



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

**“Trabajo de grado previo a la obtención del Título de Ingeniero en
Computación”**

**IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN VEHICULAR
UTILIZANDO HERRAMIENTAS “OPEN SOURCE”, PARA LA
ESCUELA DE CONDUCCIÓN DE LA UNIVERSIDAD NACIONAL DE
CHIMBORAZO.**

Autor: Hugo Hernán Flores Dávila

Director: Ing. Alfredo Colcha

Riobamba – Ecuador

2015

Los miembros del Tribunal de Graduación del proyecto de investigación de título: IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN VEHICULAR UTILIZANDO HERRAMIENTAS “OPEN SOURCE”, PARA LA ESCUELA DE CONDUCCIÓN DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO, presentado por el Sr: Hugo Hernán Flores Dávila y dirigido por el Sr. Ing. Alfredo Colcha docente de la Universidad Nacional de Chimborazo.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Jorge Delgado

Presidente del Tribunal

Ing. Jorge Delgado

Ing. Alfredo Colcha

Firma

Ing. Alfredo Colcha

Tutor del Proyecto

Ing. Alfredo Colcha

Ing. Aníbal Llanga

Firma

Ing. Aníbal Llanga

Miembro del Tribunal

Ing. Aníbal Llanga

Ing. Aníbal Llanga

Firma

AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Hugo Hernán Flores Dávila autor y al Ing. Alfredo Colcha Director del Proyecto; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

Hugo Hernán Flores Dávila.

C.I # 060247550-1

A handwritten signature in blue ink that reads "Hugo Flores Dávila." The signature is written in a cursive style with a long, sweeping underline.

AGRADECIMIENTO

Doy gracias infinitas primeramente a Dios por darme la inteligencia y constancia para terminar mi carrera de Ingeniero en Computación.

También agradezco la confianza y apoyo de mi Familia y demás personas que han contribuido positivamente para llevar a cabo esta difícil jornada de mi vida.

A la Universidad Nacional de Chimborazo; a la Facultad de Ingeniería, a la Carrera de Ingeniería en Computación, a la Unidad de Formación Académica y Profesionalización, en donde nuestros profesores supieron transmitirnos generosamente sus conocimientos como profesionales.

Al Ing. Alfredo Colcha, Tutor, quien ha sabido compartir sus conocimientos desde el inicio de mi carrera en la UNACH.

A la Ing. Diana Erazo Directora de la Escuela de Conducción de la Universidad Nacional de Chimborazo y a todo personal de la misma; por darme la oportunidad de realizar el proyecto de investigación en esta gloriosa institución.

HUGO HERNÁN FLORES DÁVILA

DEDICATORIA

Durante el transcurso de mi vida estudiantil ha sido una constante lucha para conseguir cumplir mis objetivos y de esta manera alcanzar uno de mis más grandes sueños, culminar mi carrera, por eso dedico este triunfo a todas las personas que me llenaron de amor esperanza y apoyo en todo momento.

A Dios por brindarme la fortaleza, la inteligencia y constancia permanente para seguir siempre adelante superando todos los obstáculos que se presentaron en mi camino.

A mi mami María quien siempre ha sido y será el centro de mi vida, ella me enseñó a superarme desde niño con su infinito amor, paciencia y apoyo incondicional, gracias MAMI por creer siempre en mí; a mi hija Pamelita, mi pedacito de cielo; motivo principal de mi constante lucha; a mi esposa Silvana, que siempre con su paciencia y amor estuvo conmigo apoyándome; a mis amados hermanos Gladys y Jaime, por toda la ayuda brindada, para quienes siento que nunca los voy a defraudar.

Y a mi padre, esperando que desde el cielo se sienta orgulloso de mí.

HUGO HERNÁN FLORES DÁVILA

INDICE GENERAL

PORTADA.....	i
INDICE GENERAL.....	vi
INDICE DE FIGURAS.....	xii
INDICE DE TABLAS	xiii
RESUMEN.....	xiv
SUMMARY	xv
INTRODUCCIÓN	1
CAPITULO I.....	3
1. MARCO REFERENCIAL.....	3
1.1. Planteamiento del problema	3
1.2. Formulación del problema.....	5
1.3. Objetivos	5
1.3.1. Objetivo General.....	5
1.3.2. Objetivos Específicos	5
1.4. Justificación.....	5
CAPITULO II	8
2. FUNDAMENTACIÓN TEÓRICA.....	8
2.1. Antecedentes del tema.....	8
2.2. Sistema de Gestión Información y Base De Datos.....	9
2.2.1. Sistema.....	9
2.2.2 Tipos de Sistemas	9
2.2.3. Necesidad del Análisis y Diseño de Sistemas	10
2.2.4. El Ciclo de Vida del Desarrollo de Sistemas.....	10
2.2.5. Sistema de Gestión.....	11
2.2.6. Sistema de base de datos.....	12
2.3. Gestión de Procesos.....	12
2.3.1. Gestión	12
2.3.2. Qué es Gestión de Procesos	12
2.3.3. Concepto de gestión en una organización / empresa	13
2.3.4. La gestión de la información en sistemas informáticos.....	13
2.3.5. La gestión de la información en aplicaciones web	13
2.4. Proceso	13

2.4.1. Proceso en una organización / empresa	14
2.4.2. Procesos en sistemas informáticos.....	14
2.4.3. Los procesos en aplicaciones web	15
2.4.4. Procesos de Gestión Administrativa	15
2.4.5. Procesos de Gestión Vehicular de la Escuela de Conducción de la UNACH	16
2.5. Gestión del Mantenimiento de la Flota Vehicular.....	18
2.5.1. Gestión Vehicular	18
2.5.2. Análisis de la Situación Actual de la Gestión de la Flota Vehicular ..	19
2.5.3. Objetivos del Plan.....	19
2.5.4. Planificación y Ejecución	19
2.5.5. Mantenimiento	20
2.5.6. Métodos de Mantenimiento	20
2.6.1. Funciones del DBMS.....	23
2.7. Herramientas Open Source.....	25
2.7.1. Open Source.....	25
2.7.2. Requisitos de Open Source	25
2.7.3. Características de Open Source	26
2.7.4. El Open Source como Producto de Distribución Libre	26
2.7.5. Acerca de las Licencias Open Source.....	27
2.7.6. Importancia de Open Source.....	28
2.7.7. Ventajas, Inconvenientes y Perspectivas del Futuro de Open Source .	28
2.7.8. Perspectivas de futuro	29
2.7.9. Software Libre	30
2.8. Herramientas Open Source Para Desarrollo De Sistemas.....	31
2.8.1. Framework de Desarrollo Web.....	31
2.8.2. Framework	31
2.8.3. Framework Codeigniter	31
2.8.4. Framework Zend.....	35
2.9. PHP.....	39
2.9.1. Ventajas de Php con Otros Lenguajes de Programación	40
2.9.2. Características únicas de Php.....	41

2.10. Dreamweaver	41
2.11. Xampp.....	42
2.12. Servidor Web.....	43
2.13. Servidor Apache	44
2.14. MySQL.....	44
2.14.1. Principales Características de MySQL	45
2.14.2. Por qué el uso de MySQL.....	45
2.14.3. Ventajas de MySQL.....	46
CAPITULO III.....	47
3. METODOLOGÍA	47
3.1. Tipo de Estudio	47
3.1.1. Por el propósito o finalidades perseguidas	47
3.1.2. Según la clase de medios utilizados para obtener los datos.....	47
3.1.3. Por el nivel de conocimientos que se adquieren.	48
3.1.4. Por el campo de conocimiento en que se realiza.	48
3.1.5. Conforme al razonamiento empleado.	48
3.1.6. Conforme al número de investigadores.	48
3.2. Métodos	48
3.3. Población y Muestra.....	49
3.3.1. Población	49
3.3.2. Muestra	50
3.3. Operacionalización de las Variables	50
3.3.1.- Variable independiente.	50
3.3.2.-Variable dependiente.	50
3.4. Procedimientos	51
3.4.1. Fuentes de información.....	51
3.5. Procesamiento y Análisis	51
3.5.1. Encuesta dirigida a los Instructores de la Escuela de Conducción	51
3.5.2. Análisis de los resultados de la entrevista realizada al director y secretaria de la Escuela de Conducción de la Universidad Nacional de Chimborazo.....	61
CAPITULO IV.....	63
4. PROCESO DE INGENIERIA DE SOFTWARE	63

4.1. Ingeniería de la información.....	63
4.1.1. Definición del Ámbito del Problema	63
4.1.2. Investigar la organización.....	63
4.1.3. Definición de Escenarios Problema.....	64
4.1.4. Definición de alternativas de solución.....	65
4.2. Estimaciones.....	65
4.3. Planificación	65
4.4. Análisis de Riesgos	66
4.4.1. Identificar los Riesgos	66
4.4.2. Gestión y Supervisión del Riesgo.....	67
4.5. Factibilidad.....	69
4.5.1. Factibilidad Operativa.....	69
4.5.2. Factibilidad Técnica.....	70
□ Recursos Humanos	70
□ Recursos Hardware	70
□ Recursos Software.....	70
4.5.3. Factibilidad Económica	71
□ Análisis Costo – Beneficio.....	71
□ Determinación de Beneficios	71
- Beneficios Tangibles	71
- Beneficios Intangibles	71
4.6. Fase de Análisis.....	71
4.6.1. Jerarquía de usuarios.....	71
4.7. DISEÑO E IMPLEMENTACION DE LA APLICACIÓN.....	72
4.7.1. Definición de la Metodología	72
4.8. Descripción General del Sistema	72
4.9. Definición del Prototipo	73
4.10. Definición de Requisitos Generales del Sistema.....	73
4.10.1. Requisitos funcionales relacionados con Vehículos.....	73
4.10.2. Requisitos funcionales relacionados con Mantenimiento Preventivo	73
4.10.3. Requisitos funcionales relacionados con Mantenimiento Correctivo	74
4.10.4. Requisitos funcionales relacionados con Taller.....	74

4.10.5. Requisitos funcionales relacionados con Instructores	74
4.10.6. Requisitos funcionales relacionados con Accesorios	75
4.10.7. Requisitos funcionales relacionados con Accidentes	75
4.11. Requisitos de rendimiento	76
4.12. Atributos del sistema software	76
4.12.1. Fiabilidad	76
4.12.2. Disponibilidad.....	76
4.12.3. Seguridad	76
4.12.4. Portabilidad.....	76
4.13. Requisitos no funcionales (Normas y Estándares)	77
4.14. Requisitos no funcionales – Seguridad	77
4.15. Requisitos no funcionales – Organización	77
4.16. Catálogo de Usuarios.....	77
4.17. Modelo de Negocio	77
4.18. Objetivos y Alcance del Sistema.....	77
4.19. Establecimiento de requisitos	78
4.19.1. Especificación de Casos de Uso	78
4.20. Casos de Uso	78
4.20.1. Detalle de los casos de uso	79
4.21. Modelo Conceptual.....	92
4.22. Secuencia de Actividades entre actores	92
4.23. Definición de la Interfaz	93
4.24. Diseño arquitectónico	96
4.25. Diseño de la base de datos.....	97
4.26. Diseño de procedimientos	98
4.26.1. Clase Sistema.....	98
4.26.2. Crear Mantenimiento Correctivo	100
4.26.3. Eliminar Mantenimiento Correctivo.....	108
4.26.4. Modificar Mantenimiento Correctivo	110
4.26.5. Buscar Mantenimiento Correctivo.....	116
4.27. Pruebas de Software	124
4.28. Resultados de la aplicación del software.....	125

CAPITULO V	127
5. CONCLUSIONES Y RECOMENDACIONES.....	127
5.1. Conclusiones	127
5.2. Recomendaciones	128
5.3. BIBLIOGRAFÍA.....	129
5.3.1. WEBGRAFIA.....	131
5.4. ANEXOS.....	132
5.4.1. Control de Mantenimiento de vehículos.....	132
5.4.2. Reporte de carga de combustible	133
5.4.3. Reporte de consumo de combustible	134
5.4.4. Certificado de la implementación del Sistema de Gestión Vehicular en la Escuela de Conducción de la Universidad Nacional de Chimborazo.....	135

