ordóñez & Guaman, 2015)

A continuación se muestra el reporte de resultados del Simple Test tras ejecutarse el Código:

```

-----
Test Name                SARAGURO GUALEL
Test Datatype            String
Expected Datatype       String
Result                   Passed
File Name                C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number              131
Notes

Test Name                1
Test Datatype            Integer
Expected Datatype       Integer
Result                   Passed
File Name                C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number              131
Notes

Test Name                YAGUANA ERAS RIVER GUSTAVO
Test Datatype            String
Expected Datatype       String
Result                   Passed
File Name                C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number              131
Notes

Test Name                146
Test Datatype            Integer
Expected Datatype       Integer
Result                   Passed
File Name                C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number              131
Notes

Test Name                COORD DE INFRAESTRUCTURA TECNOLOGICA
Test Datatype            String
Expected Datatype       String
Result                   Passed
File Name                C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number              131
Notes
    
```

Figura 31. Resultado de la prueba Unitaria Solicitud de Movilización  
Elaboración: (Ordóñez & Guaman, 2015)

Como se observa en la imagen del Simple Test todos los datos enviados desde la solicitud de Movilización a guardarse en la Base de Datos, pasan o son los correctos.

A continuación se describe los resultados obtenidos del Simple Test.

- Test Name: 146 (El kilometraje obtenido al determinar la ruta ).
- Test Datatype: Integer (Tipo de dato del Kilometraje enviado a guardarse).
- Expected Datatype: Integer (Tipo de Dato esperado para ser guardado en la Base de Datos).
- Result: Passed (Tipos de datos correctos).
- File Name: C:\xampp\htdocs\movile\application\controllers\movilizacion.php (Ruta del código a la cual se aplica Simple Test).
- Line Number: 131 (Línea de Código donde se encuentra el reporte).
- Notes: Ninguna.

#### 4.1.2 Prueba Unitaria para la Orden de Movilización

**Objetivo:** Con la siguiente imagen se muestra el código que nos permite evaluar la funcionalidad para emitir una Orden de Movilización en cuanto a los tipos de datos que se envían y se reciben.

**Codificación realizada en la siguiente prueba:**

```
public function prueba_simple_test_caso2(){
    $this->load->library('unit_test');

    $actual= date("d/m/Y");
    $usuario=$this->session->userdata('username');
    $tests = Array(
        (integer) $this->input->post('chofer_dif'),
        (integer) $this->input->post('cod_vehiculo'),
        (String) $actual,
        (String) $usuario,
        (String) $this->input->post('obs'));

    foreach ($tests as $test) {
        $esperado=$test;
        $this->unit->run($test,$esperado,"".$test."");
    }
    echo $this->unit->report();
}
```

Figura 32.Codificación Prueba Unitaria Orden de Movilización  
Elaboración: (Ordóñez & Guaman, 2015)

**Resultados:** A continuación se presenta la imagen de resultados obtenidos tras ejecutar el código el cual nos determina si cada tipo de dato enviado y recibido son los deseados o fallan.

Se debe acceder al submenú de Aprobación para asignar el vehículo y emitir la orden de movilización.

Tabla de asignación de vehículos.

Asignar Vehiculo						
Solicitante	Fecha solicitud	Motivo	Destino	Fecha Salida	Fecha Llegada	Asignar Vehiculo
JORGE RAFAEL ALMEIDA ORELLANA	02/09/2015	TRABAJO	SARAGURO	09/24/2015	09/07/2015	Asignacion

Figura 33. Asignación vehículo  
Elaboración: (Ordóñez & Guaman, 2015)

A continuación se muestra el reporte de resultados del Simple Test tras ejecutarse el Código:

Test Name	457
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number	264
Notes	
Test Name	02/09/2015
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number	264
Notes	
Test Name	ADMINISTRADOR
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number	264
Notes	
Test Name	Se encuentra de vacaciones el conductor asignado, por tal razón se asigna a Angel Espinosa para esta ruta.
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\movilizacion.php
Line Number	264
Notes	

Figura 34. Resultado de la Prueba Unitaria Asignación de vehículo  
Elaboración: (Ordóñez & Guaman, 2015)

Como se observa en la imagen del Simple Test todos los datos enviados para la asignación de un vehículo, y donde se obtiene la orden de Movilización. A continuación se describe los resultados obtenidos del Simple Test de la orden de Movilización..

- Test Name: Se encuentra de vacaciones el conductor asignado, por tal razón se asigna a Angel Espinosa para esta ruta (La observación del cambio de chofer al vehículo escogido para la ruta ).
- Test Datatype: String (Tipo de dato de la observación enviado para la asignación).
- Expected Datatype: String (Tipo de Dato esperado para la asignación del vehículo).
- Result: Passed (Tipos de datos correctos).
- File Name: C:\xampp\htdocs\mobile\application\controllers\movilizacion.php (Ruta del código a la cual se aplica Simple Test).
- Line Number: 264 (Línea de Código donde se encuentra el reporte).
- Notes: Ninguna.

### 4.1.3 Prueba Unitaria para la Orden de Combustible

**Objetivo:** Con la siguiente imagen se muestra el código que nos permite evaluar la funcionalidad para emitir una Orden de Combustible en cuanto a los tipos de datos que se envían y se reciben.

**Codificación realizada en la siguiente prueba:**

```
public function prueba_simple_test_caso3(){
    $this->load->library('unit_test');
    $tests = Array(
        (String) $this->input->post('fecha'),
        (double) $this->input->post('galones'),
        (integer) $this->input->post('cod_gasolina'),
        (integer) $this->input->post('cod_gasolinera'),
        (integer) $this->input->post('kilometraje_recorrido'),
        (integer) $this->input->post('cod_movilizacion'),
        (integer) $this->input->post('cod_chofer'),
        (integer) $this->input->post('cod_vehiculo'),
        (integer) $this->input->post('km_salida'),
        (integer) $this->input->post('km_retorno'),
        (String) $this->input->post('hora_salida'),
        (String) $this->input->post('hora_retorno'),
        (String) $this->input->post('partida_combus');
    );

    foreach ($tests as $test) {
        $esperado=$test;
        $this->unit->run($test,$esperado,"" . $test . "");
    }
    echo $this->unit->report();
}
```

Figura 35. Codificación de la prueba Unitaria Orden de Combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

**Resultados:** A continuación se presenta la imagen de resultados obtenidos tras ejecutar el código el cual nos determina si cada tipo de dato enviado y recibido son los deseados o fallan.

Se debe acceder al submenú de Asignación de Combustibles para emitir la orden de Combustibles.

#### Asignación de Combustibles.

CONTROL COMBUSTIBLE			
Fecha:	<input type="text" value="18/09/2015"/>	Partida de Combustible	<input type="text" value="COMBUSTIBLES Y LUBRICANTES"/>
Vale por	<input type="text" value="5"/> <small>Galones de DIESEL</small>	Gasolinera:	<input type="text" value="ESTACION DE SERVICIOS VALDIVIES"/>
Con cargo a la cuenta de	<input type="text" value="BLANCO MARINO SILVA BRAVO"/>		
Ruta	<input type="text" value="CATAMAYO"/>		
Maquinaria/Vehículo:	<input type="text" value="TOYOTA,49"/>	Kilometraje Recorrido	<input type="text" value="75"/>
Km salida:	<input type="text" value="146"/>	Km retorno:	<input type="text" value="221"/>
Hora salida:	<input type="text" value="19:00:00"/>	Hora retorno:	<input type="text" value="16:00:00"/>
<input type="button" value="Guardar"/>			

Figura 36. Formulario de Asignación de combustibles.  
Elaboración: (Ordóñez & Guaman, 2015)

A continuación se muestra el reporte de resultados del Simple Test tras ejecutarse el Código:

Test Name	2015-09-03
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\movilizacion.php
Line Number	331
Notes	
Test Name	5
Test Datatype	Float
Expected Datatype	Float
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\movilizacion.php
Line Number	331
Notes	
Test Name	3
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\movilizacion.php
Line Number	331
Notes	
Test Name	1
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\movilizacion.php
Line Number	331
Notes	

Figura 37. Resultado de la prueba de unitaria Asignación de vehículo.  
Elaboración: (Ordóñez & Guaman, 2015)

Como se observa en la imagen del Simple Test todos los datos enviados para la asignación de combustible, y donde se obtiene la orden de Combustibles. A continuación se describe los resultados obtenidos del Simple Test de la orden de Combustible..

- Test Name: 5 (Son los galones de combustible asignados para el vehículo ).
- Test Datatype: Float (Tipo de dato de los galones de combustible).
- Expected Datatype: Float (Tipo de Dato esperado para los galones de combustible).
- Result: Passed (Tipos de datos correctos).
- File Name: C:\xampp\htdocs\movile\application\controllers\movilizacion.php (Ruta del código a la cual se aplica Simple Test).
- Line Number: 331 (Línea de Código donde se encuentra el reporte).
- Notes: Ninguna.

#### 4.1.4 Prueba Unitaria para el Despacho de Combustibles

**Objetivo:** Con la siguiente imagen se muestra el código que nos permite evaluar la funcionalidad para despachar combustibles en cuanto a los tipos de datos que se envían y se reciben.

**Codificación realizada en la siguiente prueba:**

```
public function prueba_simple_test_caso4(){
    $this->load->library('unit_test');
    $cantidad_despachada=$this->input->post('cantidad_despachada');
    $cod_orden = $this->input->post('cod_orden');
    $tests = Array(
        (integer) $cantidad_despachada,
        (integer) $cod_orden);

    foreach ($tests as $test) {
        $esperado=$test;
        $this->unit->run($test,$esperado,"".$test."");
    }
    echo $this->unit->report();
}
```

Figura 38. Codificación de la prueba unitaria despacho de Combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

**Resultados:** A continuación se presenta la imagen de resultados obtenidos tras ejecutar el código el cual nos determina si cada tipo de dato enviado y recibido son los deseados o fallan.

Se debe acceder al submenú de Despacho de Combustibles.

Despacho de Combustibles.



Figura 39. Formulario de despacho de combustibles.  
Elaboración: (Ordóñez & Guaman, 2015)

A continuación se muestra el reporte de resultados del Simple Test tras ejecutarse el Código:

Test Name	12
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\movilizacion.php
Line Number	396
Notes	
Test Name	178
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\movilizacion.php
Line Number	396
Notes	

Figura 40. Resultado de la prueba unitaria despacho de combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

Como se observa en la imagen del Simple Test todos los datos enviados para el despacho de combustibles, donde se le asigna gasoles de combustible reales. A continuación se describe los resultados obtenidos del Simple Test el despacho de Combustibles..

- Test Name: 178(Código del vehículo al que se le asigna los galones de combustible).

- Test Datatype: Integer(Tipo de dato del código del vehículo).
- Expected Datatype: Float (Tipo de Dato para el código del vehículo).
- Result: Passed (Tipos de datos correctos).
- File Name: C:\xampp\htdocs\movile\application\controllers\movilizacion.php (Ruta del código a la cual se aplica Simple Test).
- Line Number: 396 (Línea de Código donde se encuentra el reporte).
- Notes: Ninguna.

#### 4.1.5 Prueba Unitaria para la emisión de la Orden de Mantenimiento

**Objetivo:** Con la siguiente imagen se muestra el código que nos permite evaluar la funcionalidad de la emisión de la orden de mantenimiento en cuanto a los tipos de datos que se envían y se reciben.

**Codificación realizada en la siguiente prueba:**

```

public function prueba_simple_test_caso5(){
    $this->load->library('unit_test');
    $id_man_preven1=$this->input->post('id_man_preven');
    $stri="" $id_man_preven1 "";
    $datos=$this->talleres_model->getPlan_preven_nombre($stri);
    $codigol=1;
    $fecha=$this->input->post('fecha');
    $detalle=$this->input->post('detalle_desperfecto1');
    $motivo=$this->input->post('tipo');
    $cod_chofer=$this->input->post('cod_chofer');
    $cod_vehiculo=$this->input->post('cod_vehiculo');
    $kilometraje=$this->input->post('kilometraje');
    $respSolicitud=$this->talleres_model->agregar_mantenimiento($fecha,$detalle,$motivo,$cod_chofer,$cod_vehiculo,
        $detalle,$kilometraje,$codigol);
    if($respSolicitud){
        $tamano=count($this->input->post('item1'));
        $cantidad=$this->input->post('cantidad1');
        $descripcion=$this->input->post('item1');
        $cont=0;
        while($cont<$tamano){
            $test1 = Array((integer) $respSolicitud,(String) $descripcion[$cont],(integer) $cantidad[$cont]);
            foreach ($test1 as $test1) {
                $esperado1=$test1;
                $this->unit->run($test1,$esperado1,"". $test1."");
            } $cont++;
        }
    }
    $tests = Array((String) $fecha,(String) $detalle,(String) $motivo,(integer) $cod_chofer,(integer) $cod_vehiculo,
        (integer) $detalle,(integer) $kilometraje,(integer) $codigol);
    foreach ($tests as $test) {
        $esperado=$test;
        $this->unit->run($test,$esperado,"". $test."");
    }
    echo $this->unit->report();
}

```

Figura 41. Codificación de la prueba unitaria Orden de mantenimiento.  
Elaboración: (Ordóñez & Guaman, 2015)

**Resultados:** A continuación se presenta la imagen de resultados obtenidos tras ejecutar el código el cual nos determina si cada tipo de dato enviado y recibido son los deseados o fallan.

Se debe acceder al menú de Talleres y después al submenú de Solicitud de Mantenimiento para emitir la orden de Mantenimiento.

## Solicitud de Mantenimiento.

12/05/2015	LSA1019	Buscar ▾	
Marca	TOYOTA	Tipo	
Modelo	BPT HILUX 4X4 CD AA	Num. GPL	3
Motor	2TR6626464	Chasis	MR0FX22G991018246
Placa	LSA1019	Año	2009
Operador	DIEGO BENIGNO VASQUEZ ERRAEZ	Combustible	EXTRA
Tipo de Vehículo			
Kilometraje de la última mantenimiento	45000000		
Kilometraje de la última movilización	167028		
Kilometraje Actual:	200000	Tipo:	Reparacion ▾
<b>DETALLE DE DESPERFECTOS</b>			
Cambio de Bujías			

Figura 42. Formulario de orden de mantenimiento.

Elaboración: (Ordóñez & Guaman, 2015)

A continuación se muestra el reporte de resultados del Simple Test tras ejecutarse el Código:

Test Name	596
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\talleres.php
Line Number	293
Notes	
Test Name	Couplera = Boquilla de engrasadora
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\talleres.php
Line Number	293
Notes	
Test Name	1
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\talleres.php
Line Number	293
Notes	
Test Name	596
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\talleres.php
Line Number	293
Notes	
Test Name	Disco de embrague (HINO)
Test Datatype	String
Expected Datatype	String
Result	Passed

Figura 43. Resultado de la prueba unitaria Solicitud de Mantenimiento

Elaboración: (Ordóñez & Guaman, 2015)

Como se observa en la imagen del Simple Test todos los datos enviados para la solicitud de Mantenimiento. A continuación se describe los resultados obtenidos del Simple Test para la Solicitud de Mantenimiento.

- Test Name: Couplera-Boquilla de engranadora (Descripción del Repuesto que se solicita para el mantenimiento, este repuesto se lo obtiene al consultar la base de datos de Activos ).
- Test Datatype: String(Tipo de dato del repuesto).