INDICE DE FIGURAS

Figura 1.	Etapas de un plan de gestión de mantenimiento.	19
Figura 2.	Métodos de mantenimiento.	22
Figura 3.	Mantenimiento preventivo	52
Figura 4.	Conocimientos de computación	53
Figura 5.	Implementación del sistema informático	54
Figura 6.	Tipo de mantenimiento.....	55
Figura 7.	Informes de gestión vehicular	56
Figura 8.	Mantenimiento y procesos de gestión vehicular	57
Figura 9.	Implementación de un sistema informático	58
Figura 10.	Importancia del sistema informático	59
Figura 11.	Eficiencia del sistema de gestión vehicular.....	60
Figura 12.	Utilización y adaptación al nuevo sistema informático	61
Figura 13.	Diagrama de casos de uso	78
Figura 14.	Clases de la base de datos	92
Figura 15.	Secuencia de actividades entre actores y el sistema.....	93
Figura 16.	Interfaz para validación de usuarios.....	94
Figura 17.	Interfaz con menú principal.....	94
Figura 18.	Interfaz para registrar un nuevo mantenimiento	94
Figura 19.	Interfaz para eliminar un mantenimiento	95
Figura 20.	Interfaz para modificar un mantenimiento	95
Figura 21.	Interfaz para buscar un mantenimiento	95
Figura 22.	Interfaz para generar reporte de un mantenimiento	96
Figura 23.	Arquitectura del sistema.....	96
Figura 24.	Estructura de la base de datos	97

INDICE DE TABLAS

Tabla 1.	Tabla comparativa de Codeigniter y Zend	38
Tabla 2.	Comparativa de frameworks Codeigniter y Zend. Postulante.....	39
Tabla 3.	Nómina de personas que conforman la población para la elaboración del proyecto	49
Tabla 4.	Variable independiente.....	50
Tabla 5.	Variable dependiente.....	50
Tabla 6.	Mantenimiento preventivo	51
Tabla 7.	Conocimientos de computación	52
Tabla 8.	Implementación del sistema informático	53
Tabla 9.	Tipo de mantenimiento.....	54
Tabla 10.	Informes de gestión vehicular	55
Tabla 11.	Mantenimiento y procesos de gestión vehicular	56
Tabla 12.	Implementación de un sistema informático	57
Tabla 13.	Importancia del sistema informático	58
Tabla 14.	Eficiencia del sistema de gestión vehicular.....	59
Tabla 15.	Utilización y adaptación al nuevo sistema informático	60
Tabla 16.	Planificación en tiempo del sistema.....	66
Tabla 17.	Medición manipulación de información referente a mantenimiento vehicular manual y mediante el sistema.....	125

RESUMEN

Este proyecto de investigación, estudia el Software Libre, las herramientas Open Source y el Framework Codeigniter. Estudio que se verá plasmado en la creación e implementación de un sistema informático de Gestión vehicular para la Escuela de Conducción de la Universidad Nacional de Chimborazo , ubicada en el cantón Riobamba provincia de Chimborazo.

El Software Libre en la actualidad está ganando espacio en el campo informático, por todas sus ventajas y características de uso, las empresas necesitan gestionar de mejor forma sus procesos en base a sistemas informáticos y el Software Libre y las Herramientas Open Source evitan el pago de licencias y proporcionan la ampliación o actualización de sistemas a bajos costos económicos.

En la implementación del sistema y una vez estudiadas las herramientas necesarias se escogió el Framework Codeigniter como el entorno de desarrollo de la aplicación, Dreamweaver para el desarrollo de la página web, XAMPP como servidor independiente de plataforma, PHP como lenguaje de programación, MySQL como gestor de base de datos y el servidor web de código abierto Apache.

Con la implementación de este software se solucionarán problemas de almacenamiento de información y se automatizarán todos los procesos referentes al tema.

El estudio está enmarcado en la investigación cuantitativa, se aplicó el método inductivo – deductivo – analítico. La población utilizada para la elaboración del mismo fueron los empleados de la Escuela de Conducción.

En conclusión se llegó a la elaboración de un software de calidad, utilizando las herramientas Open Source, además se cumplió con los objetivos propuestos en este proyecto de investigación y se satisfizo las necesidades de los usuarios del sistema propuesto.



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERIA
CENTRO DE IDIOMAS



Ms. Rosita Fernández

Fecha: 27 de Julio de 2015

SUMMARY

This research project studies the Free Software and the Open Source Framework Codeigniter tools. This study will be reflected in the creation and implementation of a computerized management system for vehicular Driving School of the National University of Chimborazo, located in Riobamba city Chimborazo province.

Free Software is currently gaining ground in the computer field, for all its advantages and characteristics of using, companies need to better manage their processes and systems based on Free Software and Open Source Tools, and they avoid paying licenses and provide expansion or upgrading systems at low economic costs.

For implementing the system and once studied the tools, we chose the Framework Codeigniter as the development environment of the application, Dreamweaver for website development, XAMPP platform as a standalone server, PHP programming language, MySQL as database manager and open source web server Apache.

With the implementation of this software storage, some problems will be solved and all relevant processes to the topic will be automated.

The study is framed in quantitative research, it was applied the inductive method - deductive - analytical. The population used for its preparation was employees of the Driving School.

In conclusion we made quality software using Open Source tools, and we have reached the goals in this research project and we have satisfied the needs of users of the proposed system.

Rosita Fernández



INTRODUCCIÓN

La Informática actualmente es una ciencia que está inmersa en todos los aspectos cotidianos de la vida. En las empresas ayuda de manera significativa a mejorar notablemente el desempeño y eficiencia de las mismas.

En toda empresa deben cuidarse todos los elementos y recursos principales que son el soporte de la misma, a todos estos recursos se los debe cuidar para alargar su vida útil, evitando así su temprano deterioro.

Todos estos procesos se los conoce en la actualidad como Gestión de Mantenimiento. El presente proyecto logrará automatizar todos los procesos de Gestión Vehicular que hasta la fecha se los hace manualmente.

La Escuela de Conducción de la UNACH, no posee un Sistema Informático que logre llevar de manera adecuada todos los procesos de gestión vehicular es por eso que se ha visto la necesidad de implementar el Sistema de Gestión Vehicular con la finalidad de tener un control de todo lo referente a los vehículos

El conocer detalladamente la flota vehicular proporciona una manera lógica y certera de dar soluciones a diferentes tipos de problemas. Con un plan definido se lograra evitar casos fortuitos de mantenimiento emergente y tendremos en su lugar un mantenimiento preventivo correcto.

Todos estos parámetros junto a la tecnología informática acompañada del uso de Herramientas Open Source dará como resultado la Implementación de un Sistema de Gestión Vehicular para la Escuela de Conducción de la UNACH.

El presente trabajo investigativo consta de los siguientes capítulos:

El Capítulo I el Marco Referencial describe, la problemática que existe actualmente en la Escuela de Conducción de la Universidad Nacional de

Chimborazo para la gestión del mantenimiento vehicular, la delimitación, objetivos y justificación del proyecto de investigación.

En el Capítulo II se hace referencia al Marco Teórico, se detalla las herramientas de software libre, servidores web, motor de base de datos, lenguaje de programación y frameworks que sirven de plataforma para el desarrollo de sistemas informáticos.

En el Capítulo III se define los métodos, mecanismos, estrategias o procedimientos a seguirse en la investigación.

En el Capítulo IV se realiza el proceso de ingeniería de la información con un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del sistema informático. Se realizó el Diseño e Implementación de la aplicación donde se definieron técnicas y características incluyendo las ventajas y desventajas para tener una idea amplia del software que se va a desarrollar. Se analiza los resultados de los beneficios del sistema informático automatizado en relación con la forma que se viene realizando de forma manual dichos procesos.

En el Capítulo V finalizando con las conclusiones, recomendaciones y anexos.

CAPITULO I

1. MARCO REFERENCIAL

1.1. Planteamiento del problema

La Escuela de Conducción de la UNACH, inicia sus actividades académicas el 21 de Noviembre de 2011, su fundador es el Dr. Jesús Estrada docente de la Institución. La Escuela es la encargada de formar nuevos profesionales de la conducción, la misma es parte de la Facultad de Ciencias Humanas y Tecnologías de la Universidad Nacional de Chimborazo UNACH.

El personal docente de la Escuela de Conducción de la UNACH, son profesionales de tercer nivel al igual que los Instructores de Conducción, garantizando de esta manera una enseñanza de clase.

La Institución cuenta con todos los requerimientos físicos, humanos y tecnológicos para actualmente ser catalogada como una Institución pionera en lo referente a Escuelas de Conducción.

La Institución mencionada ha ido creciendo paulatinamente estando actualmente avalada por la Escuela Superior Politécnica del Litoral en categoría clase “ A “ en lo referente a Escuelas, además se encuentra otorgado el permiso para la obtención de la licencia profesional tipo “D”. Al momento se cuenta con 10 vehículos marca Chevrolet Aveo 1.5, año 2011 Estándar y una Buseta marca Chevrolet NPR para 28 pasajeros.

Desde el inicio, la Escuela de Conducción ha llevado la gestión de mantenimiento de la flota vehicular en forma manual sin hacer uso de la tecnología actual, motivo por el cual se tiene problemas en los siguientes aspectos:

- Mantenimiento y control de funcionamiento de vehículos de Conducción, deficiente.

- Facturas de mantenimientos, gastos y cargas de combustible no disponibles de forma inmediata.
- Mantenimiento preventivo y mantenimiento correctivo inapropiados.
- No existe un historial de reposición de piezas averiadas
- Informes de gastos de combustible en galones y dólares deficiente
- Kilometraje recorrido por período de instrucción desconocido
- Inexistencia de elaboración de informes de gestión vehicular
- No existe un historial certero de gestión vehicular por auto de instrucción
- No existe un historial de cambio o composuras de accesorios de los vehículos de instrucción.
- No existe un historial certero de accidentes de cada unidad vehicular.
- Falta de un sistema informático que optimice el trabajo y control de la flota vehicular
- Descoordinación de todas estas actividades
- Ineficaz control de la mayoría de procesos.

En la actualidad existe una gran cantidad de documentos archivados de esta información, la que está en algún lugar de las oficinas de la Escuela de Conducción, siendo esto un problema al momento de requerirse determinada información al respecto.

En este proyecto se investigarán y analizarán todas las características y funcionalidades de las herramientas open source, se analizará e investigará el uso adecuado de servidores y bases de datos adecuadas de código abierto y el uso de software libre, también se hará la respectiva investigación de todas las demás herramientas informáticas para poner en marcha este proyecto.

Con la Instalación de este Sistema de Gestión Vehicular, se resolverán los problemas antes mencionados que actualmente existen la Escuela de Conducción.

1.2. Formulación del problema

¿De qué manera la implementación de un sistema web basado en tecnología open source incide para un adecuado control y gestión de los procesos de mantenimiento vehicular de la Escuela de Conducción de la Universidad Nacional de Chimborazo?

1.3. Objetivos

1.3.1. Objetivo General

Implementar un Sistema de Gestión Vehicular utilizando Herramientas Open Source para la Escuela de Conducción de la UNACH.

1.3.2. Objetivos Específicos

- Investigar sobre el Framework Open Source adecuado para la gestión de procesos
- Analizar los procesos de gestión y mantenimiento vehicular
- Implementar y dejar en correcto funcionamiento el sistema de gestión vehicular en la escuela de Conducción de la UNACH, utilizando herramientas open source.

1.4. Justificación

El motivo de realizar este proyecto se debe a que el Software Libre y las herramientas Open Source (Código Abierto), son instrumentos informáticos importantes y que están vigencia y constante crecimiento en la actualidad. Este tipo de Software es para todos, lo podemos modificar y utilizar para nuestros propósitos, distribuir, redistribuir y compartir.

Gran cantidad de proyectos actualmente se los realiza con Open Source y Software Libre.

Las herramientas Open Source, nos permiten modificar su código y aplicarlo a las necesidades de cada individuo, empresa o Institución.

Código abierto es la expresión con la que se conoce al software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

La Universidad Nacional de Chimborazo es una institución que está en crecimiento, por tal motivo se creó la Escuela de Conducción de la UNACH. En el aspecto social dicha institución aporta con la formación de nuevos profesionales de la conducción.

Partiendo de este hecho, el implementar esta tecnología en la Escuela de Conducción de la UNACH, hará que la misma optimice sus recursos económicos.

Las implicaciones prácticas que el proyecto conlleva son:

- El valor metodológico y teórico del proyecto, es que servirá como fuente de estudio para personas que les guste el campo informático del Software Libre, estando el proyecto a disposición de cualquier individuo que lo quiera mejorar.
- Se tendrá a los vehículos en correcto funcionamiento
- Proporcionar a alumnos e instructores de conducción seguridad al momento de conducir ya que se tendrá a los vehículos bajo estricto control de operatividad.
- Se tendrá cualquier tipo de información en referencia al tema instantáneamente y de forma clara y precisa.