- Expected Datatype: Float (Tipo de Dato esperado del repuesto).
- Result: Passed (Tipos de datos correctos).
- File Name: C:\xampp\htdocs\movile\application\controllers\talleres.php (Ruta del código a la cual se aplica Simple Test).
- Line Number: 293 (Línea de Código donde se encuentra el reporte).
- Notes: Ninguna.

#### 4.1.6 Prueba Unitaria para la consulta de Reportes

**Objetivo:** Con la siguiente imagen se muestra el código que nos permite evaluar la funcionalidad de la consulta de reportes cuanto a los tipos de datos que se reciben.

**Codificación realizada en la siguiente prueba:**

```

public function prueba_simple_test_caso6(){
    $this->load->library('unit_test');
    $conductor=$this->administracion_model->getConductores(0,0);
    foreach ($conductor as $rep_conductores) {
        $resp1= (String) $rep_conductores->nombres;
        $resp2= (integer) $rep_conductores->cedula;
        $resp3= (String) $rep_conductores->denominacion_cargo;
        $resp4= (String) $rep_conductores->novedades;
        $tests=Array($resp1,$resp2,$resp3);
        foreach ($tests as $test) {
            $esperado=$test;
            $this->unit->run($test,$esperado,"".$test."");
        }

        $tests1=Array("No registra Licencia");
        $esperado1="Tipo de Licencia A";
        foreach ($tests1 as $test1) {
            $esperad1=$test1;
            $this->unit->run($test1,$esperado1,"".$test1."","".$esperado1."");
        }
        $tests2=Array($resp4);
        $esperado2=$resp4;
        foreach ($tests2 as $test2) {
            if($test2==""){
                $tests2=Array(null);
                $esperado3=$resp4;
                $this->unit->run($test2,$esperado3,"** Cadena en blanco **");
            }else{
                $this->unit->run($test2,$esperado2,"".$resp4."");
            }
        }
    }
    echo $this->unit->report();
}

```

Figura 44. Codificación de la prueba unitaria Reportes.  
Elaboración: (Ordóñez & Guaman, 2015)

**Resultados:** A continuación se presenta la imagen de resultados obtenidos tras ejecutar el código el cual nos determina si cada tipo de dato recibido son los deseados o fallan.

Se debe acceder al menú de Reportes para la consulta de cualquier reporte.

## Reportes (Conductores).

REPORTE CONDUCTORES				
LICENCIA	NOMBRE	# CEDULA	RELACION LABORAL	NOVEDADES
NO REGISTRA LICENCIA	JOSE FRANCISCO CHALAN LOZANO	1103143549	CONDUCTOR	
NO REGISTRA LICENCIA	RODRIGO VINCIO PINZON	1101914461	CONDUCTOR	
NO REGISTRA LICENCIA	LUIS ALBERTO ROJAS FLORES	1103203277	CONDUCTOR	
NO REGISTRA LICENCIA	ANGEL MARCO QUEZADA PONCE	1102967526	CONDUCTOR	
NO REGISTRA LICENCIA	RODRIGO MARCO LUZURIAGA HERRERA	1100334679	CONDUCTOR	
NO REGISTRA LICENCIA	ALVARO DEMETRIO GRANDA IÑIGUEZ	1100440138	CONDUCTOR	
NO REGISTRA LICENCIA	NEY EFREN RIVAS POGO	1102820048	CONDUCTOR	
NO REGISTRA LICENCIA	DIEGO BENIGNO VASQUEZ ERRAEZ	1103156533	CONDUCTOR	
NO REGISTRA LICENCIA	FELICITO GUZMAN CASTILLO HERRERA	1102124052	CONDUCTOR	
NO REGISTRA LICENCIA	RENE GUSTAVO SILVA VEINTIMILLA	1101879359	CONDUCTOR	

Figura 45. Vista reportes conductores.  
Elaboración: (Ordóñez & Guaman, 2015)

A continuación se muestra el reporte de resultados del Simple Test tras ejecutarse el Código:

Test Name	CONDUCTOR
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\administracion.php
Line Number	916
Notes	
Test Name	No registra Licencia
Test Datatype	String
Expected Datatype	String
Result	Failed
File Name	C:\xampp\htdocs\mobile\application\controllers\administracion.php
Line Number	923
Notes	Tipo de Licencia A
Test Name	** Cadena en blanco **
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\administracion.php
Line Number	935
Notes	
Test Name	RENE GUSTAVO SILVA VEINTIMILLA
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\mobile\application\controllers\administracion.php

Figura 46. Resultado de la prueba unitaria de los Reportes.  
Elaboración: (Ordóñez & Guaman, 2015)

Como se observa en la imagen del Simple Test todos los datos consultados para el reporte de conductores. A continuación se describe los resultados obtenidos del Simple Test para la consulta de reportes.

- Test Name: No registra Licencia (Esta descripción se devuelve cuando no registra el conductor aún una licencia ).
- Test Datatype: String(Tipo de dato de la licencia).
- Expected Datatype: Float (Tipo de Dato esperado de la licencia)
- Result: Failed (Tipos de datos incorrecto).

- File Name: C:\xampp\htdocs\movile\application\controllers\administracion.php (Ruta del código a la cual se aplica Simple Test).
- Line Number: 293 (Línea de Código donde se encuentra el reporte).
- Notes: Tipo de Licencia A (El tipo de descripción que se esperaba para el conductor ya que registra una licencia).

Como se puede observar con el dato de la licencia nos presenta un error que se esperaba obtener una descripción distinta a la que se obtuvo, no es un error de tipo de datos, sino, de la información retornada. Por lo tanto el reporte debe ser corregido para devolver la correcta descripción, pero los demás datos si son correctos por lo que el Simple Test los da como pasados.

#### 4.1.7 Prueba Unitaria para la Administración

Código del Simple Test: Con la siguiente imagen se muestra el código que nos permite evaluar la funcionalidad de la Administración en cuanto a los tipos de datos que se envían y reciben.

**Codificación realizada en la siguiente prueba:**

```

public function prueba_simple_test_caso7(){
    $this->load->library('unit_test');
    $tests = Array( (String) $this->input->post('nombre'), (String) $this->input->post('usuario'),
        (String) md5($this->input->post('password')));

    foreach ($tests as $test) {
        $esperado=$test;
        $this->unit->run($test,$esperado,"".$test."");
    }
    $tamano=count($this->input->post('menu'));
    $descripcion=$this->input->post('menu');
    $cont=0;
    while($cont<$tamano){
        $tests1 = Array((integer) $descripcion[$cont]);
        foreach ($tests1 as $test1) {
            $esperado1=$test1;
            $this->unit->run($test1,$esperado1,"".$test1."");
        }
        $cont++;
    }

    echo $this->unit->report();
}

```

Figura 47. Codificación de la prueba unitaria del módulo Administración.  
Elaboración: (Ordóñez & Guaman, 2015)

Resultados: A continuación se presenta la imagen de resultados obtenidos tras ejecutar el código el cual nos determina si cada tipo de dato recibido son los deseados o fallan.

Se debe acceder al menú de Administración para la realizar cualquier ingreso de datos, en este caso se hizo las pruebas a registrar usuario.

## Administración (Registrar Usuario).

AGREGAR USUARIO

<b>Nombre</b> Alberto Ulloa Contrera	<b>Usuario</b> ulloaC	<b>Password</b> .....
---	--------------------------	--------------------------

**ADMINISTRACION**

USUARIOS

TIPO LICENCIA

GASOLINERA

REGISTRAR CONDUCTOR

**GESTION MOVILIZACION**

SOLICITUD DE MOVILIZACION

ORDEN DE COMBUSTIBLE

CARGAR

**TALLERES**

SOLICITUD DE MANTENIMIENTO

PLAN DE MANTENIMIENTO PREVENTIVO

**REPORTES**

LISTADO VEHICULOS

CONTROL DE MANTENIMIENTO

MATRICULA Y SOAT

NOVEDADES

ASIGNACION VEHICULO/CHOFER

COMBUSTIBLE

REGISTRAR VEHICULO

APROBACION

DESPACHO COMBUSTIBLE

HISTORIAL DE MANTENIMIENTO

LISTADO CONDUCTORES

PROVISION DE COMBUSTIBLES

ORDENES DE MOVILIZACION

Figura 48. Formulario de Registro de Usuario.

Elaboración: (Ordóñez & Guaman, 2015)

A continuación se muestra el reporte de resultados del Simple Test tras ejecutarse el Código:

Test Name	Alberto Ulloa Contrera
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\administracion.php
Line Number	57
Notes	
Test Name	ulloaC
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\administracion.php
Line Number	57
Notes	
Test Name	835863e6abdb306d873e816bab767b7d
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\administracion.php
Line Number	57
Notes	
Test Name	9
Test Datatype	Integer
Expected Datatype	Integer
Result	Passed
File Name	C:\xampp\htdocs\movile\application\controllers\administracion.php
Line Number	67
Notes	

Figura 49. Resultado de la prueba unitaria registro de usuario.

Elaboración: (Ordóñez & Guaman, 2015)

Como se observa en la imagen del Simple Test todos los datos enviados para el registro del usuario. A continuación se describe los resultados obtenidos del Simple Test para la administración:

- Test Name: 835863e6abdb306d873e816bab767b7d (Clave encriptada del usuario).
- Test Datatype: String(Tipo de dato del password).
- Expected Datatype: String (Tipo de Dato esperado del password)
- Result: Passed (Tipos de datos correcto).
- File Name: C:\xampp\htdocs\movile\application\controllers\administracion.php (Ruta del código a la cual se aplica Simple Test).

- Line Number: 67 (Línea de Código donde se encuentra el reporte).
- Notes: Ninguna.

## 4.2 Pruebas del Sistema

Las pruebas de sistema buscan diferencias entre la solución desarrollada y los requerimientos, enfocándose en la identificación de los errores que se puedan generar entre la especificación funcional y el diseño del sistema, así como, el negocio objeto de la aplicación.

Objetivo de la Prueba:	<ul style="list-style-type: none"> <li>• Validar aquellos volúmenes de datos máximos (por lo general las transacciones o informes) que pueden ser completados dentro de un período específico en el tiempo, y con un nivel de concurrencia dado (carga, concurrencia y desempeño).</li> <li>• Validar los requerimientos no funcionales de cada proyecto.</li> </ul>
Estrategia:	<ul style="list-style-type: none"> <li>• Realizar Set de Pruebas a partir de los Requerimientos no funcionales.</li> <li>• Realizar pruebas de rendimiento básico. Consiste en probar la aplicación simulando la carga esperada en el entorno de producción.</li> <li>• Realizar las pruebas de concurrencia: verificar el comportamiento de la aplicación en condiciones de sobrecarga de usuarios, que supone permitirá identificar potenciales problemas de rendimiento o cuellos de botella, antes de su pase a producción.</li> <li>• Realizar pruebas de requerimientos no funcionales: Consiste en probar la aplicación con cada uno de los requerimientos no funcionales establecidos en el proyecto.</li> <li>• Identificar posibles cuellos de botella o problemas de rendimiento.</li> <li>• Realizar pruebas de carga: Altos volúmenes de información.</li> </ul>
Herramienta requeridas:	<ul style="list-style-type: none"> <li>• PHPUNIT</li> <li>• HTTPUNIT</li> </ul>
Observaciones:	<p>La elección de la herramienta de prueba será a discreción del grupo de pruebas y su elección dependerá de la prueba que se va a realizar.</p> <p>El conjunto de pruebas que se abarcan son las siguiente:</p> <ul style="list-style-type: none"> <li>• Prueba de Estrés</li> <li>• Prueba de Interfaz de usuario</li> <li>• Prueba de rendimiento</li> <li>• Prueba de seguridad</li> </ul>

### 4.2.1 Prueba de Interfaz de Usuario

El objetivo de esta prueba es validar con los usuarios la iteración y aspectos estéticos de la interfaz de usuario.

Para más detalle de cómo se encuentra el diseño de las interfaces dentro de este proyecto se puede anexar en el anexo H donde se detalla el manual de usuario.

### 4.2.2 Prueba de rendimiento

El objetivo de las pruebas de rendimiento en un requerimiento no funcional, para determinar la respuesta del sistema, es decir la velocidad, estabilidad, confiabilidad y escalabilidad.

Nº	Caso de Prueba	Tiempo
1	Cargar la pagina	5.segundos
2	Autenticación	4 segundos

3	Emitir orden de movilización	2 segundos
4	Emitir orden de combustible	2 segundos
5	Despacho de combustible	1 segundos
6	Emitir orden de mantenimiento	2 segundos
7	Consultar Reportes	4 segundos

El tiempo promedio de aplicación móvil para responder a las solicitudes es de 4.8 segundos tomando en cuenta las siguientes configuraciones:

- Una intensidad de señal wifi buena con una velocidad de 130,4 mbps.
- En una computadora con Windows 10

### 4.3 Pruebas de Seguridad

El objetivo de las pruebas de seguridad es constatar y validar las políticas de seguridad propuestas e integradas durante el transcurso del sistema, que a continuación se describen:

#### Asistente de Seguridad de CODEIGNITER:

El asistente de seguridad integrado en los proyectos de CODEIGNITER permite implementar funciones relacionadas a la seguridad en el código PHP.

Para cargar el Asistente de Seguridad se incluye la siguiente línea de código:

```
$this->load->helper('security');
```

Las siguientes funciones se encuentran disponibles para el manejo de seguridades en el código PHP:

- `xss_clean()`: Permite dar filtrado a los ataques XSS (Cross Site Script).  
`xss_clean($ str [, $ is_image = FALSO]);`
- `dohash()`: Permite encriptar contraseñas con MD5.  
`$cadena=dohash($cadena,'md5');`
- `strip_image_tags()`: Elimina las etiquetas de las imágenes de una cadena obteniendo como resultado la URL de la imagen como texto plano.  
`$cadena=strip_image_tags($cadena);`
- `encode_php_tags()`: Convierte etiquetas PHP a entidades, si usa la función `xss_clean()` se ejecutará automáticamente la función.  
`$cadena = encode_php_tags($cadena);`

#### Procedimientos Almacenados POSTGRESQL:

Una de las utilidades más eficaces que los procedimientos almacenados brindan es en relación a la seguridad e integridad de las bases de datos, se implementa los procedimientos almacenados para todas las operaciones comunes del sistema ya que proporciona un entorno seguro y consistente, y pueden asegurar que cada operación se ejecuta apropiadamente. Al integrar en el sistema los procedimientos almacenados nos brinda seguridad en cuanto a las

aplicaciones y los usuarios que no obtendrían ningún acceso directo a las tablas de la base de datos, sólo pueden ejecutar algunos los procedimientos almacenados autorizados para su rol dentro del sistema.

#### 4.4 Pruebas Funcionales

La prueba funcional es un proceso para procurar encontrar discrepancias entre el software desarrollado y la especificación funcional. La prueba funcional normalmente es una actividad de caja negra. Esta prueba permite validar:

- Los procesos y reglas de negocio establecidas,
- Que se cumplan los requerimientos funcionales establecidos

En esta prueba se validan los Casos de Uso que fueron aprobados por el cliente, y a partir de ellos se diseñan y ejecutan los set de pruebas correspondientes. Se deben elaborar los casos de pruebas necesarios que permitan asegurar el funcionamiento de todos los flujos normales y alternos de dichos casos de uso.