Además en base a este proyecto se pueden desarrollar otras investigaciones para mejorar el buen estado de funcionamiento de todos los vehículos de la Universidad Nacional de Chimborazo y de otras instituciones.

Para el personal administrativo la implementación del Sistema es importante, ya que se tendrá la información clara, oportuna y verídica en cualquier momento, además de dar un seguimiento correcto en cuanto a gestión vehicular; lo cual mejorará la forma de llevar este control a dicho personal, siendo este el valor metodológico del proyecto.

En la actualidad el uso de Software Libre y Herramientas Open Source es un ámbito de estudio que está en vigencia y ganando espacio en relación al software convencional.

CAPITULO II

2. FUNDAMENTACIÓN TEÓRICA

2.1. Antecedentes del tema

Según la investigación realizada existen varias propuestas de Sistemas de Gestión Vehicular a nivel nacional e internacional, el propósito es el mismo; conseguir automatizar y mejorar el nivel de funcionamiento de vehículos a través de la tecnología actual, cada proyecto es diferente.

En la investigación realizada de proyectos similares se concluye en todos estos, que las propuestas sugeridas deben ser implementadas, ya que actualmente son necesarias e imprescindibles.

Además se concluye que cada proyecto deberá ajustarse a sus necesidades y requerimientos, estos son una referencia susceptible de modificaciones y mejoras. Todos los proyectos realizados son sustentables.

La tecnología y la Computación como ciencia en la actualidad avanza día a día es por eso que será utilizada en este proyecto, para la elaboración del nuevo Sistema. El mantenimiento adecuado de vehículos, utilizando un Sistema Informático a base de Software Libre y herramientas de Código Abierto es de gran ayuda para la administración de la Escuela de Conducción de la UNACH ya que se proporcionara el Sistema adecuado para el efecto.

La viabilidad de estos proyectos ya sea en el aspecto económico, administrativo y social es aceptable. La sustentación teórica y práctica de proyectos similares proporciona grandes expectativas para sustentar o implementar el Sistema de Gestión Vehicular en la Escuela de Conducción dela UNACH.

2.2. Sistema de Gestión Información y Base De Datos

2.2.1. Sistema

Un sistema es un: “Conjunto de elementos relacionados entre sí funcionalmente, de modo que cada elemento del sistema es función de algún otro elemento, no habiendo ningún elemento aislado”¹.

2.2.2 Tipos de Sistemas

Los sistemas de gestión de información según (Kendall & Kendall, J, 2011) plantea la siguiente clasificación:

2.2.2.1. Sistemas de procesamiento de transacciones (TPS).- Son sistemas de información computarizados que se desarrollaron para procesar grandes cantidades de información para las transacciones de negocios rutinarias, como nóminas e inventario.

2.2.2.2. Sistemas de información administrativa (MIS).- No sustituyen a los sistemas de procesamiento de transacciones; más bien, todos los MIS incluyen el procesamiento de transacciones. Los MIS son sistemas de información computarizados que funcionan debido a la decidida interacción entre las personas y las computadoras.

2.2.2.3. Sistemas de soporte de decisiones (DDS, o sistemas de apoyo a la toma de decisiones).- Pertenecen a una clase de sistema superior computarizados. Los sistemas DDS son similares al sistema de información administrativa tradicional debido a que ambos dependen de una base de datos como fuente de datos. La diferencia estriba en que el sistema de soporte de decisiones está más enfocado a brindar respaldo a la toma de decisiones en todas sus fases, aunque la decisión misma aún corresponde de manera exclusiva al usuario.

2.2.2.4. El software de código de fuente abierto (OOS).- Es una alternativa al desarrollo de software tradicional, en donde el código propietario no se oculta a

¹ Ferrater, J. (1979). *Sistema Diccionario de filosofía*. Recuperado el 7 de Julio de 2014, de <http://www.filosofia.org/enc/fer/sistema.htm>

los usuarios. Con el OOS, los usuarios y programadores pueden estudiar, compartir y modificar el código propietario o las instrucciones de computadora. p. 2-3-5.

2.2.3. Necesidad del Análisis y Diseño de Sistemas

En la obra fundamentos del análisis de sistemas (Kendall & Kendall, J, 2011) dice que: “El análisis y diseño de sistemas se utiliza para analizar, diseñar e implementar las mejoras en el apoyo para los usuarios y las funciones de negocios que se puedan llevar a cabo mediante el uso de sistemas de información computarizados”. p. 6.

2.2.4. El Ciclo de Vida del Desarrollo de Sistemas

El ciclo de vida del desarrollo de un sistema (SDLC) según (Kendall & Kendall, J, 2011) es: “Una metodología en fases, para el análisis y diseño, de acuerdo con la cual los sistemas se desarrollan mejor al utilizar un ciclo específico de actividades del analista y los usuarios”. p. 8.

2.2.4.1. Clasificación de los ciclos de vida del desarrollo de sistemas

La clasificación de los ciclos de vida del desarrollo de sistemas según (Kendall & Kendall, J, 2011), son:

2.2.4.1.1. Identificación de los problemas oportunidades y objetivos.

“En esta primera fase del ciclo de vida del desarrollo de sistemas, el analista se encarga de identificar correctamente los problemas, las oportunidades y los objetivos. Esta etapa es imprescindible para el éxito del resto del proyecto.” p. 9.

2.2.4.1.2. Determinación de los requerimientos de información del factor humano.

“Determinar las necesidades de los usuarios involucrados, mediante el uso de varias herramientas, para comprender la forma en que interactúan en el contexto laboral con sus sistemas de información actuales. El analista utilizara métodos interactivos como entrevistas, muestreos e investigación de datos duros.” p. 10.

2.2.4.1.3. Análisis de las necesidades del sistema.

“A partir de los diagramas de flujo de datos, de secuencia u otros tipos de diagramas se debe desarrollar un diccionario de datos para enlistar todos los elementos de datos utilizados en el sistema, así como sus especificaciones”. p. 10.

2.2.4.1.4. Diseño del sistema recomendado.

“El analista de sistemas utiliza la información recolectada para realizar el diseño lógico del sistema de información. El analista diseña los procedimientos para ayudar a que los usuarios introduzcan los datos con precisión, de manera que los datos que entren al sistema de información sean los correctos”. p. 11.

2.2.4.1.5. Desarrollo y documentación del software.

“Se desarrolla junto con los usuarios una documentación efectiva para el software, incluyendo manuales de procedimientos, ayuda en línea, sitios web con preguntas frecuentes (FAQ) y archivos Léame (Read Me) para incluir con el nuevo software”. p. 11.

2.2.4.1.6. Prueba y mantenimiento del sistema.

“Primero se completa una serie de pruebas para señalar los problemas con datos de muestra y después se utilizan datos reales del sistema actual”. p. 11

2.2.4.1.7. Implementación y evaluación del sistema.

“En esta última fase del desarrollo de sistemas se debe implementar el sistema de información. En esta fase hay que capacitar a los usuarios para operar el sistema”. p. 11.

2.2.5. Sistema de Gestión

Un sistema de gestión es un sistema que se desarrolla para establecer políticas y objetivos en una empresa o institución, para mejorar el funcionamiento de la misma y conseguir esos objetivos.

2.2.6. Sistema de base de datos

“El término sistema de base de datos se refiere a una organización de componentes que define y regula la captura, almacenamiento, administración y uso de datos dentro de un ambiente de base de datos. Desde el punto de vista de una administración general, el sistema de base de datos está compuesto de cinco partes principales que son: hardware, software, personas, procedimientos y datos”².

“Es un conjunto de componentes interrelacionados que recolectan, procesan, analizan y distribuyen información para apoyar la toma de decisiones y el control en una organización”.³

2.3. Gestión de Procesos

2.3.1. Gestión

Extraído el 1 de Julio del 2014 disponible en la web, http://es.answers.com/Q/Qu%C3%A9_es_la_gesti%C3%B3n_de_procesos, expresa que Gestión es: “El conjunto de actividades de planeación, control y ejecución que tiene como propósito establecer los elementos de un proceso en una empresa, sea ésta administrativa o técnica.”

2.3.2. Qué es Gestión de Procesos

Según BRAVO, Juan (2008) manifiesta que Gestión de Procesos es: “Una forma sistemática de identificar, comprender y aumentar el valor agregado de los procesos de la empresa para cumplir con la estrategia del negocio y elevar el nivel de satisfacción de los cliente” (p. 22).

² Coronel, C., Morris, S., & Rob, P. (2011). *BASE DE DATOS Diseño, implementación y administración* (Novena ed.). México D.F: CENGAJE Learnig.

³ Laudon, K., & Laudon, J. (2004). *Sistema de Información Gerencial*. (Octava ed.). México: Pearson Educación.

2.3.3. Concepto de gestión en una organización / empresa

Es la situación en la que los encargados de la administración y gestión de los recursos económicos y materiales deben proceder eficientemente para que la situación empresarial se encuentre acorde a los objetivos y metas planteados desde el inicio de las operaciones actuando con prudencia en los posibles riesgos e imprevistos que puedan existir en la institución la misma que podría afectar la solvencia e imagen corporativa.⁴

2.3.4. La gestión de la información en sistemas informáticos

Según (Maigua & López, 2012) manifiestan que: “La gestión de proyectos es la aplicación del enfoque de sistemas para la administración de tareas tecnológicas complejas o de proyectos cuyos objetivos se establecen explícitamente en términos de tiempo, costes y parámetros de realización” pág. 14.

2.3.5. La gestión de la información en aplicaciones web

El objetivo de este proyecto es el diseño y la implementación de una aplicación web, capaz de gestionar la flota vehicular de la Escuela de Conducción de la UNACH, esto se obtendrá gestionando la información de forma correcta y encaminándola hacia la creación de una aplicación web.

2.4. Proceso

Según www.eumed.net/libros-gratis/2011e/1084/, extraído el 1 de Julio del 2014 de <http://www.eumed.net/libros-gratis/2011e/1084/introduccion.html>, manifiesta que Proceso es: “Un como un conjunto de actividades interrelacionadas entre sí que, a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido.

En otras palabras, un proceso es la manera en la que se hacen las cosas en la empresa. Ejemplos de procesos son el de producción y entrega de bienes y/o

⁴ GLADYS, C. d. (1996). *Análisis y Administración Financiera* (Cuarta ed.). Santa Fé de Bogotá, D.C.- Colombia:Edit. Corcas Editores Ltda. p. 3, 4, 5, 6, 10

servicios, el de gestión comercial, el de desarrollo de la visión estrategia, el de desarrollo de producto. Estos procesos deben estar correctamente gestionados empleando distintas herramientas de la gestión de procesos.

2.4.1. Proceso en una organización / empresa

Según BRAVO, Juan (2008) manifiesta que Proceso es:

Una totalidad que cumple un objetivo completo y que agrega valor para el cliente. Esta unidad es un sistema de creación de riqueza que inicia y termina transacciones con los clientes en un determinado período de tiempo .Cada activación del proceso corresponde al procedimiento de una transacción, en forma irreversible, por eso se emplean los conceptos de temporalidad y de “flecha de tiempo”. El período de tiempo es hoy el punto crítico de trabajo para incrementar la productividad. (p. 27)

2.4.2. Procesos en sistemas informáticos.

Los procesos en sistemas informáticos según (Durán, 2007) manifiesta que:

Un proceso es un programa en ejecución. Hay que resaltar que un proceso es más que el código del programa. Además del código, un proceso incluye también la actividad actual, que queda representada por el valor del contador de programa y por los contenidos de los registros del procesador. Generalmente, un proceso incluye también la pila (*stack*) del proceso, que contiene los datos temporales (como los parámetros de las funciones, las direcciones de retorno y las posibles variables locales) y una sesión de datos, que contiene las variables globales. Un proceso puede incluir, además, un montón de memoria (*heap*), que es la memoria que se asigna, de forma dinámica, al proceso en el tiempo de ejecución. (p. 1).

2.4.3. Los procesos en aplicaciones web

Según (Cáceres & Marcos, E), en referencia a los procesos en aplicaciones web manifiestan que:

Según afirma Jacobson et al. (2000) el modelo de proceso más adecuado para el desarrollo de software es un proceso interactivo e incremental, puesto que a diferencia de otros modelos de proceso, como por ejemplo el modelo en cascada, permite la obtención de diversas versiones del producto software antes de la entrega final del mismo y la depuración y validación progresiva del mismo, lo que sin duda redundará en un software más satisfactorio para usuarios y cliente.

2.4.4. Procesos de Gestión Administrativa

“El proceso de gestión administrativa consiste en:

2.4.4.1. Planeación: Establecer anticipadamente los objetivos, políticas, reglas, procedimientos, programas, presupuestos, y estrategias de un organismo social. Es decir consiste en determinar lo que va a hacerse.