Objetivo de la Prueba:	Se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.
Estrategia :	<ul style="list-style-type: none"> <li>• Validación y ejecución de Set de Pruebas y escenarios definidos, teniendo en cuenta flujo normal y flujos alternativos, usando datos validos e inválidos para verificar lo siguiente: <ul style="list-style-type: none"> <li>• Los resultados esperados ocurren cuando se usan datos válidos.</li> <li>• Se despliegan mensajes de error cuando se usan datos inválidos.</li> <li>• Cada regla de negocio es propiamente aplicada.</li> <li>• Realizar set de pruebas de los requerimientos mínimos para el adecuado funcionamiento de la aplicación</li> </ul> </li> </ul>
Herramientas Requeridas:	<ul style="list-style-type: none"> <li>• Formato de casos de prueba funcionales</li> </ul>
Observaciones:	Se utiliza la plantilla de Casos de Pruebas.

A continuación se detallan las pruebas de funcionalidades por cada caso de uso del sistema con la herramienta de **CASOS DE PRUEBA**:

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
<b>CASO DE PRUEBA No.</b>	SP-01-01	VERSIÓN DE EJECUCIÓN	01
		FECHA EJECUCIÓN	24/08/2015
<b>CASO DE USO:</b>	Solicitud de Movilización	MODULO DEL SISTEMA	Gestión Movilización
<b>Descripción del caso de prueba:</b>	Se desea probar que la solicitud de movilización sea emitida al usuario con integridad y veracidad de los datos, cuando este requiera de un vehículo para movilizarse a cualquier parte de la provincia, además de corroborar la autorización del jefe de departamento.		
1. CASO DE PRUEBA			

a. Precondiciones

- Requerimiento de Movilización de un vehículo.
- Datos del empleado solicitante.
- Integración con las bases de datos de empleados.
- Ejecución en tiempo real del sistema.

b. Pasos de la prueba

- Abrir en el navegador la dirección 190.57.168.179/movile/
- Ingresar usuario y contraseña.
- Clic en el menú principal en Gestión Movilización y luego en Solicitud de Movilización.
- Tener un listado de empleados con su respectiva cedula.
- Hacer clic en Nueva Solicitud.

DATOS DE ENTRADA			TIPO ESCENARIO	RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR				SI	NO	
<b>Solicitante</b>	Número de cedula del empleado	de	Correcto	Devuelve y rellena con los datos del empleado en las casillas de: Departamento, Nombres y Apellidos pertenecientes al número de cedula ingresado.	✓		Automáticamente y de forma rápida devuelve y rellena los datos, bloqueando las casillas para evitar la distorsión de los datos.
			Incorrecto	Se presenta el mensaje "Su cedula es incorrecta"	✓		Inmediatamente devuelve el mensaje bloqueando las casillas para evitar rellenarlos con datos inexistentes.
<b>Fecha Salida y Fecha Retorno.</b>	Fechas en formato calendario.	en	Correcto	Al realizar clic o enter en las casillas automáticamente se despliega el calendario actual de fechas.	✓		Rápidamente se despliega el calendario, en cualquier navegador con el mismo formato y la fecha actual, no permite la entrada de caracteres diferentes a los de una fecha.
<b>Hora salida y Hora retorno.</b>	Horas laborales.		Correcto	Permite las casillas ingresar las horas de salida y retorno en formato del uso horario.	✓		El sistema controla que se ingrese únicamente caracteres para horas y que la hora de retorno sea mayor que la de salida.
<b>Actividades a realizarse</b>	Especificación o detalle de la movilización		Correcto	Permite la entrada de caracteres.	✓		Casilla sin controles adicionales para rellenarla desde teclado.

<b>Ruta</b>	Agregar la ruta de movilización	Correcto	Visualmente permite escoger la ruta como la provincia, cantón y parroquia en el mapa. ✓	Se visualiza el mapa tan solo cuando existe conexión a internet, una vez que se grafica la ruta adicionalmente muestra los tramos a recorrer, traza la ruta en el mapa y calcula el kilometraje de viaje de ida y vuelta del vehículo.
		Incorrecto	No se presenta el mapa. ✓	No presenta el mapa cuando no hay conexión a internet y no realiza los demás cálculos.
<b>Número de ocupantes y nombres de los ocupantes.</b>	Permite el número de ocupantes y sus respectivos nombres.	Correcto	Permite tan solo los enteros para los números de ocupantes y los caracteres para los nombres. ✓	Las casillas no tienen controles adicionales, tan solo el de número de ocupantes que controla el ingreso solo de enteros.
<b>Guardar</b>	Botón de guardado.	Correcto	Guarda los datos ✓	Se almacenan los datos presentando un mensaje de que se almacenaron con éxito.
		Incorrecto	No guarda los datos. ✓	Se presentan mensajes en las casillas donde no se haya rellenado o presentado los datos necesarios, ya que todos los campos son requeridos.

## 2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto
Un defecto encontrado es al momento de la asignación de la ruta, ya que no se muestra mensajes en caso no estar conectados a la internet, para que el usuario sepa que no podrá realizar dicha solicitud.	<input checked="" type="checkbox"/> Paso <input type="checkbox"/> Falló

### INFORMACIÓN GLOBAL DEL CASO DE PRUEBA

<b>CASO DE PRUEBA No.</b>	SP-02-02	<b>VERSIÓN DE EJECUCIÓN</b>	01
		<b>FECHA EJECUCIÓN</b>	24/08/2015

<b>CASO DE USO:</b>	Emitir Orden de Movilización	MODULO DEL SISTEMA	Gestión Movilización
---------------------	------------------------------	--------------------	----------------------

**Descripción del caso de prueba:** Se desea probar la emisión de la orden de movilización. Una vez enviado la solicitud de movilización se procede a ser aceptada por el departamento de movilización, donde se asignan vehículo y chofer.

### 3. CASO DE PRUEBA

#### c. Precondiciones

- Guardar una solicitud de movilización.
- Tener integrada con la base de datos activos.
- Hacer clic en el Gestión Movilización y luego en Aprobación.

#### d. Pasos de la prueba

- Guardar una solicitud de movilización
- Asignar un vehículo para dicha movilización.
- Clic en el menú principal en Gestión Movilización y luego en Aprobación.
- Conexión con la base de datos de Fultime.
- Clic en el link de Asignación en la tabla de Asignar Vehículo.

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE	RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI NO	
<b>Link Asignación</b>	Clic para asignar vehículo y aprobar.	Correcto	Carga una ventana donde se muestra los datos de los solicitantes y el destino.	✓	Automáticamente se cargan los datos del solicitante y la ruta.
<b>Seleccione Vehículo</b>	Modelo y Placa de vehículo.	Correcto	Al hacer clic se despliega la lista de vehículos disponibles.	✓	Se despliega la lista de vehículos disponibles para la ruta rápidamente con los datos almacenados de los vehículos.
<b>Link Asignar un chofer diferente</b>	Nombre del chofer.	Correcto	Se despliega la lista de choferes,	✓	En caso de que el chofer no esté disponible, se puede escoger otro chofer para la ruta.
<b>Observaciones</b>	Detalle del cambio de chofer.	Correcto	Permite la entrada de caracteres.	✓	Casilla sin controles adicionales para rellenarla desde teclado.
<b>Guardar</b>	Botón de guardado.	Correcto	Guarda los datos	✓	Se almacenan los datos presentando un mensaje de que se almacenaron con éxito.
		Incorrecto	No guarda los datos.	✓	Se presentan mensajes en las casillas donde no se haya rellenado o presentado los datos

necesarios, ya que todos los campos son requeridos.

#### 4. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto
Ninguno	<input checked="" type="checkbox"/> Paso <input type="checkbox"/> Falló

#### INFORMACIÓN GLOBAL DEL CASO DE PRUEBA

<b>CASO DE PRUEBA No.</b>	SP-03-03	<b>VERSIÓN DE EJECUCIÓN</b>	01
<b>CASO DE USO:</b>	Emitir Orden de Combustible	<b>FECHA EJECUCIÓN</b>	25/08/2015
		<b>MODULO DEL SISTEMA</b>	Gestión Movilización

**Descripción del caso de prueba:** Se desea probar la orden de combustible que será entregada al conductor en esta se especifica tipo de combustible y cantidad asignada. Estos deben quedar guardado, para luego emitir los reportes.

#### 5. CASO DE PRUEBA

##### e. Precondiciones

- **Aprobar la solicitud de movilización.**
- **Asignar vehículo.**
- **Hacer clic en el Gestión Movilización y luego en Despacho de Combustibles o después de guardar la asignación clic en Asignar Combustible.**

##### f. Pasos de la prueba

- **Aprobar la solicitud de movilización.**
- **Asignar vehículo.**
- **Clic en Asignar Combustible.**
- **O clic en Gestión de Movilización**
- **Clic en Despacho de Combustibles.**

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE	RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI NO	
<b>Fecha</b>	Fecha de creación de la orden de combustibles.	Correcto	Se despliega el calendario con las fechas actuales.	✓	Despliega el calendario con formatos de fechas actuales.
<b>Partida de Combustibles</b>	Tipo de proyecto al que se le asigna la orden.	Correcto	Al hacer clic se despliega la lista partidas actuales.	✓	Datos almacenados en la base de datos de las partidas o proyectos.
<b>Gasolinera</b>	Estación de servicio de combustibles.	Correcto	Se despliega los nombres de las gasolineras.	✓	Datos almacenados en la base de datos de las gasolineras.
<b>Con cargo a la</b>	Nombre del chofer.	Correcto	Nombre del chofer que se	✓	Esta casilla se rellena con los

<b>cuenta de:</b>			asignó a dicha movilización, la casilla esta bloqueada		datos del chofer que fue asignado para determinada solicitud de movilización.
<b>Ruta</b>	Detalle de la ruta	Correcto	Se presenta la ruta seleccionada en la solicitud de movilización.	✓	Esta casilla está bloqueada para evitar distorsión de la ruta cuando inicialmente se la determino.
<b>Maquinaria/Vehículo</b>	Modelo y placa del vehículo.	Correcto	Se presenta el modelo y la placa del vehículo asignado para la movilización	✓	Esta casilla está bloqueada, ya que anteriormente se asignó el vehículo.
<b>Kilometraje recorrido</b>	Cantidad de kilometraje recorrido.	Correcto	Se presenta el kilometraje de la ruta determinada.	✓	Esta casilla presenta el kilometraje de la ruta determinada, y no se encuentra bloqueada ya que puede variar el kilometraje recorrido en vista de percances.
<b>Kilometraje salida</b>	Cantidad de kilómetros de salida.	Correcto	Se ingresa el kilometraje con el que parte el vehículo.	✓	Casilla ingresada desde el teclado, controlando que se ingresen datos numéricos.
<b>Kilometraje retorno</b>	Cantidad de kilometraje de retorno.	Correcto	Se calcula el kilometraje con el que debe regresar el vehículo.	✓	La casilla bloqueada al momento de hacer clic sobre ella se calcula el kilometraje de retorno.
		Incorrecto	No se calcula el kilometraje de retorno.	✓	La casilla permanece sin datos, sino se ingresa el kilometraje de salida o sino se presenta la cantidad de kilómetros de la ruta determinada.

<b>Hora salida y hora retorno</b>	Horas de la salida y retorno.	Correcto	Se ingresan las horas de salida y retorno previstas para el vehículo.	✓	Las casillas son rellenas por teclado, controlando que solo se ingresen caracteres apropiados y en formatos de horas.
-----------------------------------	-------------------------------	----------	---	---	---

<b>Guardar</b>	Botón de guardado.	Correcto	Guarda los datos	✓	Se almacenan los datos presentando un mensaje de que se almacenaron con éxito.
----------------	--------------------	----------	------------------	---	--

Incorrecto	No guarda los datos.	✓	Se presentan mensajes en las casillas donde no se haya rellenado o presentado los datos necesarios, ya que todos los campos son requeridos.
------------	----------------------	---	---

**6. RESULTADOS DE LA PRUEBA**

Defectos y desviaciones	<b>Veredicto</b>
<b>Se debería presentar un mensaje cuando no se ingresa el kilometraje de salida.</b>	<input checked="" type="checkbox"/> <b>Paso</b> <input type="checkbox"/> <b>Falló</b>

**INFORMACIÓN GLOBAL DEL CASO DE PRUEBA**

<b>CASO DE PRUEBA No.</b>	SP-03-03	<b>VERSIÓN DE EJECUCIÓN</b>	01
		<b>FECHA EJECUCIÓN</b>	25/08/2015
<b>CASO DE USO:</b>	Emitir Orden de Combustible	<b>MODULO DEL SISTEMA</b>	Gestión Movilización

**Descripción del caso de prueba:** Se desea probar la orden de combustible que será entregada al conductor en esta se especifica tipo de combustible y cantidad asignada. Estos deben quedar guardado, para luego emitir los reportes.

**7. CASO DE PRUEBA**

**g. Precondiciones**

- Aprobar la solicitud de movilización.
- Asignar vehículo.
- Hacer clic en el Gestión Movilización y luego en Despacho de Combustibles o después de guardar la asignación clic en Asignar Combustible.

**h. Pasos de la prueba**

- Aprobar la solicitud de movilización.
- Asignar vehículo.
- Clic en Asignar Combustible.
- O clic en Gestión de Movilización
- Clic en Despacho de Combustibles.

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	

<b>Fecha</b>	Fecha de creación de la orden de combustibles.	Correcto	Se despliega el calendario con las fechas actuales.	✓	Despliega el calendario con formatos de fechas actuales.
<b>Partida de Combustibles</b>	Tipo de proyecto al que se le asigna la orden.	Correcto	Al hacer clic se despliega la lista partidas actuales.	✓	Datos almacenados en la base de datos de las partidas o proyectos.
<b>Gasolinera</b>	Estación de servicio de combustibles.	Correcto	Se despliega los nombres de las gasolineras.	✓	Datos almacenados en la base de datos de las gasolineras.
<b>Con cargo a la cuenta de:</b>	Nombre del chofer.	Correcto	Nombre del chofer que se asignó a dicha movilización, la casilla está bloqueada	✓	Esta casilla se rellena con los datos del chofer que fue asignado para determinada solicitud de movilización.
<b>Ruta</b>	Detalle de la ruta	Correcto	Se presenta la ruta seleccionada en la solicitud de movilización.	✓	Esta casilla está bloqueada para evitar distorsión de la ruta cuando inicialmente se la determino.
<b>Maquinaria/Vehículo</b>	Modelo y placa del vehículo.	Correcto	Se presenta el modelo y la placa del vehículo asignado para la movilización	✓	Esta casilla está bloqueada, ya que anteriormente se asignó el vehículo.
<b>Kilometraje recorrido</b>	Cantidad de kilometraje recorrido.	Correcto	Se presenta el kilometraje de la ruta determinada.	✓	Esta casilla presenta el kilometraje de la ruta determinada, y no se encuentra bloqueada ya que puede variar el kilometraje recorrido en vista de percances.
<b>Kilometraje salida</b>	Cantidad de kilómetros de salida.	Correcto	Se ingresa el kilometraje con el que parte el vehículo.	✓	Casilla ingresada desde el teclado, controlando que se ingresen datos

				numéricos.
<b>Kilometraje retorno</b>	Cantidad de kilometraje de retorno.	Correcto	Se calcula el kilometraje con el que debe regresar el vehículo. ✓	La casilla bloqueada al momento de hacer clic sobre ella se calcula el kilometraje de retorno.
		Incorrecto	No se calcula el kilometraje de retorno. ✓	La casilla permanece sin datos, sino se ingresa el kilometraje de salida o sino se presenta la cantidad de kilómetros de la ruta determinada.
<b>Hora salida y hora retorno</b>	Horas de la salida y retorno.	Correcto	Se ingresan las horas de salida y retorno previstas para el vehículo. ✓	Las casillas son rellenas por teclado, controlando que solo se ingresen caracteres apropiados y en formatos de horas.
<b>Guardar</b>	Botón de guardado.	Correcto	Guarda los datos ✓	Se almacenan los datos presentando un mensaje de que se almacenaron con éxito.
		Incorrecto	No guarda los datos. ✓	Se presentan mensajes en las casillas donde no se haya rellenado o presentado los datos necesarios, ya que todos los campos son requeridos.

8. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto
Se debería presentar un mensaje cuando no se ingresa el kilometraje de salida.	<input checked="" type="checkbox"/> Paso <input type="checkbox"/> Falló

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA

<b>CASO DE PRUEBA No.</b>	SP-03-03 SP-04-03	VERSIÓN DE EJECUCIÓN	01
		FECHA EJECUCIÓN	25/08/2015

<b>CASO DE USO:</b>		Emitir Orden de Combustible. Despacho de Combustibles.	<b>MODULO DEL SISTEMA</b>	Gestión Movilización	
<b>Descripción del caso de prueba:</b>		Se desea probar la orden de combustible que será entregada al conductor en esta se especifica tipo de combustible y cantidad asignada. Estos deben quedar guardado, para luego emitir los reportes.			
9. CASO DE PRUEBA					
i. Precondiciones					
<ul style="list-style-type: none"> <li>Aprobar la solicitud de movilización.</li> <li>Asignar vehículo.</li> <li>Hacer clic en el Gestión Movilización y luego en Despacho de Combustibles o después de guardar la asignación clic en Asignar Combustible.</li> </ul>					
j. Pasos de la prueba					
<ul style="list-style-type: none"> <li>Aprobar la solicitud de movilización.</li> <li>Asignar vehículo.</li> <li>Clic en Asignar Combustible.</li> <li>O clic en Gestión de Movilización</li> <li>Clic en Despacho de Combustibles.</li> </ul>					
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE	RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI NO	
<b>Fecha</b>	Fecha de creación de la orden de combustibles.	Correcto	Se despliega el calendario con las fechas actuales.	✓	Despliega el calendario con formatos de fechas actuales.
<b>Partida de Combustibles</b>	Tipo de proyecto al que se le asigna la orden.	Correcto	Al hacer clic se despliega la lista partidas actuales.	✓	Datos almacenados en la base de datos de las partidas o proyectos.
<b>Gasolinera</b>	Estación de servicio de combustibles.	Correcto	Se despliega los nombres de las gasolineras.	✓	Datos almacenados en la base de datos de las gasolineras.
<b>Con cargo a la cuenta de:</b>	Nombre del chofer.	Correcto	Nombre del chofer que se asignó a dicha movilización, la casilla esta bloqueada	✓	Esta casilla se rellena con los datos del chofer que fue asignado para determinada solicitud de movilización.
<b>Ruta</b>	Detalle de la ruta	Correcto	Se presenta la ruta seleccionada en la solicitud de movilización.	✓	Esta casilla está bloqueada para evitar distorsión de la ruta cuando inicialmente se la determino.
<b>Maquinaria/Vehí</b>	Modelo y placa del	Correcto	Se presenta el modelo y la placa del vehículo asignado	✓	Esta casilla está bloqueada, ya que

<b>culo</b>	vehículo.		para la movilización		anteriormente se asignó el vehículo.
<b>Kilometraje recorrido</b>	Cantidad de kilometraje recorrido.	Correcto	Se presenta el kilometraje de la ruta determinada.	✓	Esta casilla presenta el kilometraje de la ruta determinada, y no se encuentra bloqueada ya que puede variar el kilometraje recorrido en vista de percances.
<b>Kilometraje salida</b>	Cantidad de kilómetros de salida.	Correcto	Se ingresa el kilometraje con el que parte el vehículo.	✓	Casilla ingresada desde el teclado, controlando que se ingresen datos numéricos.
<b>Kilometraje retorno</b>	Cantidad de kilometraje de retorno.	Correcto	Se calcula el kilometraje con el que debe regresar el vehículo.	✓	La casilla bloqueada al momento de hacer clic sobre ella se calcula el kilometraje de retorno.
		Incorrecto	No se calcula el kilometraje de retorno.	✓	La casilla permanece sin datos, sino se ingresa el kilometraje de salida o sino se presenta la cantidad de kilómetros de la ruta determinada.
<b>Hora salida y hora retorno</b>	Horas de la salida y retorno.	Correcto	Se ingresan las horas de salida y retorno previstas para el vehículo.	✓	Las casillas son rellenadas por teclado, controlando que solo se ingresen caracteres apropiados y en formatos de horas.
<b>Guardar</b>	Botón de guardado.	Correcto	Guarda los datos	✓	Se almacenan los datos presentando un mensaje de que se almacenaron con éxito.
		Incorrecto	No guarda los datos.	✓	Se presentan mensajes en las casillas donde no se haya rellenado o

				presentado los datos necesarios, ya que todos los campos son requeridos.
<b>Orden de Submenú Combustible</b>	Correcto	Al ser guardada el despacho de combustible se envía esta orden al submenú de Orden de Combustibles.	✓	Se imprime la orden de combustibles con los datos almacenados en el submenú Orden Combustible.
	Incorrecto	Si no se guarda el despacho	✓	No se imprime la orden.

10. RESULTADOS DE LA PRUEBA

<b>Defectos y desviaciones</b>	<b>Veredicto</b>
Ninguno.	<input checked="" type="checkbox"/> Paso <input type="checkbox"/> Falló

**INFORMACIÓN GLOBAL DEL CASO DE PRUEBA**

<b>CASO DE PRUEBA No.</b>	SP-05-04	<b>VERSIÓN DE EJECUCIÓN</b>	01
		<b>FECHA EJECUCIÓN</b>	25/08/2015
<b>CASO DE USO:</b>	Emitir Orden de Mantenimiento	<b>MODULO DEL SISTEMA</b>	Talleres
<b>Descripción del caso de prueba:</b>	Se desea probar la emisión de una orden de mantenimiento ingresada por el coordinador previa solicitud del conductor se especifica el tipo de mantenimiento y los repuestos que se deben asignar y el trabajo a realizarse.		

11. CASO DE PRUEBA

k. Precondiciones

- Solicitud de mantenimiento o reparación por los conductores.
- Chequeo Vehicular.
- Ingresar al sistema.
- Entrar al menú de talleres.

l. Pasos de la prueba

- Solicitud de mantenimiento por parte de los conductores.
- Clic en Talleres.
- Clic en Solicitud de Mantenimiento.
- Conexión a la base de datos de activos.

**DATOS DE ENTRADA**

CAMPO	VALOR	TIPO ESCENARIO	RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
				SI	NO	
<b>Fecha</b>	Fecha de creación de la orden de solicitud de mantenimiento.	Correcto	Se despliega el calendario con las fechas actuales.	✓		Despliega el calendario con formatos de fechas actuales.
		Correcto	Al momento de escribir una placa vehicular, busca los datos del vehículo al hacer clic en el	✓		Búsqueda de los datos almacenados en la base de

<b>Buscar</b>	Botón y casilla de búsqueda de vehículos.		Incorrecto	botón buscar con la placa ingresada. Se muestra mensaje de que el vehículo no se encuentra registrado.	✓	datos de los vehículos. No se produce ninguna búsqueda y se muestra un mensaje de error, y los demás campos siguen bloqueados.
<b>Marca</b>	Marca del Vehículo	del	Correcto	Casilla bloqueada con la marca del vehículo.	✓	Marca del vehículo extraída de la base de datos.
<b>Tipo</b>	Tipo del Vehículo	del	Correcto	Casilla bloqueada con el tipo del vehículo.	✓	Tipo del Vehículo extraída de la base de datos.
<b>Modelo</b>	Modelo del Vehículo	del	Correcto	Casilla bloqueada con el modelo del vehículo.	✓	Modelo del Vehículo extraída de la base de datos.
<b>Núm. GPL</b>	Número del Vehículo registrado en el GPL	del	Correcto	Casilla bloqueada con el número del vehículo asignado en el GPL.	✓	Número del Vehículo extraída de la base de datos.
<b>Motor</b>	Motor del Vehículo	del	Correcto	Casilla bloqueada con el motor del vehículo.	✓	Motor del Vehículo extraída de la base de datos.
<b>Chasis</b>	Chasis del Vehículo	del	Correcto	Casilla bloqueada con el chasis del vehículo.	✓	Chasis del Vehículo extraída de la base de datos.
<b>Placa</b>	Placa del Vehículo	del	Correcto	Casilla bloqueada con la placa del vehículo.	✓	Placa del Vehículo extraída de la base de datos.
<b>Año</b>	Año de fabricación del Vehículo	de	Correcto	Casilla bloqueada con el Año de fabricación del vehículo.	✓	Año de fabricación del Vehículo extraída de la base de datos.
<b>Operador</b>	Nombre conductor asignado al vehículo.	del	Correcto	Casilla bloqueada con el Nombre del conductor asignado al vehículo.	✓	Nombre del conductor asignado al vehículo extraído de la base de datos.
<b>Combustible</b>	Tipo de combustible del Vehículo	de	Correcto	Casilla bloqueada con el Tipo de combustible del vehículo.	✓	Tipo de combustible del Vehículo extraída de la base de datos.
<b>Tipo Vehículo</b>	Pesado/Liviano		Correcto	Casilla bloqueada con el Tipo del vehículo (Pesado/Liviano).	✓	Tipo del Vehículo (Pesado/Liviano) extraída de la base de datos.

<b>Kilometraje del último mantenimiento.</b>	Cantidad de kilometraje del anterior mantenimiento.	Correcto	Casilla bloqueada con la cantidad de kilometraje del último mantenimiento.	✓	Cantidad de kilometraje del último mantenimiento extraído de la base de datos.
<b>Kilometraje de la última movilización.</b>	Cantidad de kilometraje de la anterior movilización.	Correcto	Casilla bloqueada con la cantidad de kilometraje de la última movilización.	✓	Cantidad de kilometraje de la última movilización extraída de la base de datos.
<b>Kilometraje Actual</b>	Kilometraje del vehículo actual.	Correcto	Se ingresa por teclado el kilometraje del vehículo actual.	✓	Si el kilometraje es mayor al del vehículo que actualmente registra se habilita las opciones de mantenimiento o reparación.
		Incorrecto	Mensaje de error	✓	Si el kilometraje es menor, se lanza un mensaje de error y las opciones siguen bloqueadas.
<b>Tipo</b>	Mantenimiento / Reparación	Correcto	Se despliega la lista de Mantenimiento/Reparación	✓	Combo box que se activa tras ingresar el kilometraje mayor al último mantenimiento.
<b>Reparación</b>	Reparación	Correcto	Al hacer clic en reparación se habilitan las casillas detalle de desperfectos y trabajo realizado.	✓	Se habilitan las casillas.
<b>Mantenimiento</b>	Mantenimiento	Correcto	Al hacer clic en reparación se habilitan las casillas detalle de desperfectos y trabajo realizado.	✓	Se habilitan las casillas con los datos para el mantenimiento extraídos del plan de mantenimientos preventivos dependiendo del kilometraje ingresado.
		Incorrecto	Al hacer clic en reparación se habilitan las casillas detalle de desperfectos y trabajo realizado y barra con mantenimientos disponibles.	✓	Se habilitan las casillas con la barra de mantenimientos con los que se puede iniciar para el mantenimiento extraído del plan de mantenimientos preventivos dependiendo

						del kilometraje ingresado.
<b>Link Agregar Ítems</b>	Link		Correcto	Se habilitan las casillas para agregar repuestos.	✓	Se habilitan las casillas con datos de repuestos extraídos de la base de datos.
<b>Seleccione</b>	Listas de Repuestos Vehículo	de pro	Correcto	Se despliega la lista de repuestos por marcas de vehículos.	✓	Al hacer clic sobre una marca de repuestos para los vehículos, se habilita otra casilla de repuestos específicos para esa marca de vehículo.
<b>Seleccione</b>	Lista de repuestos para la marca seleccionada	de marca	Correcto	Se despliega la lista de repuestos con la cantidad disponible.	✓	Al hacer clic sobre el repuesto automáticamente se presenta el nombre en la casilla bloqueada de ítem.
<b>Cantidad</b>	Cantidad repuestos		Correcto	Casilla rellena desde teclado.	✓	Se controla que no se exceda al número de disponibles por repuestos y que sean solo datos numéricos enteros.
<b>Ítem</b>	Nombre repuesto	del	Correcto	Se presenta el nombre del repuesto seleccionado.	✓	Casilla permanece bloqueada,
<b>Cantidad, Descripción, Controles.</b>	Datos de repuestos seleccionados.	de los	Correcto	Se visualizan los datos de los repuestos seleccionados.	✓	Se visualizan los datos de los repuestos con un icono de control "eliminar" para el repuesto que no se desea solicitar.
<b>Guardar</b>	Botón guardado.	de	Correcto	Guarda los datos	✓	Se almacenan los datos presentando un mensaje de que se almacenaron con éxito, y se imprime la solicitud de mantenimiento.

## 12. Defectos y desviaciones

## Veredicto

Ninguno

Paso

Falló

**INFORMACIÓN GLOBAL DEL CASO DE PRUEBA**

<b>CASO DE PRUEBA No.</b>	SP-06-05	<b>VERSIÓN DE EJECUCIÓN</b>	01
		<b>FECHA EJECUCIÓN</b>	25/08/2015
<b>CASO DE USO:</b>	Consultar Reportes	<b>MODULO DEL SISTEMA</b>	Reportes

**Descripción del caso de prueba:**

Se desea probar la generación de reportes que requieren los usuarios en cuanto a movilización y mantenimiento.

13. CASO DE PRUEBA

m. Precondiciones

- Ingresar al sistema.
- Datos almacenados.

n. Pasos de la prueba

- Clic en el módulo de Reportes.
- Clic en cualquier submenú de los reportes como pueden ser;
  - Listado de Vehículos.
  - Listado de Conductores.
  - Control de Mantenimiento
  - Provisión de Combustibles.
  - Matricula y SOAT.
  - Ordenes de Movilización
  - Novedades.

DATOS DE ENTRADA

CAMPO	VALOR	TIPO ESCENARIO	RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
				SI	NO	
<b>Fecha desde y Fecha hasta</b>	Fechas con las que se desea buscar un reporte.	Correcto	Se despliega el calendario de fechas actuales.	✓		Despliega el calendario con formatos de fechas actuales.
<b>Conductor</b>	Nombre del conductor.	Correcto	Se busca el nombre del conductor en la lista.	✓		Se despliega una lista con los nombres de los conductores almacenados en la base de datos.
<b>Vehículo</b>	Modelo y placa	Correcto	Se busca el vehículo por el modelo y la placa en la lista.	✓		Se despliega una lista con los vehículos almacenados en la base de datos.
<b>Funcionario:</b>	Nombre del empleado.	Correcto	Se busca el nombre del empleado en la lista	✓		Se despliega una lista con los nombres de los empleados almacenados en la base de datos.