2.4.4.2. Organización: La organización agrupa y ordena las actividades necesarias para lograr los objetivos, creando unidades administrativas, asignando funciones, autoridad, responsabilidad y jerarquías; estableciendo además las relaciones de coordinación que entre dichas unidades debe existir para hacer óptima la cooperación humana, en esta etapa se establecen las relaciones jerárquicas, la autoridad, la responsabilidad y la comunicación para coordinar las diferentes funciones.

2.4.4.3. Integración: Consiste en seleccionar y obtener los recursos financieros, materiales, técnicos y humanos considerados necesarios para el adecuado funcionamiento del organismo social.

La integración agrupa la comunicación y la relación armónica de los elementos humanos y materiales, selección, entrenamiento y compensación del personal.

2.4.4.4. Dirección: Es la acción e influencia interpersonal del administrador para lograr que subordinados obtengan los objetivos encomendados, mediante la toma de decisiones, la motivación, la comunicación y coordinación de esfuerzos. La dirección contiene ordenes, relaciones personales jerárquicas y toma de decisiones.

2.4.4.5. Control: establece sistemas para medir resultados y corregir las desviaciones que se presenten, con el fin de asegurar que los objetivos planteados se logren”.⁵

Como investigador se considera que la Gestión de Procesos es el acto de controlar y administrar de manera lógica y adecuada los recursos, para lograr optimizar el trabajo, dando como resultado un mejor nivel de desempeño eficiente y eficaz.

2.4.5. Procesos de Gestión Vehicular de la Escuela de Conducción de la UNACH

En la actualidad todos los procesos de gestión vehicular en la Escuela de Conducción de la UNACH se los realiza manualmente, es decir toda la información acerca de todos estos procesos se envía, recibe y archiva en papel. Motivo por el cual al momento de surgir un requerimiento de información es difícil su búsqueda.

Dichos procesos son los siguientes:

- Carga de combustible en los vehículos de instrucción.- El instructor que desea cargar combustible en el vehículo cargo suyo, pide una factura al director de Escuela, la cual debe tener su firma, fecha, número de unidad y nombre del instructor; el instructor procede a acercarse a la gasolinera ya asignada y fija y realiza la respectiva carga de combustible, en la misma

⁵ Chiavenato, I. (s.f.). *Promonegocios*. Recuperado el 3 de Julio de 2014, de <http://odauts.com/blogsuts/admongeneral/files/2013/10/Proceso-Administrativo.pdf>

factura se detalla cantidad de galones de gasolina, tipo de combustible, cantidad de dinero en dólares y kilometraje, la factura es firmada por el instructor que realiza la carga y la copia de la factura queda en la gasolinera.

Luego se entrega la factura original en secretaría y se la archiva.

- **Mantenimiento preventivo.-** Este mantenimiento se refiere a cambios de aceite y filtros o lavada del automóvil, de igual forma el instructor solicita al Director de Escuela este mantenimiento, el Director llama al taller asignado por la UNACH; el instructor lleva la unidad a este taller y ahí se realiza este trabajo. El propietario del taller envía una proforma con todos los pormenores del trabajo realizado, costos y nombres de cosas, elementos, piezas, etc., utilizadas; el instructor entrega la proforma en secretaría y se la archiva.

Dentro de este tipo de mantenimiento entra el mantenimiento especificado en el manual de usuario de la empresa Chevrolet. Es decir el mantenimiento de los 5000, 10000, 15000, 200000 kilómetros, etc.

- **Mantenimientos correctivos.-** Este mantenimiento se refiere a cambios de partes o piezas que estas averiadas o que han cumplido su vida útil ya sean mecánicas, eléctricas, de carrocería o interiores, de igual forma el instructor solicita al Director de Escuela el tipo de arreglo necesario para el auto, el Director llama al taller acorde a la necesidad de mantenimiento del auto ya asignado por la UNACH; el instructor lleva la unidad a este taller y ahí se realiza el trabajo requerido. El propietario del taller envía una proforma con todos los pormenores del trabajo realizado, costos y nombres de piezas, dispositivos o materiales utilizados, el instructor entrega la proforma en secretaría y se la archiva.
- **Reposición de adhesivos y accesorios del vehículo y pormenores.**

- Informe de kilometraje recorrido por unidad vehicular, al final de cada semestre.- Esta información es requerida a los instructores al final de cada período de clases.
- Informe de cantidad de galones de combustible consumidos por unidad vehicular durante el período de instrucción.
- Informe de cantidad de dinero por el combustible consumido por unidad vehicular durante el período de instrucción.

Todos estos procesos se los lleva a cabo cada semestre de instrucción, obteniéndose gran cantidad de documentos que se los archiva año tras año, obteniéndose dificultades al momento de requerirse un determinado informe.

2.5. Gestión del Mantenimiento de la Flota Vehicular

2.5.1. Gestión Vehicular

La gestión vehicular es promover todos los procesos adecuados para un correcto control de los vehículos.

Para obtener un eficiente control de los procesos de gestión vehicular se deben abarcar todos los aspectos relacionados con el mismo, sólo así se logrará obtener el resultado deseado.

Un buen plan de gestión de mantenimiento vehicular comprende de las siguientes etapas:

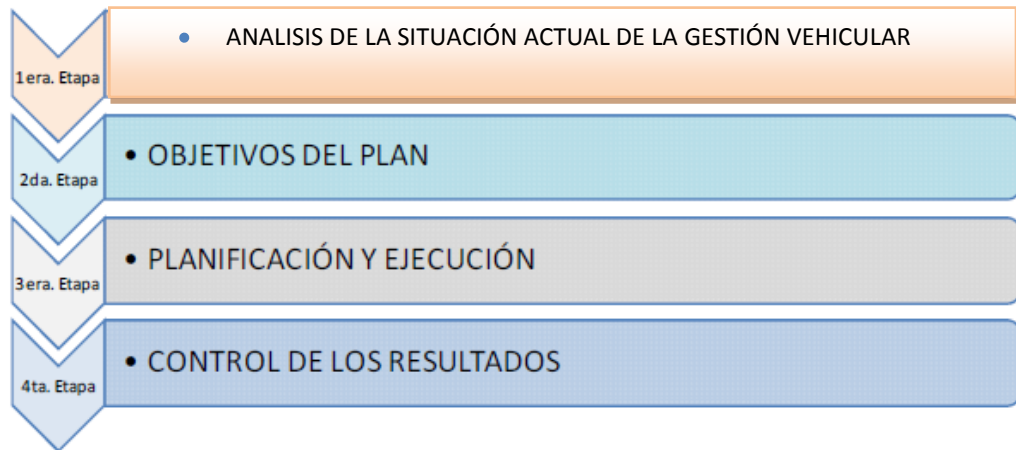


Figura 1. Etapas de un plan de gestión de mantenimiento. Fuente: Navarro, L y otros. Gestión integral de mantenimiento. 1997.

2.5.2. Análisis de la Situación Actual de la Gestión de la Flota Vehicular

Este es el primer paso para evaluar el estado en que se encuentra el departamento que está a cargo de gestionar todo lo referente a los vehículos, así se identificará debilidades y puntos de mejora para alcanzar los objetivos deseados en la implementación del nuevo sistema informático. Los procedimientos utilizados deberán ser cualitativos y cuantitativos, por lo tanto este resultado será la alternativa para futuras mejoras.

2.5.3. Objetivos del Plan

En base al análisis anterior se definen los objetivos del plan de gestión vehicular.

Los objetivos serán concretos y ayudarán a mejorar la administración de la flota vehicular.

2.5.4. Planificación y Ejecución

2.5.4.1. Definición de planificación:

"Es el proceso de establecer objetivos y escoger el medio más apropiado para el logro de los mismos antes de emprender la acción". (Goodstein, 1998).

2.5.4.2. Definición de ejecución:

"Proceso en el cual se ejecuta lo establecido en la planificación partiendo de los resultados obtenidos en la investigación, se pretende alcanzar los objetivos propuestos". (Ander-Egg, 2003).

Como investigador se buscarán los requerimientos para que el nuevo sistema sea útil y funcional y se procederá a fabricarlo.

2.5.5. Mantenimiento

“La Asociación Francesa de Normalización (AFNOR), define el mantenimiento como un conjunto de actividades destinadas a mantener o restablecer un bien a un estado o a unas condiciones dadas de seguridad en el funcionamiento, para cumplir con una función requerida. Estas actividades suponen una combinación de prácticas técnicas, administrativas y de gestión”⁶.

2.5.6. Métodos de Mantenimiento

No existen elementos ideales o duraderos por siempre, todo tiene un tiempo de vida útil es así que cualquier componente sufre degradación, por consiguiente vendrán las averías y anomalías.

“Entendiéndose como condición normal de un equipo, el estado en el cual trabajando dentro de sus especificaciones, puede mantener sus nivel productivo, optimizando los recursos, con la calidad exigida, con la seguridad necesaria, sin pérdidas energéticas y con control de la contaminación ambiental, toda desviación de esta tolerancia es el efecto de una anomalía o avería; si no las evitamos o eliminamos una vez aparecidas, no se mantiene esa condición, reduciendo la capacidad productiva del equipo y por tanto las competencias de la empresa”⁷.

ANOMALÍA – DEFECTO. Suceso en un equipo o cualquiera de sus elementos, que denota, que este experimenta un estado fuera de su condición normal de utilización; pero, que no determina imposibilidad. Es *progresiva*, permite prever su aparición y dar seguimiento a su evolución, ya que está relacionada con el desgaste, desajustes, ruidos, etc.: proporcionado tiempo para evitar averías.⁸

⁶Boucly, F. (1988). *Gestión del mantenimiento*. Madrid-España: Asociación Española de Normalización y Certificación (AENOR), p. 19.

⁷Navarro, L., & otros. (1997). *Gestión integral de mantenimiento*. Barcelona, España: MARCOMBO S.A, p. 5-6.

⁸ Navarro, L., & otros, Op. Cit., p. 11-12; Boucly, F, Op. Cit., p. 22-23; Tavares, L., *Administración moderna del mantenimiento*, Formato PDF, Rio de Janeiro-Brasil-2013, p.27.

AVERÍA - FALLO. Suceso no previsible, en un equipo o cualquiera de sus elementos, que denota, que este experimenta un estado fuera de su condición normal de utilización, forzando su indisponibilidad. Es súbita de carácter aleatorio e instantáneo y casi siempre depende de la acumulación de diversos factores difíciles de predecir.⁹

Tanto las averías como las anomalías pueden ser de menor grado, mediante un correcto plan de gestión vehicular y mantenimiento.

⁹ Respectiva definición en los libros mencionados en la "cita 3".

MÉTODOS DE MANTENIMIENTO

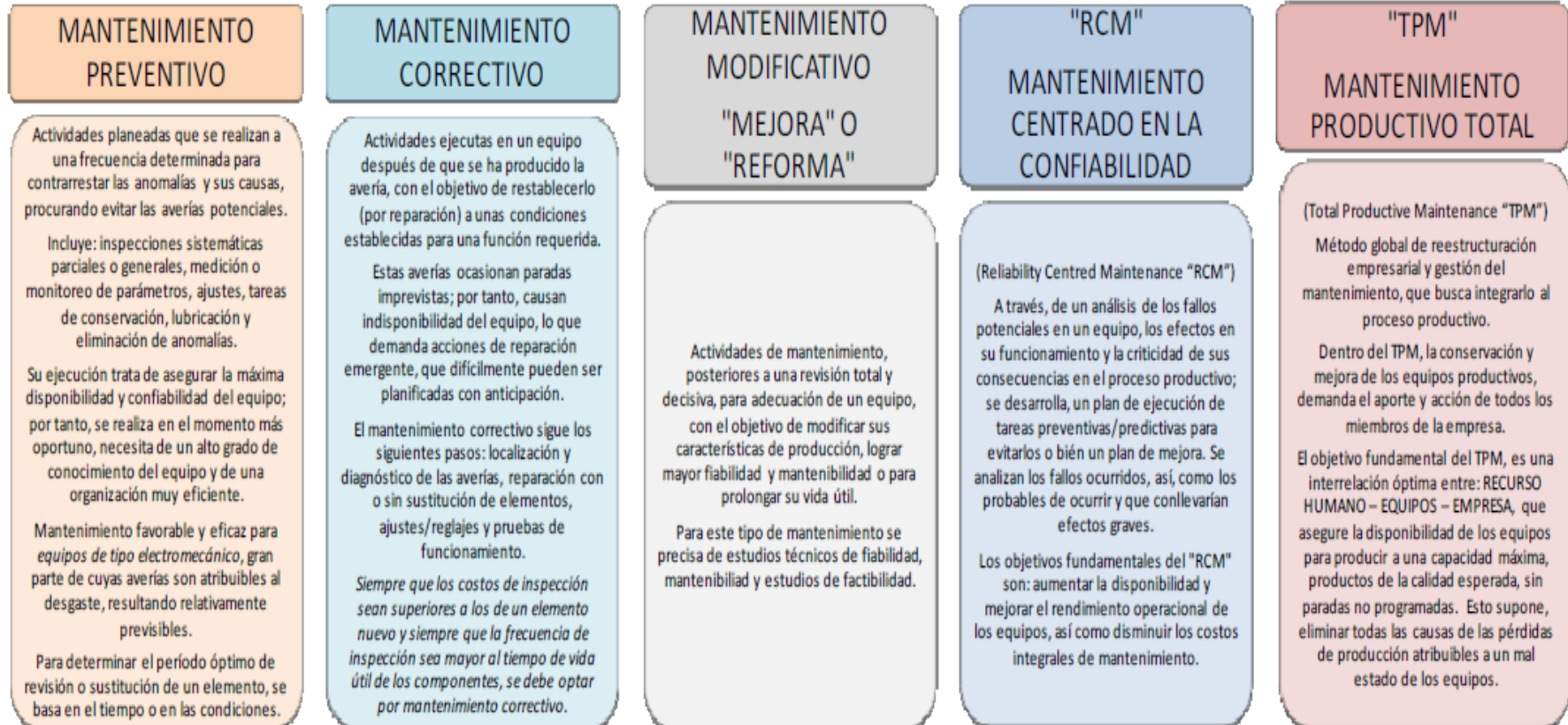


Figura 2. Métodos de mantenimiento. Fuente: Tavares, Lourival., Administración moderna del mantenimiento, 2003; García Garrido, Santiago, Ingeniería del mantenimiento, 2009