<b>Dirección Coordinación</b>	Nombre de los departamentos.	Correcto	Se busca el nombre del departamento en la lista.	✓	Se despliega una lista con los nombres de los departamentos almacenados en la base de datos.
<b>Tipo</b>	Tipo del vehículo	Correcto	Se busca el tipo del vehículo en la lista.	✓	Se despliega en la lista los tipos de vehículos.
<b>Generar Reporte</b>	Botón.	Correcto	Al hacer clic se genera el reporte en base a las búsquedas que se seleccionaron.	✓	Se presenta en una tabla los datos de acuerdo a lo seleccionado.
<b>Exportar Reporte</b>	Botón	Correcto	Genera un archivo Excel con los datos de consulta.	✓	Se genera el archivo Excel con los datos de búsqueda seleccionados.
<b>Vigencia</b>	Alerta	Correcto	Se muestra en la tabla de Matrícula y SOAT.	✓	Cuando se encuentra en vigencia tanto la matrícula como el SOAT se muestran imágenes de visto correcto con un aviso de vigencia.
<b>Vencida</b>	Alerta	Correcto	Se muestra en la tabla de Matrícula y SOAT.	✓	Cuando se encuentra en vencido tanto la matrícula como el SOAT se muestran imágenes de agotado con un aviso de vencido.

#### 14. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto
Ninguno	<input checked="" type="checkbox"/> Paso <input type="checkbox"/> Falló

#### INFORMACIÓN GLOBAL DEL CASO DE PRUEBA

<b>CASO DE PRUEBA No.</b>	SP-07-06	<b>VERSIÓN DE EJECUCIÓN</b>	01
		<b>FECHA EJECUCIÓN</b>	25/08/2015
<b>CASO DE USO:</b>	Administración	<b>MODULO DEL SISTEMA</b>	Administración

**Descripción del caso de prueba:** Se desea probar con el control de usuarios y roles, asignación vehículo/chofer y el ingreso y almacenamiento de información importante como son:

- Tipo de Licencias.
- Combustible.
- Gasolinera.
- Registrar Vehículo.
- Registrar Conductor.

#### 15. CASO DE PRUEBA

o. Precondiciones

- Ingresar al sistema.
- Información,
- Conexión a bases externas como son:
  - Fulltime
  - Bienes
  - Activos

p. Pasos de la prueba

- Clic en el módulo de Administración.
- Clic en cualquier submenú de los reportes como pueden ser;
  - Usuarios
  - Asignación Vehículo/Chofer
  - Tipo Licencia.
  - Combustible
  - Gasolinera
  - Registrar Vehículo.
  - Registrar Conductor.

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
<b>Usuarios</b>						
<b>Añadir</b>	Botón	Correcto	Se presenta el formulario para añadir un nuevo usuario	✓		Se presenta el formulario para añadir el usuario con los menús que podrá manejar dependiendo de su rol.
<b>Actividades Usuarios</b>	Nombre del conductor.	Correcto	Se presenta el formulario de búsqueda.	✓		Formulario de logs de los usuarios.
<b>Editar</b>	Icono	Correcto	Formulario para editar usuarios.	✓		Se cargan los datos almacenados del usuario para ser editados.
<b>Eliminar</b>	Icono	Correcto	Al hacer clic se muestra el mensaje de verificación del borrado.	✓		Si se acepta en el mensaje se borra el registro del usuario de toda la base de datos y si no se acepta el registro permanece almacenado.
<b>Nombre</b>	Nombre del usuario	Correcto	Se registra el nombre del usuario	✓		Casilla que se rellena con el teclado.
<b>Usuario</b>	Nick del usuario.	Correcto	Se registra el Nick del usuario.	✓		Casilla que se rellena con el teclado.
<b>Password</b>	Contraseña del usuario	Correcto	Se registra la contraseña del usuario.	✓		Casilla que se rellena con el teclado.
	Usuarios. Tipo Licencia. Gasolinera.			✓		Las casillas son de verificación se almacenan

<b>Casillas de menús y submenús.</b>	Registrar Conductor. Solicitud de Movilización. Orden de Combustible. Solicitud de Mantenimiento. Plan de mantenimiento preventivo. Listado de vehículos. Control de Mantenimiento. Matricula y SOAT. Novedades. Asignar vehículo/chofer. Combustible. Registrar Vehículo. Aprobación. Despacho Combustible. Historial de Mantenimiento. Listado de Conductores. Provisión de combustibles. Ordenes de movilización.	Correcto	Se selecciona las casillas de los submenús que puedan manipular los usuarios de acuerdo a su rol.	de acuerdo a las seleccionadas.
<b>Guardar</b>	Botón de guardado	Correcto	Guarda los datos	Se almacenan los datos de los usuarios.
<b>Asignar Vehículo / Chofer</b>				
<b>Eliminar</b>	Icono.	Correcto	Al hacer clic en el icono de eliminar se borra el conductor.	Este icono permite eliminar el conductor a ese vehículo para asignar a otro conductor.
<b>Asignar</b>	Icono	Correcto	Se cargan un listado con los datos de los conductores.	AL seleccionar algún nombre de la lista de conductores se asigna a ese vehículo un conductor.
<b>Fecha de asignación:</b>	Formato de fecha.	Correcto	Se despliega el calendario de fechas.	Calendario de fechas actuales.
<b>Chofer</b>	Nombre del chofer	Correcto	Lista con los nombres de los choferes	Nombres de los choferes extraídos de la base de datos.
<b>Acta de entrega.</b>	Barra de carga de archivos.	Correcto	Se selecciona un archivo.	Carga archivos de formato PDF, EXCEL Y WORD.
<b>Detalle</b>	Descripción de la asignación	Correcto	Se ingresa el detalle de la asignación	Casilla que se rellena desde teclado.

<b>Revisado / Confirmado</b>	Casillas de verificación	Correcto	Se selecciona SI o NO	✓	Casillas de verificación.
<b>Guardar la Asignación</b>	Botón.	Correcto	Almacena los datos de las asignaciones.	✓	Guarda los nuevos datos.
<b>Visualizar</b>	Icono	Correcto	Carga un formulario.	✓	Se visualizan todos los datos del vehículo y del chofer asignado. Permitiendo además de editar el detalle, la acta actual (cargando otro archivo), y si ha sido revisado y confirmado.
<b>Guardar la Visualización</b>	Botón.	Correcto	Almacena las ediciones de la visualización.	✓	Guarda los nuevos datos.
<b>Tipo Licencia</b>					
<b>Añadir</b>	Botón	Correcto	Carga Formulario.	✓	Abre el formulario para registra el tipo de licencia.
<b>Eliminar</b>	Icono	Correcto	Al hacer clic se muestra el mensaje de verificación del borrado.	✓	Si se acepta en el mensaje se borra el registro del tipo de licencia de toda la base de datos y si no se acepta el registro permanece almacenado.
<b>Editar</b>	Icono	Correcto	Formulario para editar el tipo de licencia.	✓	Se cargan los datos almacenados de las licencias para ser editados.
<b>Tipo licencia</b>	Nombre del tipo de licencia	Correcto	Se registra el nombre del tipo de licencia.		Casilla que se rellena desde teclado
<b>Guardar el nuevo registro.</b>	Botón.	Correcto	Almacena el nuevo registro del tipo de licencia.	✓	Guarda los nuevos datos.
<b>Combustible</b>					
<b>Añadir</b>	Botón	Correcto	Carga Formulario.	✓	Abre el formulario para registra el tipo de combustible.
<b>Eliminar</b>	Icono	Correcto	Al hacer clic se muestra el mensaje de verificación del borrado.	✓	Si se acepta en el mensaje se borra el registro del tipo de combustible de toda la base de datos y si no se acepta el registro permanece almacenado.

<b>Editar</b>	Icono	Correcto	Formulario para editar el tipo de combustible	✓	Se cargan los datos almacenados de los combustibles para ser editados.
<b>Nombre del Combustible.</b>	Nombre del Combustible	Correcto	Se registra el nombre del combustible.	✓	Casilla que se rellena desde teclado.
<b>Precio</b>	Valor del galón de combustible.	Correcto	Se registra el valor del galón de combustible.	✓	Casilla que se rellena desde teclado, permitiendo la entrada solo de caracteres enteros y decimales.
<b>Guardar el nuevo registro.</b>	Botón.	Correcto	Almacena el nuevo registro del tipo de combustible.	✓	Guarda los nuevos datos.

#### Gasolinera

<b>Añadir</b>	Botón	Correcto	Carga Formulario.	✓	Abre el formulario para registrar la estación de servicios.
<b>Eliminar</b>	Icono	Correcto	Al hacer clic se muestra el mensaje de verificación del borrado.	✓	Si se acepta en el mensaje se borra el registro de la gasolinera de toda la base de datos y si no se acepta el registro permanece almacenado.
<b>Editar</b>	Icono	Correcto	Formulario para editar la estación de servicios.	✓	Se cargan los datos almacenados las gasolineras para ser editados.
<b>Nombre de la Gasolinera</b>	Nombre de la Gasolinera	Correcto	Se registra el nombre de la gasolinera		Se rellena la casilla desde teclado.
<b>Guardar el nuevo registro.</b>	Botón.	Correcto	Almacena el nuevo registro de las gasolineras.	✓	Guarda los nuevos datos.

#### Registrar Vehículo

<b>Añadir</b>	Botón	Correcto	Carga Formulario.	✓	Abre el formulario para registrar los vehículos.
<b>Eliminar</b>	Icono	Correcto	Al hacer clic se muestra el mensaje de verificación del borrado.	✓	Si se acepta en el mensaje se borra el registro del vehículo de toda la base de datos y si no se acepta el registro permanece

					almacenado.
<b>Editar</b>	Icono	Correcto	Formulario para editar los vehículos.	✓	Se cargan los datos almacenados los vehículos para ser editados.
<b>Vehículo (Seleccione)</b>	Lista de Vehículos.	Correcto	Se listan todos los vehículos del parque automotor del GPL.	✓	Datos extraídos de la base de datos externa de activos.
<b>Precio Adquisición. Chasis. Motor. Placa. Año Fabricación. Color. Código del bien. Condición real Modelo. Marca. Serie. Novedades. Tipo Comprobante. Fecha Adquisición.</b>	Datos extraídos.	Correcto	Se presentan en las casillas los datos del vehículo seleccionado.	✓	Todos estos datos son extraídos de la base de datos externa de activos y se presentan en las casillas que se encuentran bloqueadas.
<b>Combustible</b>	Tipo de combustible.	Correcto	Tipo de combustible para asignar al vehículo.	✓	Datos extraídos de la base de datos dbmovil.
<b>Estado Pertenencia.</b>	Nombre de la entidad de pertenencia.	Correcto.	Nombre de la entidad a la que pertenece el vehículo.	✓	Datos extraídos de la base de datos dbmovil.
<b>Núm. Vehículo.</b>	Número del vehículo.	Correcto	Número que se le asigna al vehículo en el GPL para registrarlo.	✓	Casilla rellena desde teclado, controlando el ingreso solo de datos numéricos.
<b>Casa Comercial</b>	Casa Comercial.	Correcto	Casa comercial de donde se adquirió el vehículo.	✓	Casilla rellena desde teclado.
<b>País de Origen</b>	País de Origen.	Correcto	País de origen de donde se lo importó o fabricó al vehículo.	✓	Casilla rellena desde teclado.
<b>Tonelaje</b>	Tonelaje	Correcto	Tonelaje que resiste el vehículo.	✓	Casilla rellena desde teclado, controlando el ingreso solo de datos numéricos.
<b>Rendimiento por galón.</b>	Rendimiento del Vehículo.	Correcto	Rendimiento del vehículo por galón de combustible.	✓	Casilla rellena desde teclado, controlando el ingreso solo de datos numéricos.
<b>Tipo</b>	Pesado/Liviano	Correcto	Tipo vehículo (Pesado/Liviano).	✓	Datos extraídos de la base de datos dbmovil.

<b>Fecha Vencimiento SOAT</b>	Formato de fechas.	de	Correcto	Calendario con fechas actuales.	✓	Se despliega el calendario de fechas actuales.
<b>Fecha Vencimiento Matricula</b>	Formato de fechas.	de	Correcto	Calendario con fechas actuales.	✓	Se despliega el calendario de fechas actuales.
<b>Notificaciones de talleres.</b>	Notificaciones de desperfectos.		Correcto	Notificaciones desde talleres de algún desperfecto o accidente del vehículo.	✓	Caja de texto amplia rellena desde teclado.
<b>Foto vehículo</b>	Imagen.		Correcto	Imagen del vehículo.	✓	Link para subida de archivos, jpg., jpeg., png.
<b>Cambiar foto</b>	Imagen		Correcto	Imagen nueva del vehículo.	✓	Link para subir la nueva foto.
<b>Guardar</b>	Botón		Correcto	Guarda los datos	✓	Almacena los datos en la base de datos dbmovil.
			Incorrecto	Mensajes incompleto de el formulario.	✓	Campos obligatorios.
<b>Registra Conductor</b>						
<b>Añadir</b>	Botón		Correcto	Carga Formulario.	✓	Abre el formulario para registrar los conductores.
<b>Eliminar</b>	Icono		Correcto	Al hacer clic se muestra el mensaje de verificación del borrado.	✓	Si se acepta en el mensaje se borra el registro de los conductores de toda la base de datos y si no se acepta el registro permanece almacenado.
<b>Editar</b>	Icono		Correcto	Formulario para editar a los conductores.	✓	Se cargan los datos almacenados los conductores para ser editados.
<b>Conductor (Seleccione)</b>	Lista		Correcto	Lista de cedulas de los conductores que faltan registrarse.	✓	Se despliega las cedulas de los conductores a ser registrados desde la base de datos fulltime.
<b>Nombres. Apellidos. Cedula. Cargo. Correo. Departamento. Cargo.</b>	Datos extraídos.		Correcto	Datos extraídos al seleccionar una cedula.	✓	Datos del conductor extraídos de la base de datos fulltime, que se presentan en las casillas bloqueadas.
<b>Teléfono</b>	Número telefónico.		Correcto	Número telefónico del conductor.	✓	Casilla rellena desde teclado y control de ingreso solo

						datos numéricos.
<b>Licencia.</b>	Tipo de licencia	Correcto	Tipo de licencia que posee el conductor.	✓		Despliegue de las lista con los tipos de licencias extraídos de la base de datos dbmovil.
<b>Novedades.</b>	Descripción del conductor.	Correcto	Descripción de alguna novedad del conductor.	✓		Casilla rellena desde teclado.
<b>Nombre del Banco</b>	Nombre del Banco	Correcto	Nombre del Banco donde posea una cuenta el conductor.	✓		Casilla rellena desde teclado.
<b>Tipo de Cuenta.</b>	Tipo de Cuenta bancaria.	Correcto	Tipo de cuenta bancaria que posee el conductor.	✓		Lista desplegable con los tipos de cuentas.
<b>Número de Cuenta</b>	Número de Cuenta.	Correcto	Número de cuenta del conductor.	✓		Casilla rellena desde teclado y con ingreso solo de datos numéricos.
<b>Guardar</b>	Botón	Correcto	Guarda los datos	✓		Almacena los datos en la base de datos dbmovil.
		Incorrecto	Mensajes incompleto de el formulario.	✓		Campos obligatorios.
16. RESULTADOS DE LA PRUEBA						
<b>Defectos y desviaciones</b>						<b>Veredicto</b>
Ninguno						<input checked="" type="checkbox"/> <b>Paso</b> <input type="checkbox"/> <b>Falló</b>

## ANEXO H: MANUAL DE USUARIO

### 1. Acceder al Sistema de Gestión Vehicular del GPL.

En el navegador se debe digitar la siguiente URL: <http://190.57.168.179/mobile/> y se carga la página principal para acceder al sistema.