2.6. Base de Datos

Una base de datos bien estructurada es fundamental para la elaboración de cualquier proyecto.

Según (Kendall & Kendall, J, 2011), manifiesta que:

Una base de datos es una fuente central de datos con el fin de que varios usuarios la compartan para su uso en varias aplicaciones. El corazón de una base de datos es el sistema de administración de bases de datos (DBMS), el cual permite crear, modificar y actualizar la base de datos, la recuperación de los datos y la generación de informes y pantallas. p. 403.

Otro concepto de base de datos según (Coronel, Morris, & Rob, P, 2011), dice que:

Una **base de datos** es una estructura computarizada compartida e integrada que guarda un conjunto de:

- Datos del usuario final, es decir, datos sin elaborar que son de interés para el usuario final.
- **Metadatos**, o datos acerca de datos, por medio de los cuales los datos del usuario final son integrados y manejados. p. 7.

2.6.1. Funciones del DBMS

La parte principal de una base de datos es el sistema de administración de base de datos (DBMS).

Según (Coronel, Morris, & Rob, P, 2011), dicen que:

“El DBMS sirve como intermediario entre el usuario y la base de datos. La estructura misma de la base de datos esta guardada como un conjunto de archivos y la única forma de tener acceso a los datos de estos archivos es a través del DBMS”. p. 7.

2.6.2. Ventajas del DBMS

Las ventajas de tener un DBMS son varias, las principales según (Coronel, Morris, & Rob, P, 2011), son:

“Primero el DBMS hace posible que los datos de la base sean compartidos entre las múltiples aplicaciones o usuarios. En segundo término, el DBMS integra las numerosas y diferentes visitas de usuarios de los datos en un solo depósito de datos que lo abarca todo”. p. 8.

Como postulante se razona que una Base de Datos es un gran grupo de datos o información guardada o almacenada en computadoras, en forma organizada, siempre a nuestra disposición en forma automática.

2.6.3. Gestor de Base de Datos

Un gestor de base de datos según www.mcgraw-hill.es, extraído el 29 de Septiembre del 2014 a las 15:51, disponible en la pagina web <http://www.mcgraw-hill.es/bcv/guide/capitulo/8448148797.pdf>, manifiesta que:

Definimos un **Sistema Gestor de Bases de Datos SGBD**, llamado también DBMS (Data Base Management System) como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina **Base de Datos** o **BD**, (Data Base Base). p. 7.

Otra definición según (Sánchez, 2007), dice que un gestor de base de datos es: “La porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica”. p. 7.

2.6.4. Objetivos de un Sistema Gestor de Base de Datos

Según (Silberschat, Korth, & Sudarshan, 2002), “El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto *práctica* como *eficiente*”. p. 1.

2.7. Herramientas Open Source

2.7.1. Open Source

Según www.mclibre.org recuperado el 10 de Julio del 2014, a las 22:47 disponible en la Web <http://www.mclibre.org/descargar/docs/libros/libro-blanco-sl-iii-200702.pdf> dice que: “Open Source representa también una profunda agitación en las técnicas de desarrollo al ofrecer una respuesta innovadora a la creciente demanda de interoperabilidad entre los sistemas del futuro y una demostración de la eficacia de los frutos surgidos de la <inteligencia colectiva> en internet”.

Según www.opensource.org recuperado el 10 de Julio del 2014, a las 23:21 disponible en la Web <http://opensource.org/> dice que: ”Software de código abierto es software que se puede utilizar libremente, cambiando, y compartiendo (en forma modificada o no modificada) por cualquier persona.”

En otro concepto según tuxfiles.wordpress.com recuperado el 10 de Julio del 2014 a las 23:30 disponible en la web <http://tuxfiles.wordpress.com/free-software-vs-open-source/> manifiesta que: “Código Abierto se refiere a la posibilidad de ver el código fuente del programa. Pero el concepto va más allá del simple hecho de ver el código sino que la idea de que todos puedan ver el código fuente es que este evolucione, mejore y se desarrolle según las necesidades de los usuarios. Se corrigen los errores de manera mucho más rápida que si o hiciera una empresa con software privativo y por ende una producción de software mejor.

2.7.2. Requisitos de Open Source

Como en software libre, el Código abierto también tiene una serie de requisitos que cumplir para ser considerado Código abierto:

- Libre redistribución: el software debe poder ser regalado o vendido libremente.
- Código fuente: el código fuente debe estar incluido u obtenerse libremente.
- Trabajos derivados: la redistribución de modificaciones debe ser permitida.

- Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas solo como parches.
- Sin discriminación de personas o grupos: nadie puede dejarse fuera.
- Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos
- Distribución de la licencia: debe aplicarse los mismos derechos a todo el que reciba el programa.
- La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto también sea de código abierto.
- La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.”

Open Source es de gran utilidad para aquellas personas que se dedican a desarrollar páginas web, ya que es un software libre y su código fuente está abierto a todas las personas interesadas en él.

2.7.3. Características de Open Source

Según www.anf.es/es/pdf/bbs-open-source.pdf, extraído el 18 de Septiembre del 2014 a las 11:27, disponible en la Web <https://www.anf.es/es/pdf/bbs-open-source.pdf>, dice:

“El software Open Source se caracteriza por:

- Un muy bajo coste de implementación y mantenimiento.
- Un sistema sumamente estable, de muy baja incidencia de errores críticos.
- Permitir que las aplicaciones sean copiadas y distribuidas para su mejora continua.
- No ser susceptible a los virus comunes”.

2.7.4. El Open Source como Producto de Distribución Libre

Según MUNDOINFORMATICA.PORTALMUNDOS.COM recuperado el 16 de septiembre del 2014, a las 15:00 disponible en la web <http://mundoinformatica.portalmundos.com/herramientas-opensource-para->

el-desarrollo-y-diseño-de-paginas-web/nos dice que una Herramienta Open Source es:

El producto de distribución libre es un producto libre al tratarse de productos informáticos nos referimos a su código fuente con lo que podremos obtenerlo, modificarlo y redistribuirlo sin ningún problema. Esta filosofía a dado lugar al movimiento de software libre, el cual esta siendo impulsado activamente en España por organismos autonómicos tales como la junta de Extremadura o la junta de Andalucía y que tiene en Richard Stallman, creador de la FSF (Free Software Foundation), a su máximo exponente y profeta. Open Source a este sistema operativo en sus múltiples distritos y meta-distritos, dado que también podemos encontrar su equivalente para Windows u otro sistema en GNU.

2.7.5. Acerca de las Licencias Open Source

En lo referente a las licencias Open Source según opensource.org/ recuperado el 16 de Septiembre del 2014 a las 15:35, disponible en la web. <http://opensource.org/licenses> nos dice que:

Licencias de código abierto son licencias que cumplen con la definición de código abierto – en breve, permiten al software ser utilizado libremente, modificarlo, y se comparte. Para ser aprobada por la Open Source Initiative (también conocido como el OSI) una licencia debe pasar por el proceso de revisión de la licencia de Open Source Initiative.

Las siguientes licencias son aprobadas por la OSI :

- Apache Licence 2.0
- BSD 3-Cláusula de “Nuevo” o licencia “revisado”

- BSD “-Cláusula “simplificado” o licencia “FreeBSD”
- GNU General Public Licence (GPL)
- Biblioteca GNU o “lesser” General Public Licence (LGPL)
- Licencia MIT
- Mozilla Public Licence 2.0
- Common Development and Distribution Licence
- Eclipse Public Licence.

Las licencias open source son un tipo de licencias gratuitas de gran utilidad para la realización de proyectos.

2.7.6. Importancia de Open Source

Como investigador se considera que el uso de herramientas Open Source es de gran ayuda para los creadores de software como en mi caso personal, ya que es manipulable a las necesidades de cada programador según el tipo de proyecto que se vaya a desarrollar, aclarando que se tiene libre disponibilidad de modificarlo, usarlo o distribuirlo pero en ocasiones no será gratuito, tendrá un costo mínimo en comparación con el software tradicional.

2.7.7. Ventajas, Inconvenientes y Perspectivas del Futuro de Open Source

2.7.7.1. Ventajas

- Ahorro de costes, las distribuciones Open Source son generalmente gratuitas.
- El código es abierto, es decir ninguna empresa es propietaria del mismo, por lo tanto los programadores hacen el uso deseado por ellos.
- Mayor seguridad y calidad en los programas, ya que al ser libre el grupo de programadores pueden detectar mejor los errores y mejorar el programa.
- Incentiva la competencia

- Al ser libre se desarrolla y evoluciona más rápido, es mejor que el software de código cerrado.

2.7.7.2. Inconvenientes

Según (Cobo, Gómez, Pérez, & Rocha, 2005), algunos de los inconvenientes más notables son:

- Dificultad en cambiar o dejar de utilizar determinadas aplicaciones de software cerrado que se han convertido en un estándar.
- La utilización de software *open source* no es la mejor opción para determinadas empresas que basen su negocio en algoritmos secretos o planteados.
- La utilización de herramientas *open source* generalmente requiere, tanto para el usuario doméstico como para el personal de la empresa, un proceso de adaptación y formación, no porque las herramientas *open source* sean más complicadas sino simplemente debido a que los usuarios nunca han trabajado con ellas.
- Existencia de proyectos de desarrollo y comunidades en algunos casos sin estructuras jerárquicas ni organizativas claras que dan al usuario sensación de desorganización p. 52.

2.7.8. Perspectivas de futuro

Según (Cobo, Gómez, Pérez, & Rocha, 2005), al referirse a las perspectivas de open Source dice:

Hablar de *open source* es más que hablar de software gratis o de disponer de un código fuente. Supone una nueva forma de producir, desarrollar y distribuir software en la que el usuario, las comunidades e Internet adquieren un papel protagonista.

El continuo incremento de usuarios públicos y privados, de comunidades de desarrollo y de mejoras técnicas afianzan que la corriente *open source* no es una moda pasajera dentro del desarrollo de software. Más aun, predicen y hacen aventurar que su verdadero esplendor está por llegar, siendo indicativo de esto mismo la creación del “*Observatorio de software de fuente abierta*” por parte de la unión europea para estudiar este fenómeno en los actuales y futuros estados miembros y la elaboración desde la UNESCO de un programa de actuaciones en apoyo del software libre.

Por último, las voces más optimistas predicen en artículos y comentarios la rápida desaparición de las compañías de software cerrado p. 53.

2.7.9. Software Libre

Al referirse a software libre (Stallman, 2004) dice:

Con software libre nos referimos a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Nos referimos especialmente a cuatro clases de libertad para los usuarios de software:

- Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.
- Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades –el acceso al código fuente es condición indispensable para esto.
- Libertad 2: la libertad para distribuir copias y ayudar así a tu vecino.
- Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad –el

acceso al código fuente es condición indispensable para esto. p. 45.

El software libre es de gran utilidad para todas aquellas personas que por no tener dinero para obtener una licencia por su elevado costo, pueden utilizar este tipo de software; además de permitir desarrollar habilidades como desarrollador de sistemas.

2.8. Herramientas Open Source Para Desarrollo De Sistemas

2.8.1. Framework de Desarrollo Web

2.8.2. Framework

Según www.desarrollomovilmultiplataforma.blogspot.com/ extraído el 25 de Septiembre del 2014 a las 14:57, disponible en web <http://desarrollomovilmultiplataforma.blogspot.com/2012/08/aspectos-teoricos-framework> dice que un:

Un framework es una colección organizada de clases que constituyen un diseño reutilizable para un dominio específico de software. Contiene un conjunto de librerías, componentes de software y directrices arquitectónicas que ofrece al desarrollador un kit de herramientas completo para construir una aplicación de principio a fin, siempre teniendo en cuenta que es necesario adaptarlo a cada aplicación en particular.

2.8.3. Framework CodeIgniter

Existen diferentes tipos de Framework Open Source, previo un análisis estimativo entre los más aptos para la elaboración de este proyecto; se escogió el Framework CodeIgniter, por motivos que se detallan a continuación.

El framework CodeIgniter según www.etnassoft.com/ extraído el 25 de Septiembre del 2014 a las 15:38, manifiesta:

CodeIgniter es un framework para desarrollo de aplicaciones – un conjunto de herramientas – para gente que construye sitios web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido que lo que podría hacer si se escribiera el código desde cero, proveyéndole un rico conjunto de bibliotecas para tareas comunes, así como una interfaz sencilla y una estructura lógica para acceder e esas bibliotecas.

2.8.3.1. Características de Codeigniter

Las principales características de CodeIgniter según (Tamayo, Lemes, & Naranjo) son:2.8.3.1.1. Versatilidad: Es capaz de trabajar en la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde solo tenemos un acceso FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.

2.8.3.1.2. Compatibilidad: Es compatible con cualquier versión PHP, por lo que hace que se pueda utilizar en cualquier servidor.

2.8.3.1.3. Facilidad de instalación: Su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos.

2.8.3.1.4. Ligereza: El núcleo es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, solo cuando se van a utilizar realmente.

2.8.3.1.5. Documentación: La documentación es fácil de seguir y de asimilar, porque está escrita en modo tutorial. Esto no facilita mucho la referencia rápida, cuando Ya sabemos acerca del framework y queremos consultar sobre una función o un método concreto. p. 7 – 8.

2.8.3.2. Ventajas y desventajas de Codeigniter

2.8.3.2.1. Ventajas de Codeigniter

CodeIgniter es un framework muy potente sus ventajas principales, según (Bermúdez, Delgado, & Gonzáles , 2014) son:

Este framework tiene un juego de librerías para tareas comúnmente necesarias, así como una interfaz simple y estructura lógica para acceder a esas librerías; permite creativamente enfocarse en un proyecto específico minimizando la cantidad de código necesaria para una tarea dada. Es fácil de instalar y de aprender, características que provocan que sea ´preferido por aquellas personas que dispongan de poco tiempo para realizar un proyecto; otra de sus ventajas es que proporciona la escritura de código repetitivo. p. 68.

2.8.3.2.2. Desventajas de Codeigniter

En lo referente a las desventajas de CodeIgniter, que son pocas según Franco, A. (2013). *APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MEDICAS EN EL CENTRO MEDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO; UTILIZANDO EL PATRÓN DE ARQUITECTURA MVC EN PHP* (Tesis de ingeniería en sistemas computacionales, Universidad Técnica del Norte). Extraído de <http://repositorio.utn.edu.ec/handle/123456789/1693> manifiesta que:

“Dentro de los inconvenientes que son mínimos podemos destacar:

- Curva de aprendizaje: necesidad de aprender nuevas funciones, estructuras y métodos de programación.
- Dificultad para adaptar el código escrito en PHP tradicional”. p. 39.

2.8.3.3. Historia y Evolución de Codeigniter

Según Franco, A. (2013). *APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MEDICAS EN EL CENTRO MEDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO; UTILIZANDO EL PATRÓN DE ARQUITECTURA MVC EN PHP* (Tesis de ingeniería en sistemas computacionales, Universidad Técnica del Norte). Extraído de <http://repositorio.utn.edu.ec/handle/123456789/1693> manifiesta que:

CodeIgniter fue desarrollado originalmente por Ricks Ellis (*CEO* de EllisLab). El framework se escribió para obtener buen rendimiento en el mundo real, donde muchas de las bibliotecas de clases, *helpers*, y subsistemas se tomaron prestados

del código base de *ExpressionEngine*. Actualmente, el equipo de desarrollo de ExpressionEngine lo desarrolla y lo mantiene. p. 40.

2.8.3.4. Funcionalidades de Codeigniter

Las funcionalidades de un programa no nos dicen nada concreto acerca de la calidad de su código o de su verdadero poder de función, por eso lo mejor es siempre probarlo.

Según www.etnassoft.com, extraído el 29 de Septiembre del 2014 a las 14:47, disponible en la web: <http://www.etnassoft.com/biblioteca/codeigniter-guia-del-usuario-en-espanol-> , algunas de las funcionalidades de CodeIgniter en desarrollos de aplicaciones web sean estas pequeños o grandes son:

- Sistema basado en Modelo-Vista-Controlador
- Soporte para base de datos con Active Record
- Validación de datos y formularios
- Seguridad y filtrado XSS
- Administración de sesiones
- Clase para enviar Email. Soporta adjuntos, Email de HTML/Texto, varios protocolos (sendmail, SMTP, y Mail) y más
- Biblioteca de manipulación de imágenes (recorte, redirección, rotación, etc). Soporta GD, ImageMagic, y NetPMB
- Clase para subir archivos
- Clase para FTP
- Localización
- Paginación
- Encriptación de datos
- Evaluación de rendimiento
- Caché de página completa
- Historial de errores
- Perfilado de la aplicación
- Clase para calendarios
- Clase para agente de usuario

- Clase para codificación Zip
- Clase de motor de plantillas
- Clase para Trackback
- Biblioteca XML-RPC
- Clase para pruebas de unidad
- URLs amigables para los motores de búsqueda
- Ruteo URI flexible
- Soporte para hooks y extensiones de clase
- Amplia biblioteca de funciones “helper”

2.8.3.5. Requisitos del Servidor

Los requisitos del servidor deben ser:

- PHP (<http://www.php.net>) versión 4.3.2 o superior
- Una base de datos es necesaria para la programación de la mayoría de aplicaciones web. Actualmente las bases de datos soportadas con MySQL (4.1 +), MySQLLi, MS SQL, Postgre, Oracle, SQLite, y ODBC.

2.8.4. Framework Zend

De los frameworks analizados se toma atención en el framework Zend ya que según http://es.encydia.com/pt/Zend_Framework, extraído el 24 de Enero del 2015 a las 09:52 dice que:

Zend Framework es un framework de aplicación web de código abierto, orientado a objetos, implementado en PHP 5 y licenciado como New BSD Licence. Zend Framework - frecuentemente referido como **ZF** _ y desarrollado con el objeto de simplificar el desarrollo web mientras promueve las mejores prácticas en la comunidad de desarrolladores PHP.

La arquitectura usela-gana del ZF permite que los desarrolladores reutilicen componentes cuando y donde ellos tengan sentido en sus aplicaciones sin requerir otros componentes Zf además de las dependencia mínimas. No hay por lo tanto ningún paradigma o normalizado que todos los usuarios Zend Famework deban

seguir, aunque ZF suministre componentes para los patrones del proyecto MVC y Table Gateway que son usados en la mayoría de aplicaciones ZF.

2.8.4.1. Historia

Zend Framework fue concebido en el año 2005 mientras muchos nuevos frameworks, tales como Ruby on Rails y Spring Framework, estaban ganando popularidad en la comunidad web development. ZF fue públicamente anunciado por primera vez en la Zend Conference. A la vez ningún framework ampliamente usado ha sido disponibilizado para la comunidad PHP para llenar completamente necesidades de desarrollo web similares. Los proyectistas de Zend Framework buscaron combinar características de uso final y rapid application development (RAD) de esos nuevos frameworks con la simplicidad de, apertura y practicidad del mundo real que es altamente valorada en la comunidad PHP.

2.8.4.2. Licenciamiento

Zend Framework está licenciado bajo Open Source Initiative (OSI)-approved New BSD Licence, y todos los contribuidores de código abierto deben firmar un Contributor Licence Agreement (CLA) software Foundation's CLA. El licenciamiento y las políticas de contribución fueron establecidas para frustrar cualesquier cuestión de propiedad intelectual de los usuarios ZF comerciales.

2.8.4.3. Requisitos

Zend Framework requiere PHP 5.1.4 o superior, aunque la Guía de Referencia del Programador Zend Framework recomienda PHP 5.2.4 o superior por mejoras de seguridad y desempeño incluidas entre esas versiones de PHP. PHPUnit 3.0 o superior es requerido para rodar la prueba unitaria embarcada con Zend Framework.

2.8.4.4. Características

- Todos los componentes son PHP 5 completamente orientados a objetos y tiene conformidad con Y STRICT
- Arquitectura úsela-gana con débil acoplamiento de componentes e interdependencias mínimas

- Implementación MVC extensivel soportando esbozos y templates basados en PHP por patrón
- Implementación flexible de Table Gateway para acceder datos de un banco de datos relacional en un ambiente orientado a objetos
- Soporte para múltiples sistemas de bancos de datos, incluye MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, e Informix Dynamic Server
- Autenticación y autorización basada en ACL usando una variedad de sistemas de backend
- Filtro de datos y validación para fortalecimiento de la seguridad de la aplicación
- Gestión de sesión
- Componente de configuración para promover una gestión de configuración consistente a través de Zend Framework y aplicaciones ZF
- Composición y entrega de email, recuperación vía mbox, Maildir, POP3 e IMAP4
- Indexación y búsqueda que soporta el formato de archivo índice Lucene
- Internacionalización y localización
- Creación de formularios usando PHP, archivos de configuración o XML
- Componente nativo PHP para lectura, actualización y creación de documentos PDF.

2.8.4.5. Código, documentación y patrones de prueba

Contribuciones de código de prueba para el Zend Framework son sujetas a un código riguroso, documentación y patrones de prueba. Todo código debe ir al centro de patrones de codificación y los prueba unitarios deben alcanzar un 80% de cobertura de código antes que el código correspondiente pueda ser movido para la marca de reléase.

2.8.4.6. Organizaciones que utilizan Zend Framework

Organizaciones usuarias de Zend Framework incluyen:

- Brainbits
- Berlin Museums
- Digital Sublimity
- Eurotransplant
- GNU/Linux Matters sea
- IBM
- Marseille
- Nokia

El Framework Zend es un software potente para creación de sistemas, en el cual el programador debe tener vastos conocimientos del mismo.

2.8.4.7. Tabla comparativa de framework Codeigniter y Zend

Estos dos framework están diseñados para la creación de proyectos webs, esta es una tabla comparativa según <http://formatovic.com/frameworks-php-breve-comparativa-de-codeigniter-symfony-y-zend/>

Tabla 1. Tabla comparativa de Codeigniter y Zend, <http://formatovic.com/frameworks-php-breve-comparativa-de-codeigniter-symfony-y-zend/>

	<u>Codeigniter</u>	<u>Zend</u>
PHP5	✓	✓
Múltiple DB's (Bases de datos)	✓	✓
ORM (Mapeo objeto relacional)	-	✓
Objetos DB	✓	✓
Plantillas	✓	✓
Caché	✓	✓
Validación	✓	✓
Ajax	Algo	✓
Autenticación	Biblioteca externa	✓
Módulos	-	✓
EDP (Programación dirigida a eventos)	-	-

Tabla 2. Comparativa de frameworks Codeigniter y Zend. Postulante

	Codeigniter	Zend
Fecha de creación	Año 2006	Año 2005
Uso	Fácil	Prudente
Licenciamiento	GNU	Open Source Initiative (OSI)
Código de programación	Open Source	Open Source
Nivel de Arquitectura	Pequeña pero potente	Alta
Instalación	Fácil	Prudente
Curva de aprendizaje	Fácil	Intermedio

CodeIgniter es un Framework potente, su instalación y uso son fáciles, además al ser un programa Open Source es gratuito y existen diversos manuales de usuario e información disponibles en español, lo cual facilita esta investigación, utiliza PHP, es apto para proyectos medianos y pequeños, la curva de aprendizaje es más corta que otros frameworks como el framework Zend, por consiguiente es idóneo para la elaboración de este proyecto,

2.9. PHP

En referencia a PHP (Cobo, Gómez, Pérez, & Rocha, 2005) manifiestan que :

PHP es un lenguaje interpretado del lado del servidor que surge tanto del lado de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares los programas son integrados directamente dentro del código HTML p. 23.