### 2. Ingresar al Sistema de Gestión Vehicular del GPL.

Se digita el usuario y contraseña asignados para poder acceder al sistema, ingresándolos respectivamente donde se muestra en letras en las cajas de texto **Username** y **Contraseña**.



Figura 50. Pantalla de Inicio de Sesión  
Elaboración: (Ordóñez & Guaman, 2015)

### 3. Menú Principal

Este menú se carga una vez que se haya ingresado con las credenciales correctas, y los submenús se cargan de acuerdo a los privilegios otorgados para cada usuario del sistema.

El menú muestra de manera estructurada:

- Nombre del usuario que accedió.
- Inicio
- Administración
- Reportes.
- Gestión Movilización
- Talleres.
- Salir.

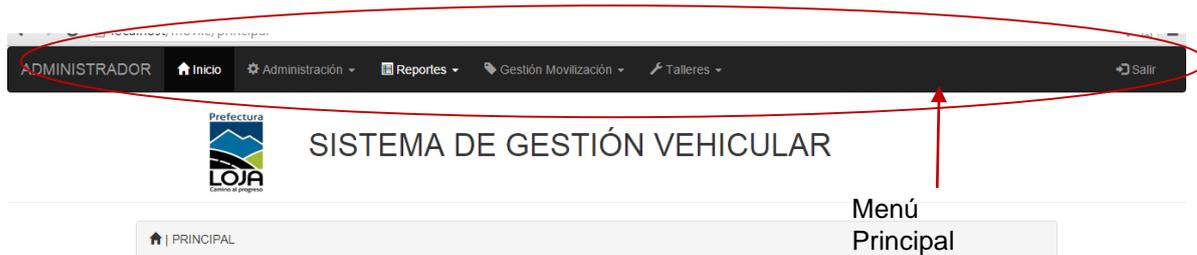


Figura 51. Pantalla de Ingreso al Sistema de Gestión Vehicular.  
Elaboración: (Ordóñez & Guaman, 2015)

#### 4. Módulo de Administración

Es un módulo visible en el menú principal, que al hacer clic en el nombre “**Administración**” se despliega el siguiente submenú:

- Usuarios.
- Asignación vehículo/chofer.
- Tipo Licencia.
- Combustible.
- Gasolinera.
- Registrar Vehículo.
- Registrar Conductor.

##### 4.1. Usuarios

Este submenú nos permite administrar las cuentas de usuarios, habilitarle o deshabilitarle privilegios, eliminar, actualizar y agregar usuarios, así, como también una lista visual de todos los usuarios registrados.

Tabla con todos los usuarios registrados, visualizando el nombre y login del usuario.

ADMINISTRACIÓN DE USUARIOS		+ Añadir + Actividades Usuarios	
NOMBRE USUARIO	NOMBRE LOGIN	ACCIONES	
ALBITA BARRIGA	abarriga		
FRANCISCO PALACIO	fpalacio		
Prueba1	Prueba1		
Andrea	Prueba		
pruebas2	pruebas2		
YONDE ORDOÑEZ	ylordonez		

Permiten editar los datos, y privilegios de los usuarios.

Permiten borrar a los usuarios.

Figura 52. Interfaz de la administración de usuarios.  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en “Añadir” se nos habilita el formulario para registrar al usuario.



Figura 53. Interfaz para añadir un Usuario al Sistema.

Elaboración: (Ordóñez & Guaman, 2015)

Clic en Añadir.

Formulario de registro de usuario.

Figura 54. Formulario para registrar un usuario al sistema.

Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en “**Actividades Usuarios**” nos permite buscar mediante una fecha de inicio y fin las actividades de los usuarios, en otras palabras los logs que son las actualizaciones y las borradas de datos que se realicen por un usuario específico.



Figura 55. Interfaz para listar las actividades de los usuarios.

Elaboración: (Ordóñez & Guaman, 2015)

Una vez que se hace clic en “Actividades Usuarios” se carga el siguiente formulario de búsqueda para obtener los logs de los usuarios.

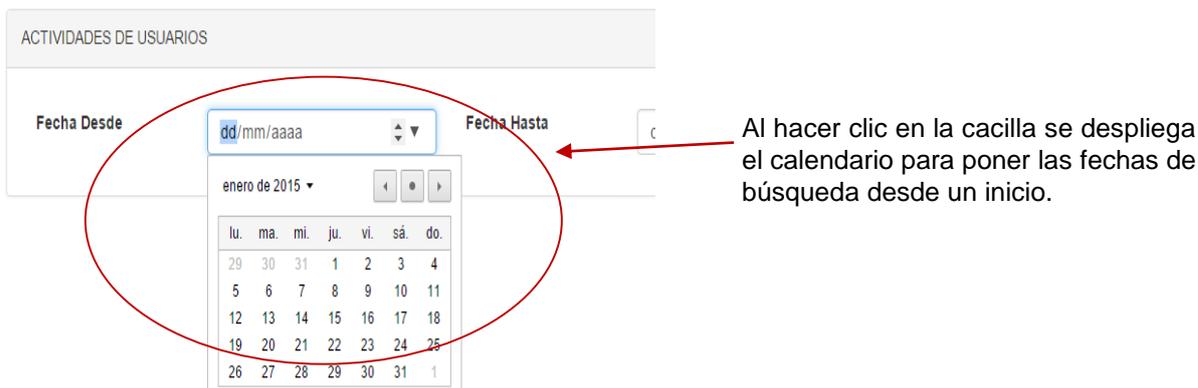


Figura 56. Fecha de búsqueda para listar los usuarios.  
Elaboración: (Ordóñez & Guaman, 2015)

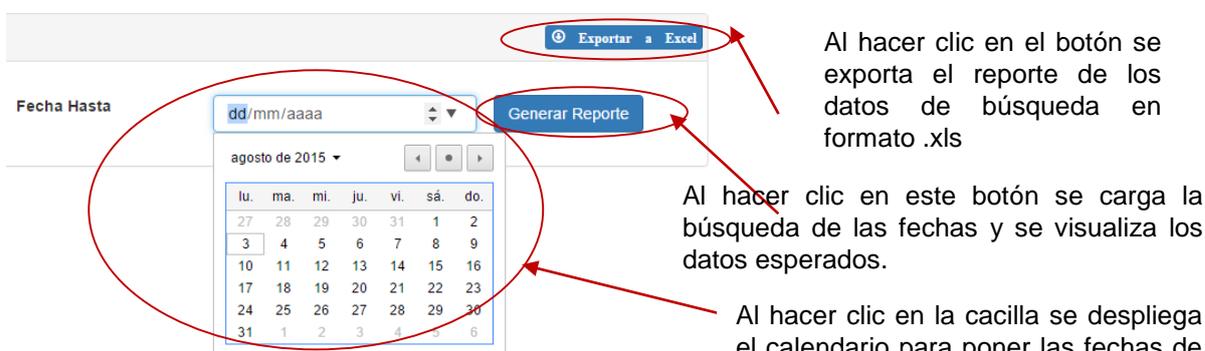


Figura 57. Campos de búsqueda para listar los usuarios.  
Elaboración: (Ordóñez & Guaman, 2015)

Cuando se hace clic en **“Generar Reporte”** se nos presenta la siguiente visualización del reporte dependiendo de las fechas desde y fechas hasta.

Fecha Desde 01/01/2015 Fecha Hasta 31/08/2015 **Generar Reporte**

NUMERO	NOMBRE TABLA	NOMBRE OPERACIÓN	FECHA
1	movilizacion	U	2015-05-16
2	movilizacion	U	2015-05-16
3	movilizacion	U	2015-05-16
4	movilizacion	I	2015-05-16
5	movilizacion	U	2015-05-16
6	movilizacion	U	2015-05-16

1 2 3 4 5 »

Figura 58. Reporte de logs.  
Elaboración: (Ordóñez & Guaman, 2015)

Cuando se hace clic en **“Exportar a Excel”** se descarga la visualización de los datos en formato .xls.

#### 4.2. Asignación vehículo/chofer

Al hacer clic en el submenú se nos presenta la siguiente ventana que nos permite las siguientes opciones y los datos de los vehículos ya asignados y los no asignados a un chofer.

VEHICULOS Y CHOFERES						
NÚMERO	TIPO	MARCA	PLACA	CONDICIÓN	CHOFER	VISUALIZAR
1		TOYOTA	LEA0343	BUENO	NOMBRE DEL CONDUCTOR ✕	🔍
2		CHEVROLET	LEA0338	BUENO	+	
51		NISSAN PATROL	LEA0357	BUENO	+	
70		NISSAN PATHFINDER	LEA0423	BUENO	+	
73		TOYOTA	LEA0443	BUENO	+	
97		CHEVROLET	LEI1079	BUENO	NOMBRE DEL CONDUCTOR ✕	🔍

Permite Agregar un chofer a un vehículo.

Se visualiza el nombre del chofer asignado al vehículo y el icono de borrar ✕ al lado borra al chofer asignado.



Permite la visualización de los datos del vehículo asignado a un chofer.

Figura 59. Interfaz para asignar Vehículo chofer.  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en el icono de agregar chofer  se carga la ventana para asignar el chofer.

**ASIGNAR CHOFER**

Tipo	Marca	<b>CHEVROLET</b>	Numero	<b>2</b>	Placa	<b>LEA0338</b>	
Combustible	<b>EXTRA</b>	Modelo	<b>RODEO V6 4X4 DLX T/M A/C</b>	Año de Fabricación	<b>2000</b>	Condición	<b>BUENO</b>

Fecha:

Acta de Entrega:  Ningún archivo seleccionado

Detalle:

Revisado y Confirmado:  Si  No

Chofer:

- ROMULO ITALO GUAMAN ALDAZ
- SELECCIONE--
- ANGEL MEDARDO ESPINOSA
- BLANCO MARINO SILVA BRAVO
- ROMULO ITALO GUAMAN ALDAZ**
- WILSON PEDRO LANCHE TANDAZO
- CESAR AUGUSTO POMA CISNEROS
- SEGUNDO ISAURO MORENO CHAMBA
- EDINSON JAVIER FERNANDEZ ESPINOZA
- ULVIO ZARUMA RIERA
- HIDALGO DE JESUS PINEDA
- JOSE DARIO RIGUEZ ROJAS
- SEGUNDO CARLOS VICTORIANO LIMA MIRANDA
- FRANCO VINICIO ARMIJOS CORDOVA
- VICTOR EDUARDO LOJA CALDERON
- RUTBEL YOVANY AMBULUDI CALERO
- WILMAN TEODOR ROJAS ORDÓÑEZ
- JOFFRE ULISES FIERRO JARAMILLO
- CESAR AUGUSTO COLLASORZO SANCHEZ

Figura 60. Formulario para la Asignación de un chofer a un vehículo.  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en el icono de visualizar  se carga la ventana con los datos del vehículo ya asignado a un chofer.

Tipo	Marca	TOYOTA	Numero	1	Placa	LEA0343	
Combustible	EXTRA	Modelo	LAND CRUISER PRADO VX T/M	Año de Fabricación	2002	Condicion	BUENO

Fecha:

Chofer:

Acta Actual: [Valladolid\\_Zhingre\\_Roberto\\_Carlos3.pdf](#)

Ningún archivo seleccionado

Detalle:

Revisado y Confirmado:

Si  No

Figura 61. Formulario de visualización de asignación vehículo chofer.  
Elaboración: (Ordóñez & Guaman, 2015)

### 4.3. Tipo Licencia

Al hacer clic en el submenú se nos presenta la vista visualizando los datos de las licencias guardadas con las siguientes opciones.

TIPO LICENCIA	CONTROLES	
OPERADOR DE EXCAVADORA		
OPERADOR DE TRACTOR		
TIPO A		
TIPO B		
TIPO C		
TIPO C MOTOCICLETA NIVELADURA		
TIPO D		
TIPO D1		
TIPO E		
TIPO G		
TR		

Permite borrar los datos de las licencias.

Permite editar los datos de las licencias.

Permite agregar un nuevo tipo de licencia.

Figura 62. Interfaz de administración del tipo licencias.  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en **“Añadir”** se nos presenta la ventana para insertar un nuevo tipo de licencia.

INSERTAR TIPO LICENCIAS

Tipo de Licencia:

Figura 63. Formulario Añadir nueva licencia  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en el icono de editar nos permite editar los datos de las licencias.

Figura 641. Formulario de actualización tipo licencia.  
Elaboración (Ordóñez & Guaman, 2015)

#### 4.4. Combustible

Al hacer clic en el submenú se nos presenta la venta visualizando los datos de los tipos de combustibles con las siguientes opciones.

Permite borrar los datos.

Permite editar los datos.

Permite añadir o insertar un nuevo tipo de combustibles.

Figura 652. Interfaz de administración de los combustibles.  
Elaboración: (Ordóñez & Guaman, 2015).

Al hacer clic en “**Añadir**” se nos presenta la ventana para insertar un nuevo tipo de combustible.

Figura 663. Formulario para añadir un nuevo combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en el icono de editar nos permite editar los datos de los combustibles.

Figura 67. Formulario para editar un combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

#### 4.5. Gasolinera

Al hacer clic en el submenú se nos presenta la venta visualizando los datos de las estaciones

de servicio de combustibles con las siguientes opciones.

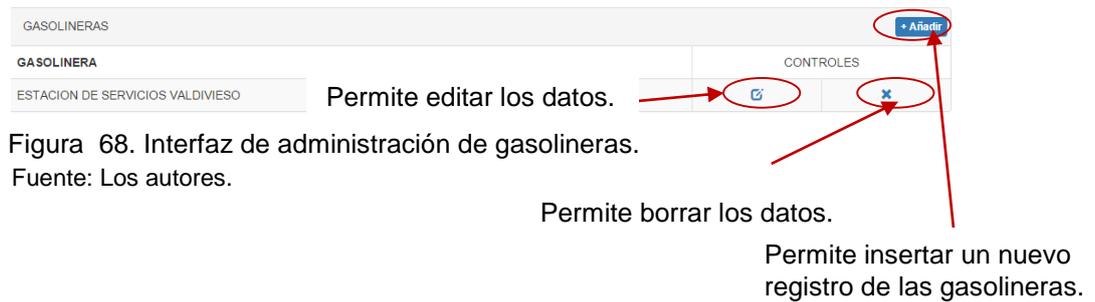


Figura 68. Interfaz de administración de gasolineras.  
Fuente: Los autores.

Al hacer clic en “**Añadir**” se nos presenta la ventana para insertar una nueva estación de servicios de combustibles.

AÑADIR GASOLINERA

Nombre de la Gasolinera:

Guardar

Figura 69. Formulario para añadir nueva gasolinera  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en el icono de editar  nos permite editar los datos de las estaciones de servicios de combustibles.

Actualizar Gasolineras

Nombre del Combustible:

ESTACION DE SERVICIOS VALDIVIESO

Guardar

Figura 70. Formulario para editar gasolinera.  
Elaboración: (Ordóñez & Guaman, 2015)

#### 4.6. Registrar Vehículo

Al hacer clic en el submenú se nos presenta la venta visualizando los datos de los vehiculos con las siguientes opciones.