PHP es un lenguaje gratuito y multiplataforma, es decir trabaja en la mayoría de servidores web.

2.9.1. Ventajas de Php con Otros Lenguajes de Programación

Según manifiestan (Cobo, Gómez, Pérez, & Rocha, 2005), algunas de las ventajas de usar PHP son:

Comparado con ASP, la principal ventaja de PHP es su carácter multiplataforma. Por otro lado los programas en ASP resultan más lentos y pesados, y también menos estables. En los entornos Microsoft la ventaja de ASP es que los servidores web de Microsoft soportan directamente ASP sin necesidad de ninguna instalación adicional.

Comparando el lenguaje PHP con el lenguaje Perl, utilizado habitualmente en la programación CGI, puede decirse que PHP fue diseñado para el desarrollo de scripts orientados a web, mientras que Perl fue diseñado muchas cosas más y debido a esto, se hace muy complicado. La sintaxis de PHP es menos confusa y más estricta, pero sin perder la flexibilidad.

En comparación con ColdFusion, PHP es más rápido y eficiente para tareas complejas de programación, además PHP resulta mas estable y usa una menor cantidad de recursos. Por el contrario, ColdFusion posee un mejor gestor de errores, un buen motor de búsquedas, abstracciones de bases de datos y un gran número de funcionalidades para el procesamiento de fechas. Finalmente ColdFusion no está disponible para todas las plataformas.

En definitiva, PHP es uno de los lenguajes más utilizados actualmente en el desarrollo de aplicaciones web y viene experimentando un

constante crecimiento en su nivel de utilización en internet p. 23.

Otro concepto de PHP según (Vaswani, 2010), dice que:

Este lenguaje se utiliza en más de 20 *millones* de sitios Web y en más de un tercio de, los servidores Web en todo el mundo; no es algo despreciable, especialmente cuando se considera que el lenguaje ha sido desarrollado por completo por una comunidad de voluntarios repartida en todo el mundo y está disponible en internet ; sin costo alguno;

Durante los últimos años PHP se ha convertido, de *facto*, en la opción para el desarrollo de aplicaciones Web orientadas a base de datos, sobre todo por su escalabilidad, facilidad, uso y el amplio soporte para diferentes bases de datos, y formatos de estos. p. 4.

2.9.2. Características únicas de Php

PHP es diferente de otros lenguajes de programación como ASP.NET o JSP por estas razones:

(Vaswani, 2010), expone las siguientes características de PHP:

- Rendimiento
- Portabilidad
- Fácil de usar
- Código libre
- Soporte comunitario
- Soporte a aplicaciones de terceros. p. 6.

PHP es un lenguaje de programación para creación de páginas web dinámicas, es más seguro ya que no permite ver su código de programación y su utilización es gratuita y está a disposición de cualquier persona.

2.10. Dreamweaver

En la elaboración de este proyecto se utilizara Dreamweaver el cual según http://recursos.fundacionesplai.org/intranet/dinamizadores/recursos_tecnologicos/

[ejercicios_de_dreamweaver/materiales/Manual%20de%20Dreamweaver.pdf](#)
extraído el 24 de Enero del 2015 a las 13:38 manifiesta que:

Macromedia Dreamweaver es un editor de código de HTML, profesional para el diseño visual y la administración de sitios y páginas web. Tan si prefiere controlar el código HTML como si prefiere trabajar en un entorno de edición visual.

Otro concepto según <http://helpx.adobe.com/es/dreamweaver/using/creating-opening-documents.html>, extraído el 24 de Enero del 2015 a las 13:57 dice que:

Dreamweaver ofrece un entorno flexible para trabajar con varios documentos web. Además de documentos HTML, es posible crear y abrir una gran variedad de documentos basados en texto, incluidos archivos ColdFusion Markup Lnguaje (CFML), ASP, JavaScript y hojas de estilos en cascada (CSS). También admite archivos de código fuente, como Visual Basic, .NET, C# y Java.

Dreamweaver proporciona varias opciones para crear un documento nuevo. Puede crear cualquiera de los tipos de documentos siguientes:

- Un documento nuevo o plantilla en blanco
- Un documento basado en uno de los diseños de página predefinidos ue viene con Dreamweaver, incluidos más de 30 diseños de páginas basados en CSS
- Un documento basado en una de las plantillas existentes

Dreamweaver es un entorno informático que sirve para la elaboración de páginas web, idóneo para este proyecto.

2.11. Xampp

Según www.udemy.com recuperado el 18 de Septiembre del 2014 a las 20:00, disponible en la página web <https://www.udemy.com/blog/es/tutorial-xampp-como-usar-xampp-para-ejecutar-tu-propio-servidor-web/> dice que:

XAMPP son las siglas de Multiplataforma (X), Apache (A), MySQL (M), PHP (P) y Perl (P): Es una distribución sencilla y ligera de Apache que hace extremadamente fácil crear un servidor web local de prueba para los desarrolladores. Todo lo que se necesita para configurar un servidor web – aplicación servidor (Apache), base de datos (MySQL), y lenguaje de script (PHP) – está incluido en un simple fichero extraíble. XAMPP es también multiplataforma, lo que significa que funciona igual de bien tanto en Linux como en Mac y Windows. Dado que la mayoría de servidores web actuales usan los mismos componentes que XAMPP, hace que la transición de un servidor de pruebas local a un servidor de producción sea muy fácil también.

XAMPP es un servidor gratis, multiplataforma de gran utilidad y fácil instalación, con él se puede gestionar una base de datos como MySQL.

2.12. Servidor Web

Según www.ibrugor.com › Home › Desarrollo web, extraído el 18 de Septiembre del 2014 a las 20:50 disponible en la página web <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/> manifiesta que Servidor Web es:

Es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo http.

Los servidores web están alojados en un ordenador que cuenta con conexión a Internet. El web server, se encuentra a la espera de que algún navegador le haga una petición, como por ejemplo, acceder a una

página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red.

2.13. Servidor Apache

Según www.ibrugor.com › Home › Desarrollo web, extraído el 18 de Septiembre del 2014 a las 20:43 disponible en la página web <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/> manifiesta que:

El servidor web Apache HTTP, llamado también Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que se destaca por su seguridad y rendimiento.

2.14. MySQL

Según (Pavón, 2011), manifiesta que: “MySQL es, la base de datos elegida por la gran mayoría de programadores en PHP. Soporta el lenguaje SQL y la conexión de varios usuarios, pero, en general, se utiliza para aplicaciones de tamaño pequeño-medio”. p. 16.

MySQL es una base de datos muy popular en el planeta ya que es open source, es decir podemos acceder al código fuente o código de programación.

Otro concepto, según Gilfillan, I (n.a). *La Biblia MySQL*. Extraído de <http://intranet.inifap.gob.mx/normateca/doc/MANUAL/La%20Biblia%20de%20MySQL.pdf>, manifiesta que:

MySQL es un *sistema de administración de base de datos relacional* (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2. p. 40.

MySQL utiliza un lenguaje de consultas SQL.

2.14.1. Principales Características de MySQL

Según <http://downloads.mysql.com/docs/refman-5.0-es.pdf>, extraído el 7 de Octubre del 2014 a las 14:41 manifiesta que algunas de las principales características del software de base de datos MySQL son:

- Interioridades y portabilidad
- Escrito en C y C++
- Probado con un amplio rango de compiladores diferentes
- Funciona en diferentes plataformas
- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- APIs disponibles para C, C++, Eiffel, Java, Perl, Python, Ruby, y Tcl.
- Uso completo de multi-threaded del kernel.. Pueden usarse multiples CPUs si están disponibles.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Usa tablas de disco B-tree (MyISAM) muy rápidas con comprensión de índice.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si se desea añadir una interfaz SQL para una base de datos propia.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son utilizadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras la inicialización para consultas.
- El código MySQL se prueba con Pury (un detector de memoria perdida comercial) asi como un Valgrind, una herramienta GPL.

2.14.2. Por qué el uso de MySQL

Según el criterio de algunos autores al referirse al porqué del uso de MySQL manifiestan que entre las más importantes están:

- Su uso es muy fácil

- Su funcionalidad es optima
- Portabilidad
- Velocidad
- Costo
- Asistencia

2.14.3. Ventajas de MySQL

Las ventajas de MySQL son muchas, pero según el criterio de varios autores tenemos las siguientes:

- Fácil de configurar e instalar
- MySQL es Open Source
- El costo de requerimientos para la elaboración de bases de datos es bajo, podrá ejecutarse en un computador de escasos recursos
- Soporta varios Sistemas Operativos
- Es uno de los gestores de base de datos más veloz que existen
- La conectividad, y seguridad hacen de este software MySQL, un software altamente apropiado para crear bases de datos de cualquier tamaño.

MySQL es el gestor de base datos más utilizado en la actualidad por diseñadores de proyectos, ya que soporta diferentes tipos de plataformas y es Open Source.

CAPITULO III

3. METODOLOGÍA

El presente proyecto está enmarcado dentro de la investigación cuantitativa, su objetivo principal es el estudio de propiedades y fenómenos cuantitativos y las relaciones existentes en el mismo. Este tipo de investigación nos permitió analizar los inconvenientes iniciales del tema propuesto en este proyecto, mediante el empleo de encuestas, entrevistas y observaciones directas, que nos proporcionaron un porcentaje de eficiencia y aceptabilidad de los procesos de gestión vehicular por parte del personal que maneja estos datos. Al conocer estos porcentajes la investigación cuantitativa nos permitió medir y contrastar el grado de eficiencia inicial y final, una vez implementado el sistema propuesto.

3.1. Tipo de Estudio

En la elaboración de este proyecto se utilizó el siguiente tipo de estudio:

3.1.1. Por el propósito o finalidades perseguidas

3.1.1.1. Investigación básica.- Con la investigación básica se recolectó información científica en relación al tema propuesto.

3.1.1.2. Investigación aplicada.- Con la utilización de la investigación aplicada se resolvió el problema de estudio.

3.1.2. Según la clase de medios utilizados para obtener los datos

Se utilizó:

3.1.2.1. Investigación documental.- Con la investigación documental se indagó en documentos científicos referentes al tema.

3.1.2.2. Investigación de campo.- En este tipo de investigación entran las encuestas y entrevistas, que se realizaron al personal administrativo e instructores de la Escuela de Conducción de la UNACH. Todo esto se lo realizó en el lugar y tiempo real en que ocurren los fenómenos a estudiarse.

3.1.2.3. Investigación experimental.- Esta técnica de investigación sirvió para comprobar o rechazar la hipótesis planteada en este proyecto.

3.1.3. Por el nivel de conocimientos que se adquieren.

Se utiliza investigación exploratoria. La importancia radica en el uso de sus resultados para abrir líneas de investigación y proceder a su consecuente comprobación.

3.1.4. Por el campo de conocimiento en que se realiza.

Se necesitó de la Investigación Científica. Ya que la investigación científica es la rastreo en forma intencionada que una persona realiza en busca de conocimientos o de soluciones a dificultades de carácter científico; el método científico indica la ruta a seguir en la investigación y las técnicas precisas para la correcta elaboración de este proyecto..

3.1.5. Conforme al razonamiento empleado.

Conforme al razonamiento empleado se utilizó:

3.1.5.1. Investigación descriptiva.- La investigación descriptiva trabaja sobre realidades de hecho y presenta una interpretación correcta del problema que tenemos en la gestión vehicular en la Escuela de Conducción de la UNACH.

3.1.5.2. Experimental.- Con este tipo de investigación manejamos las variables, para saber de qué modo o por que causa se produce un acontecimiento en particular.

3.1.6. Conforme al número e investigadores.

Se utilizó Investigación individual, ya que la investigación, elaboración e implementación del proyecto está a cargo de una sola persona.

3.2. Métodos

El presente trabajo de investigación se encuentra enmarcado en base a los métodos Inductivo-Deductivo y Analítico.

Inductivo.- Utilizando este método se observó el problema de la Escuela de Conducción en forma general, para posteriormente ahondar en el tema.

Deductivo.- Una vez recopilada y analizada toda la información se llegó a dar solución al problema inicial.

Analítico.- Mediante este método fue posible analizar todos los problemas existentes en la forma de llevar la gestión vehicular en la Escuela de Conducción, fraccionando el problema y analizándolo por partes, así se dio solución a cada uno de ellos, por consiguiente se obtuvo los resultados propuestos en el proyecto.