REGISTRAR VEHICULO						<a href="#">+ Añadir</a>
TIPO	VEHICULO	MOTOR	CHASIS	COLOR	CONTROLES	
VOLQUETA	HINO GH8JGSD LSA1058		JHDFS1ELVCXX16628	AMARILLO		
CAMION	HINO GH8JMSA LSA1057		9F3GH8JMSCXX12936	BLANCO		
VOLQUETA	HINO FS1ELVD LSA1063		JHDFS1ELVCXX16618	AMARILLO		
JEEP	CHEVROLET RODEO 4X4 LEA437		927102093	NEGRO		
CAMIONETA	KIA PREGIO CILINDRAJE 3000 LEA323		CHEVROLET PICKUP	ROJO		
CAMION	HINO FS1ELVD LSA1069		9F3GD8JLSDXX13344	BLANCO		
VOLQUETA	DIMEX 55119075 G01126294		JNBPKC212CAE01583	AMARILLO		
JEEP	HINO FC9		IFH11VJ9529006687	AZUL		
CAMION	HINO GH8		IF3GH8JMSCXX12931	BLANCO		

Permite editar los datos.

Permite borrar los datos.

Figura 71. Interfaz para la administración de vehículos.

Elaboración: (Ordóñez & Guaman, 2015)

Permite añadir un nuevo vehículo.

Al hacer clic en “Añadir” se nos presenta la ventana para insertar un nuevo vehículo.

INSERTAR VEHICULOS

VEHICULO

Permite buscar el vehículo a insertar mediante la placa

Figura 72. Interfaz para añadir un nuevo vehículo.

Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en una placa en la barra de búsqueda se despliega los datos previamente guardados del vehículo.

PRECIO ADQUISICION	<input type="text" value="0"/>	CHASIS	<input type="text" value="YK1109016778"/>
MOTOR	<input type="text"/>	TIPO VEHICULO	<input type="text" value="CAMIONETA"/>
PLACA	<input type="text" value="LEA 370"/>	AÑO FABRICACION	<input type="text" value="1999"/>
COLOR	<input type="text" value="ROJO"/>		
CODIGO DEL BIEN	<input type="text" value="500100010005"/>	CONDICION REAL	<input type="text" value="BUENO"/>
MODELO	<input type="text" value="LUV1999"/>	MARCA	<input type="text" value="CHEVROLET PICKUP"/>
SERIE	<input type="text" value="141.01.05.01.11.07.06.003"/>	NOVEDADES	<input type="text" value="DESCRIPCION: NN OBSERVACIOI"/>
TIPO COMPROBANTE	<input type="text" value="NN"/>	FECHA ADQUISICION	<input type="text" value="1998-09-02"/>

Figura 73. Formulario para añadir un nuevo vehículo.

Elaboración: (Ordóñez & Guaman, 2015)

Y también se despliega las casillas para registrar datos no guardados del vehículo.

COMBUSTIBLE:	--SELECCIONE--	ESTADO PERTENENCIA:	--SELECCIONE--
NUM. VEHICULO:	<input type="text"/>	CASA COMERCIAL:	<input type="text"/>
PAIS ORIGEN:	<input type="text"/>	TONELAJE:	<input type="text"/>
RENDIMIENTO x GALON:	<input type="text"/>	TIPO:	--SELECCIONE--
KM/GL			
FECHA VENCIMIENTO SOAT:	dd/mm/aaaa	FECHA VENCIMIENTO MATRICULA:	dd/mm/aaaa
NOTIFICACIONES TALLERES:	<input type="text"/>		
FOTO VEHICULO:	<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado		
<input type="button" value="Guardar"/>			

Figura 74. Formulario de ingreso vehículo.  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en el icono de editar  nos permite editar los datos de los vehículos.

TIPO COMPROBANTE	NN	FECHA ADQUISICON	2012-08-22
COMBUSTIBLE	DIESEL	ESTADO PERTENENCIA	GOBIERNO PROVINCIAL
NUMERO VEHICULO GPL	41	CASA COMERCIAL	TOYOTA
PAIS DE ORIGEN	AUSTRALIA	TONELAJE	24
TIPO	LIVIANOS		
NOTIFICACIONES TALLERES	NADA		
FECHA VENCIMIENTO SOAT	12/05/2015	FECHA VENCIMIENTO MATRICULA	12/05/2015
FOTO ACTUAL		CAMBIAR FOTO	<input type="button" value="Examinar..."/> No se ha seleccionado ningún archivo.
<input type="button" value="Guardar"/>			

Figura 75. Formulario de actualización de un vehículo.  
Elaboración: (Ordóñez & Guaman, 2015)

#### 4.7. Registrar Conductor

Al hacer clic en el submenú se nos presenta la venta visualizando los datos de los conductores con las siguientes opciones.

ADM. CONDUCTORES						<input type="button" value="+ Añadir"/>
CEDULA	NOMBRES Y APELLIDOS	CORREO	CARGO	LICENCIA	CONTROLES	
1102757240	C	Permite editar los datos.	job.ec	ASISTENTE DE GUARDALMACEN	OPERADOR DE TRACTOR	
1713295358	DIEGO NEY CUEVA ABARCA	dcueva@prefecturaoja.gob.ec	ASISTENTE DE GUARDALMACEN	OPERADOR DE TRACTOR	OPERADOR DE TRACTOR	
9999999999	u u	u@jjj.com	PREFECTO/A	OPERADOR DE TRACTOR	OPERADOR DE TRACTOR	

Figura 76. Interfaz de Administración del Conductor.  
Elaboración: (Ordóñez & Guaman, 2015)

Permite borrar los datos.

Permite añadir un nuevo conductor.

Al hacer clic en **“Añadir”** se nos presenta la ventana para insertar un nuevo conductor.

Permite buscar mediante el número de cedula a los conductores.

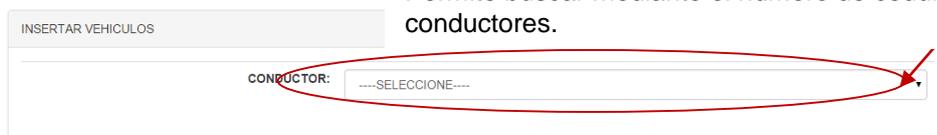


Figura 77. Interfaz para añadir un nuevo conductor.  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en un número de cedula en la barra de búsqueda se despliega los datos previamente guardados del conductor.



Figura 78. Formulario para añadir un nuevo conductor  
Elaboración: (Ordóñez & Guaman, 2015)

Y también se despliega las casillas para registrar datos no guardados del conductor.



Figura 79 Formulario para añadir un nuevo Conductor.  
Elaboración: (Ordóñez & Guaman, 2015)

## 5. Módulo Reportes

Al hacer clic sobre el nombre **“Reportes”** se despliega el submenú que se detalla a continuación:

- Control de Mantenimiento.
- Provisión de Combustibles.
- Matricula Vehicular.
- Ordenes de Movilización.
- Novedades.

### 5.1. Control de Mantenimiento

Al hacer clic en el submenú se visualiza los datos guardados de los mantenimientos realizados a los vehículos con la opción de exportarlo en Excel y también generar búsquedas avanzadas.

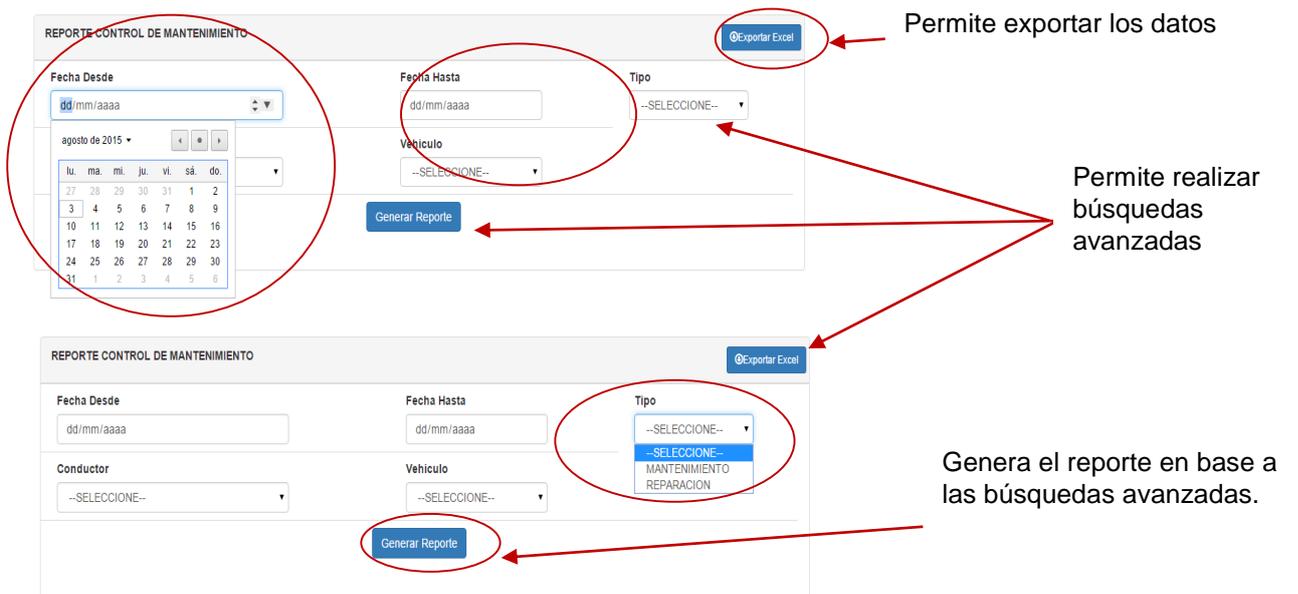


Figura 80. Interfaz para realizar un Búsqueda avanzada de mantenimiento.  
Elaboración: (Ordóñez & Guaman, 2015)

## 5.2. Provisión de Combustibles.

Al hacer clic en el submenú se visualiza los datos guardados de las provisiones de combustibles emitas para los vehículos con calculos de gastos en galones de combustible y precio, y la opción de exportarlo en Excel y también generer busquedas avanzadas.



Figura 81. Interfaz de Búsqueda avanzada para el reporte de control de combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

### 5.3. Matrícula Vehicular

Al hacer clic en el submenú se visualiza los datos guardados de los matriculas de los vehículos con las alertas de vigencia y caducidad de los mismos, y con la opción de exportarlo en Excel y también generer busquedas avanzadas.

Permite exportar el reporte

Permite visualizar las matrículas y SOAT en vigencia o vencidos.

PLACA	MARCA	TIPO DE VEHICULO	ESTADO VEHICULO	FECHA DE VENCIMIENTO MATRICULACION	FECHA DE VENCIMIENTO SOAT
LEA0352	HINO		BUENO	VIGENCIA	VIGENCIA
LEA0354	HINO		BUENO	VIGENCIA	VIGENCIA
LEA0351	HINO		BUENO	VIGENCIA	VIGENCIA
LEA0357	NISSAN PATROL		BUENO	VIGENCIA	VIGENCIA
LO9898	YAMAHA		BUENO	VENCIDO 10/03/2015	VENCIDO 27/03/2015
LEA0396	CHEVROLET		BUENO	VIGENCIA	VIGENCIA
LEA0389	CHEVROLET		BUENO	VIGENCIA	VIGENCIA

Figura 82. Interfaz del reporte de vencimiento de matrícula vehicular.  
Elaboración: (Ordóñez & Guaman, 2015)

### 5.4. Ordenes de Movilización

Al hacer clic en el submenú se visualiza los datos guardados de las movilizaciones de los vehículos con la opción de exportarlo en Excel y también generer busquedas avanzadas.

Permite Exportar el reporte

Búsqueda Avanzada

Genera el reporte en base a la Búsqueda

FECHA SALIDA	FECHA RETORNO	FUNCIONARIO RESPONSABLE	VEHICULO	NRO	RUTA	HORA/SALIDA	HORA/RETORNO	ACTIVIDAD A REALIZARSE	CONDUCTOR
2015-05-30	2015-05-31	LUIS ALONSO OCHOA SANCHEZ	HYUNDAI	1	PUYANGO-ALAMOR	09:00:00	19:00:00	MOTIVO DE PRUEBA	HECTOR GUILLERMO PONTON QUEVEDO

Figura 834. Interfaz para generar el reporte de Órdenes de Movilización.  
Elaboración: (Ordóñez & Guaman, 2015)

### 5.5. Novedades

Al hacer clic en el submenú se visualiza los datos guardados de las novedades o notificaciones de los vehículos con la opción de exportarlo en Excel y también generer busquedas avanzadas.

Permite exportar los datos del reporte en formato



REPORTE NOVEDADES		
VEHICULO	NOVEDADES	NOTIFICACIONES TALLERES
CHEVROLET - LUV V6 4X4 - LEA0389	prueba	No registra notificaciones de talleres alguna
TOYOTA - HILUX PICK UP (1981) - LEA382	DESCRIPCION: NN OBSERVACION: NN	DFGGD
HINO - GH8JMSA - LSA1056	DESCRIPCION: NN OBSERVACION: NN	fkhekr
HINO - FC9JKSZ - LEA343	DESCRIPCION: NN OBSERVACION: NN	DQD
HINO - FS1ELVD - LSA1069	DESCRIPCION: NN OBSERVACION: NN	que n o me falle

Figura 84. Interfaz para generar reporte de Novedades.  
Elaboración: (Ordóñez & Guaman, 2015)

## 6. Módulo de Gestión Movilización

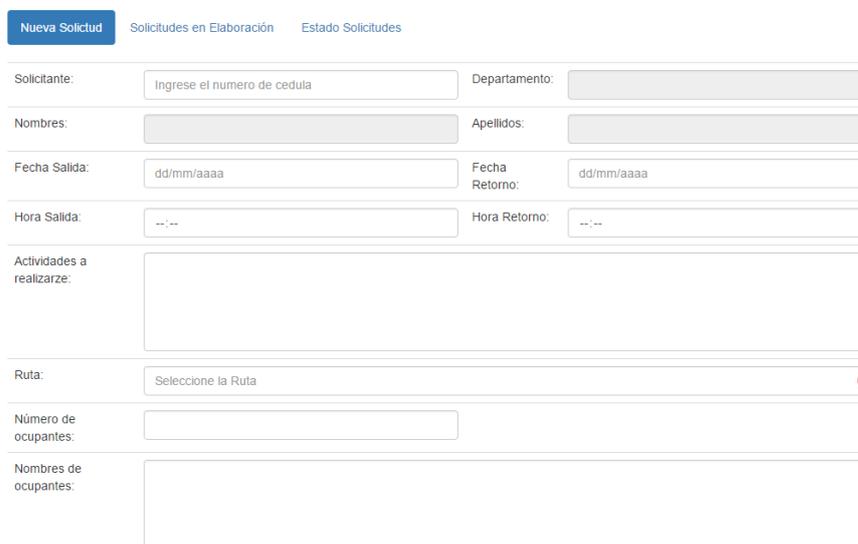
Al hacer clic sobre el nombre “**Gestión Movilización**” se despliega el submenú que se detalla a continuación:

- Solicitud de Movilización.
- Asignar Vehículo
- Orden de Combustible.
- Despacho Combustible.

### 6.1. Solicitud de Movilización

Permite crear la solicitud para la movilización en el cual se incluye el google maps para la asignación de las rutas y el control de asignamiento de combustible, esta solicitud se divide en 3 partes:

- **Nueva Solicitud**



Nueva Solicitud   Solicitudes en Elaboración   Estado Solicitudes

Solicitante:  Departamento:

Nombres:  Apellidos:

Fecha Salida:  Fecha Retorno:

Hora Salida:  Hora Retorno:

Actividades a realizar:

Ruta:  **Ruta**

Número de ocupantes:

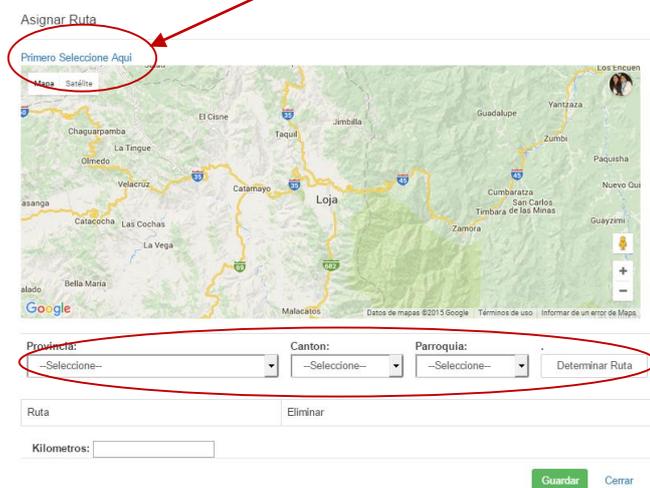
Nombres de ocupantes:

Permite escoger la ruta para Asignación de combustible.

Figura 85. Formulario para registrar una Solicitud de Movilización.  
Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en “**Ruta**” carga una ventana en la cual se visualiza el mapa y permite escoger la ruta provincial y nacional para la movilización de los vehículos, y calcula el kilometraje a recorrer.

Primero seleccione para cargar el mapa.



Permite escoger la provincia cantón y parroquia.

Figura 86. Interfaz para seleccionar la ruta  
Elaboración: (Ordóñez & Guaman, 2015)

La siguiente imagen visualiza la ruta graficada en el mapa, en letrar el recorrido y el kilometraje total.

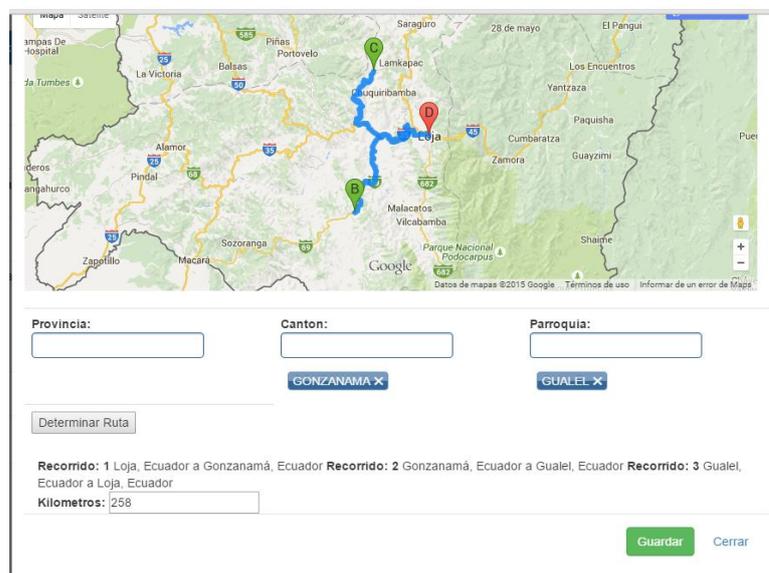


Figura 87. Visualización de la ruta.  
Elaboración: (Ordóñez & Guaman, 2015)

- **Solicitudes en Elaboración**

Cuando no se guarda una solicitud queda temporalmente almacenada para su posterior envío a la aprobación.

Nueva Solicitud Solicitudes en Elaboración Estado Solicitudes

Permite editar la solicitud

Permite borrar la solicitud

Permite guardar y enviar aprobar la solicitud

FECHA SOLICITUD	MOTIVO	DESTINO	FECHA SALIDA	ACCIONES
31/05/2015	MOTIVO DE PRUEBA	CATAMAYO	2015-05-31	[editar] [borrar] [guardar]

Figura 88. Interfaz de Administración de las solicitudes.  
Elaboración: (Ordóñez & Guaman, 2015)

- **Estado de Solicitudes**

## 6.2. Asignación de Vehículo

En esta sección llegan las solicitudes guardadas y enviadas a aprobación se chequea los datos y se procede a aprobar o negar.

Asignar Vehículo						
Solicitante	Fecha solicitud	Motivo	Destino	Fecha Salida	Fecha Llegada	Asignar Vehículo
No existen solicitudes para su Aprobación						

Figura 89. Asignación de Vehículo  
Elaboración: (Ordóñez & Guaman, 2015)

## 6.3. Orden de Combustible

Una vez que se apruebe la solicitud llega para ser despachada dicha solicitud con el combustible en base al kilometraje previamente guardado en la asignación de rutas.

Permite asignar el combustible para cada solicitud aprobada.

Asignar Combustible						
Motivo/Actividad	Destino/Proyecto	Fecha Salida	Fecha Llegada	Chofer/Operador	Vehículo/Maquinaria	Asignar Combustible
MOTIVO DE PRUEBA DE PASAJEROS	CATAMAYO	2015-05-31	2015-05-31	BLANCO MARINO SILVA BRAVO	TOYOTA #49	[Asignar Combustible]
MOTIVO DE PRUEBA	CALVAS	2015-05-31	2015-05-31	HIDALGO DE JESUS PINEDA	CHEVROLET #2	[Asignar Combustible]

Figura 90. Asignación de Orden de Combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

Cuando se hace clic en **“Asignar Combustible”** se visualiza el formulario de asignación en que se encuentra el kilometraje de la ruta y los galones asignados para dicha ruta.

CONTROL COMBUSTIBLE

Fecha:  Partida de Combustible:

Vale por:  Galones de DIESEL Gasolinera:

Con cargo a la cuenta de:

Ruta:

Maquinaria/Vehículo:  Kilometraje Recorrido:

Km salida:  Km retorno:

Hora salida:  Hora retorno:

Kilometraje de la ruta calculado.

Figura 91. Formulario para la asignación de Combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

#### 6.4. Despacho de Combustible

Permite visualizar la cantidad a despachar calculada del vehiculo que fue solicitado y adicionalmente se puede asignar galones provisionales.

Despachar Combustible				
FECHA	OPERADOR	MAQUINARIO N°.	CANTIDAD ASIGNADA	DESPACHAR
2015-05-31	NELSON ALBERTO LAPO CHAUNAY	MAZDA 75	10 galones	Asignacion
2015-05-31	ANGEL MARCO QUEZADA PONCE	CHEVROLET 21	5 galones	Asignacion
2		T 21	1 galones	Asignacion
2		T 21	1 galones	Asignacion
2015-05-30	EDINSUN JAVIER FERNANDEZ ESPINOZA	CHEVROLET 21	1 galones	Asignacion

Visualiza los galones de combustibles calculados para la ruta requerida.

Permite Asignar galones de combustible que fueron despachados.

Figura 92. Interfaz de Despacho de combustible.  
Elaboración: (Ordóñez & Guaman, 2015)

### 7. Módulo de Talleres

Al hacer clic sobre el nombre “Talleres” se despliega el submenú que se detalla a continuación:

- Solicitud de Mantenimiento.
- Historial de Mantenimiento.
- Plan de Mantenimiento Preventivo.

#### 7.1. Solicitud de Mantenimiento

Se realiza la solicitud de mantenimiento para un vehiculo buscandolo por la placa y se puede escoger mantenimiento o reparación, en caso de mantenimiento se analiza el kilometraje de la ultima movilizaciòn y de mantenimiento para revisar en plan programado de mantenimiento preventivo que va en una escala de kilometraje y características especiales de los vehículos.

Permite buscar mediante la placa de los vehículos y agregar los datos del vehículo para realizar el mantenimiento o reparación

Opción para escoger un mantenimiento o reparación

Permite agregar la cantidad y la descripción del repuesto que necesita para el mantenimiento o reparación del vehículo.

Figura 93. Formulario para una Solicitud de Mantenimiento.  
Elaboración: (Ordóñez & Guaman, 2015)

### 7.2. Historial de Mantenimiento

Mediante el ingreso de la placa vehicular se puede visualizar los mantenimientos de los vehículos realizados. Y el kilometraje del vehículo en el cuál se realizara el proximo mantenimiento.

Búsqueda del vehículo por la placa.

Fecha	Detalle	Motivo	Kilometraje	Proxim
2015-06-27	Cambio de aceite motor filtro	REPARACION	450000000	45000
2015-05-25	Cambio de aceite motor filtro	REPARACION	199800	--
2015-05-25	Cambio de aceite motor filtro	REPARACION	187000	--
2015-05-25	Cambio de aceite motor filtro	REPARACION	25000	--
2015-05-25	Cambio de aceite motor filtro	REPARACION	179000	--
2015-04-21	Cambio filtro de aire	REPARACION	10000	--
2015-04-20	Cambio de aceite motor filtro	REPARACION	5000	--

Figura 94. Interfaz para el mostrar el Historial de Mantenimiento.  
Elaboración: (Ordóñez & Guaman, 2015)

### 7.3. Plan de Mantenimiento Preventivo

Permite crear, actualizar y borrar los mantenimientos programados en base al kilometraje y características especiales de los vehículos.

PLAN DE MANTENIMIENTO PREVENTIVO			
Kilometraje	Descripción	Gasolina	Centros
20000	Limpieza Cuerpo Aceleración	Todos	<input checked="" type="checkbox"/>
10000	Cambio filtro de aire	Todos	<input checked="" type="checkbox"/>
10000	Limpieza y calibración de frenos	Todos	<input checked="" type="checkbox"/>
60000	Cambio de Banda de accesorios	Todos	<input checked="" type="checkbox"/>
60000	Cambio Banda de Distribución	Todos	<input checked="" type="checkbox"/>

Añade nuevos mantenimientos preventivos.

Edita los datos.   
 Borra los datos.

Figura 955. Interfaz para Administración de Plan de mantenimiento Preventivo.  
 Elaboración: (Ordóñez & Guaman, 2015)

Al hacer clic en “**Añadir**” se visualiza la ventana en la cual se registra el nuevo mantenimiento preventivo.

Permite escoger las características y el kilometraje de los vehículos para insertar el nuevo mantenimiento preventivo.

Agregar Nuevo Mantenimiento

Figura 96. Interfaz de administración para el plan de mantenimiento.  
 Elaboración: (Ordóñez & Guaman, 2015)

## ANEXO I: ACTAS DE ENTREGA

Loja, 27 de Octubre del 2015

### A PETICIÓN VERBAL

Ing. Pablo Vallejo

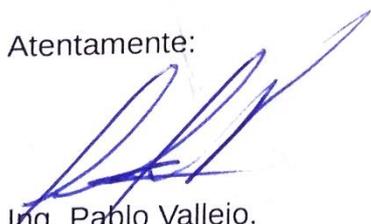
**COORDINADOR DE INFRAESTRUCTURA TECNOLÓGICA DEL GPL**

### CERTIFICA:

Haber autorizado a la Srta. Andrea Silvana Guamán López con cédula 1105603862 y al Sr. Yonder Leonel Ordoñez Ramón con cédula 1105168767, estudiantes de la titulación de Sistemas Informáticos y Computación perteneciente a la Universidad Técnica Particular de Loja, hacer uso de la información del Departamento de Gestión Vehicular para que desarrolle su trabajo de titulación denominado "Diseño e implementación del Sistema de Gestión Vehicular", el mismo que ha sido aprobado y validado.

Es todo cuanto puedo certificar en honor a la verdad y autorizo hacer uso de la presente en lo que estime conveniente.

Atentamente:



Ing. Pablo Vallejo.

**COORDINADOR DE INFRAESTRUCTURA TECNOLÓGICA  
PREFECTURA DE LOJA**



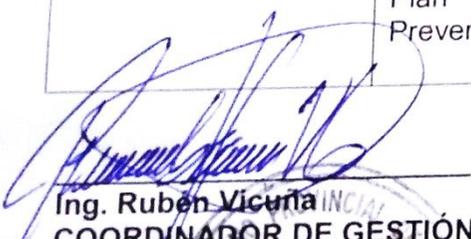
## ANEXO I: VALIDACIÓN DEL SISTEMA.



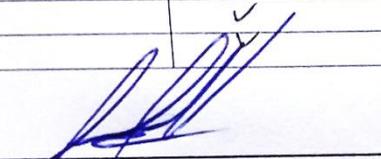
### VALIDACIÓN DEL SISTEMA GESTION VEHICULAR

Modulo/Menú	Submenú	Actividades	Aceptación	Observación
Logeo del Sistema		Ingreso al Sistema Web	✓	
<u>Administración</u>	Usuarios	Añadir	✓	
		Actualizar	✓	
		Eliminar	✓	
		Actividades Usuarios (Log)	✓	
	Asignación vehículo/chofer	Asignar	✓	
		Actualizar	✓	
		Eliminar Asignación	✓	
	Tipo Licencia	Añadir	✓	
		Actualizar	✓	
		Eliminar	✓	
	Combustibles	Añadir	✓	
		Actualizar	✓	
		Eliminar	✓	
	Gasolinera	Añadir	✓	
		Actualizar	✓	
		Eliminar	✓	
	Registrar Vehículo	Reporte Excel	✓	
		Añadir	✓	
		Actualizar	✓	
		Eliminar	✓	
Registrar Conductor	Reporte Excel	✓		
	Añadir	✓		
	Actualizar	✓		
	Eliminar	✓		
<u>Reportes</u>	Cohtrol de Mantenimiento	Búsqueda Avanzada	✓	
		Generar Reporte	✓	
		Reporte Excel	✓	
		Búsqueda Avanzada	✓	

	Provisión de Combustibles	Generar Reporte	✓	No se generó excel.	
		Reporte Excel	✓		
	Matricula y SOAT	Alertas	✓		
		Reporte Excel	✓		
	Ordenes de Movilización	Búsqueda Avanzada	✓		
		Generar Reporte	✓		
		Reporte Excel	✓		
	Novedades	Reporte Excel	✓		
	<u>Gestión Movilización</u>	Solicitud de Movilización	Búsqueda de datos		✓
			Ruta		✓
Solicitudes en Elaboración			✓		
Estado de Solicitudes			✓		
Orden de Combustible		Asignar Combustible	✓		
Despacho Combustible		Despacho de Combustibles	✓		
Asignar Vehículo		Asignar Vehículo	✓		
		Asignar otro conductor	✓		
<u>Talleres</u>		Solicitud de Mantenimiento	Búsqueda de datos	✓	
			Mantenimientos Preventivos	✓	
	Reparaciones		✓		
	Asignación de Repuestos/Lubricantes		✓		
	Historial Mantenimiento	Próximo Mantenimiento	✓		
		Consulta de datos	✓		
	Plan de Mantenimiento Preventivo	Añadir	✓		
		Actualizar	✓		
		Eliminar	✓		

  
**Ing. Rubén Vicuña**  
**COORDINADOR DE GESTIÓN VEHICULAR**  
**PREFECTURA DE LOJA**



  
**Ing. Pablo Ramiro Vallejo**  
**COORDINADOR DE INFRAESTRUCTURA TECNOLÓGICA**  
**PREFECTURA DE LOJA**