3.3. Población y Muestra

3.3.1. Población

Para la elaboración de este proyecto se toma como población a:

- 7 instructores de la escuela de Conducción de la UNACH
- Personal administrativo de la misma (Director administrativo, Secretaria, e inspectora de la Escuela de Conducción de la UNACH).

Tabla 3. Nómina de personas que conforman la población para la elaboración del proyecto.

Nómina de personas que conforman la población para la elaboración del proyecto de investigación.	
Personal administrativo	
Cargo	Nombre
1 Directora administrativa	Ing. Diana Erazo
1 Secretaria	Ing. Gina Zabala
1 Inspector general	Lcda. Isabel Ávalos
Instructores de conducción	
Sr. Gabriel Santos	Vehículo # 01
Lcdo. Roberto Santos	Vehículo # 02
Lcdo. Pablo Damián	Vehículo # 05
Ab. Ramiro Yépez	Vehículo # 07
Tglo. José Luís Gallegos	Vehículo # 08
Sr. Hugo Flores Dávila	Vehículo # 09
Ing. Gonzalo Melena	Vehículo # 10
Total de la población = 10 Personas.	

3.3.2. Muestra

En base a la población existente en la Escuela de Conducción de la UNACH, y dada que es pequeña; para la muestra en la elaboración de este proyecto se escoge el 100 % de la misma, por consiguiente no es meritorio realizar el proceso de la muestra.

3.3. Operacionalización de las Variables

3.3.1.- Variable independiente.

Tabla 4. Variable independiente

Variable Independiente	Indicadores	Técnica	Fuente Verificación / Instrumento
Sistema de Gestión Vehicular en la Escuela de Conducción de la UNACH	<ul style="list-style-type: none"> Analizar los módulos que forman el Sistema 	Investigación	Libros
	<ul style="list-style-type: none"> Identificar la mejor Herramienta Open Source 	Investigación	Libros
	<ul style="list-style-type: none"> Diseñar la Interfaz adecuada para facilitar el uso del Sistema 	Investigación	Libros

3.3.2.-Variable dependiente.

Tabla 5. Variable dependiente

Variable Dependiente	Indicadores	Técnica	Fuente Verificación / Instrumento
Gestión de los procesos de mantenimiento preventivo de los vehículos en la Escuela de conducción de la UNACH	<ul style="list-style-type: none"> Controlar Agilizar Procesos Organizar Identificar 	Observación Entrevista Entrevista Encuesta	Cuestionario para Instructores y Personal Administrativo

3.4. Procedimientos

3.4.1. Fuentes de información

Las fuentes de información que contribuyeron para la elaboración de este proyecto de investigación fueron:

a) Fuentes de información primarias.- Ya que los datos fueron tomados mediante el uso de entrevistas y encuestas dirigidas al personal administrativo e instructores, además se empleó la observación directa del problema de estudio en la Escuela de Conducción de la UNACH.

b) Fuentes de información secundarias.- Las fuentes de información secundarias se obtuvo de libros, folletos e internet; todos estos con temas relacionados al proyecto de investigación

3.5. Procesamiento y Análisis

La estadística descriptiva es el método de investigación empleado de la investigación cuantitativa, para el procesamiento y análisis general de los datos obtenidos. Además nos permitirá obtener los lineamientos necesarios para la correcta creación y futura implementación del sistema de gestión vehicular de la Escuela de Conducción de la UNACH utilizando herramientas Open Source.

3.5.1. Encuesta dirigida a los Instructores de la Escuela de Conducción de La UNACH.

PREGUNTA N° 1

¿Conoce usted a los cuantos kilómetros de recorrido se debe dar el respectivo mantenimiento preventivo de los autos de práctica de conducción?

Tabla 6.

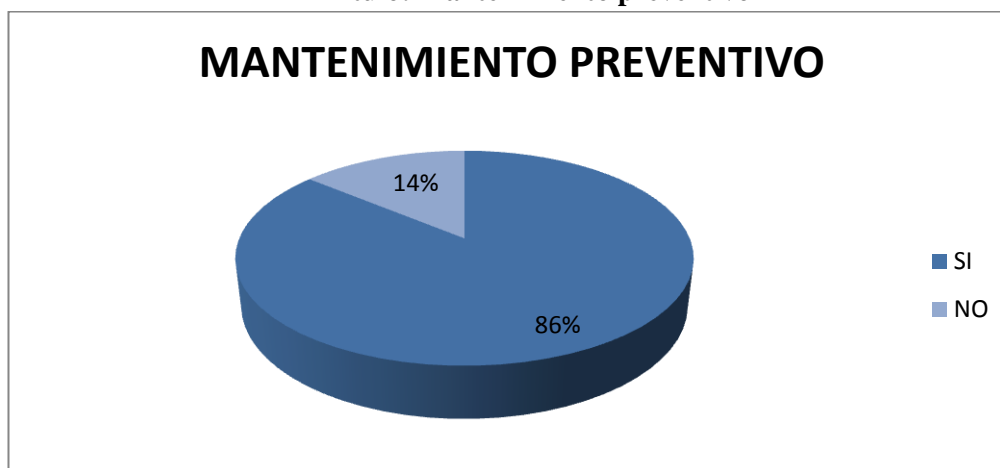
Título: Mantenimiento preventivo

RESPUESTA	TABULACIÓN	PORCENTAJE
SI	6	86 %
NO	1	14 %
TOTAL	7	100 %

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Figura 3.

Título: Mantenimiento preventivo



Fuente: Tabla 6

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN:

El 86% de los encuestados mencionan que si conocen a cuantos kilómetros de recorrido del auto, se debe realizar el respectivo mantenimiento preventivo, lo cual es un aporte adicional para que el sistema de gestión vehicular sea mas preciso, 14 % dice desconocerlo.

PREGUNTA N° 2

¿Tiene Ud. suficientes conocimientos en Informática o Computación?

Tabla 7.

Título: Conocimientos de computación

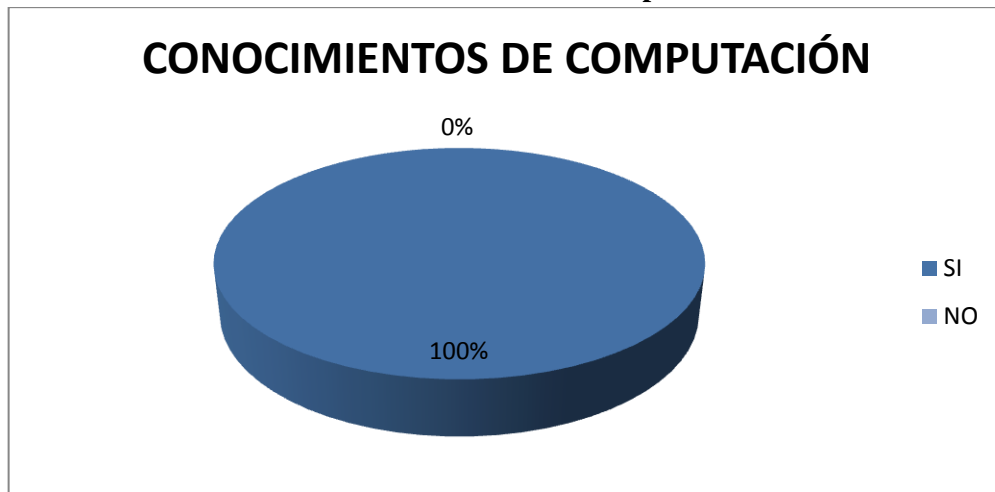
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	7	100 %
NO	0	0 %
TOTAL	7	100 %

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 4.

Título: Conocimientos de computación



Fuente: Tabla 7

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN:

El 100 % de los encuestados afirman tener conocimientos suficientes en computación, lo cual permitirá el manejo del sistema sin problemas.

PREGUNTA N° 3

¿Con la implementación de un sistema informático de gestión vehicular, cree Ud. que la elaboración de informes sería ágil y tener documentación para su respaldo lo ayudaría?

Tabla 8.

Título: Implementación del sistema informático

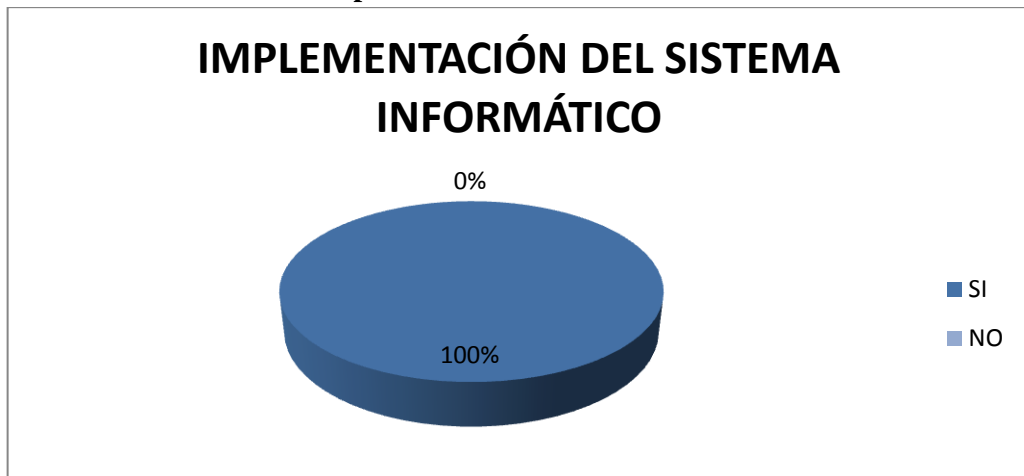
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	7	100 %
NO	0	0 %
TOTAL	7	100 %

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 5.

Título: Implementación del sistema informático



Fuente: Tabla 8

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN:

El 100 % de los encuestados cree que la implementación de un sistema informático, los ayudaría a agilizar los informes de mantenimientos requeridos y tener la documentación almacenada en un computador sería más confiable que tenerla guardada en papel.

PREGUNTA N° 4

¿Conoce Ud. el tipo de mantenimiento que se debe hacer en el auto de práctica dependiendo del kilometraje en el que se encuentre. Adicional al que se realiza cada 5000 Km?

Tabla 9.

Título: Tipo de mantenimiento

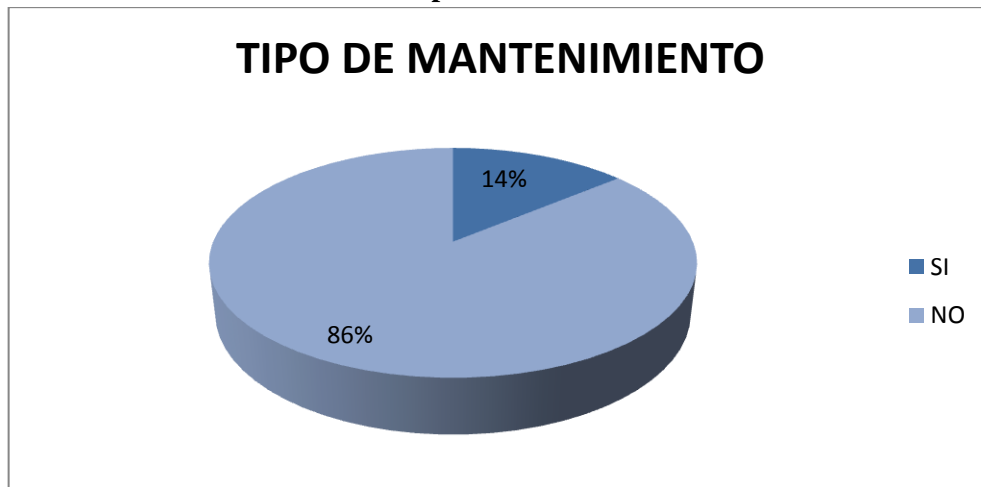
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	1	14 %
NO	6	86 %
TOTAL	7	100 %

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 6.

Título: Tipo de mantenimiento



Fuente: Tabla 9

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN

El 14 % de los encuestados dice saber a los cuantos kilómetros se deben realizar otros tipos de mantenimiento diferente al habitual, y el 86 % dicen no saber que tipo de trabajo y cuáles son los dispositivos o sistemas a revisarse en este mantenimiento. Con la creación e implementación del sistema de gestión vehicular se tendrá esta información en forma inmediata y al alcance de todos.

PREGUNTA N° 5

¿Si se requiere o si se le solicita presentar un informe en referencia a la gestión vehicular, la información es veraz y oportuna?

Tabla 10.

Título: Informes de gestión vehicular

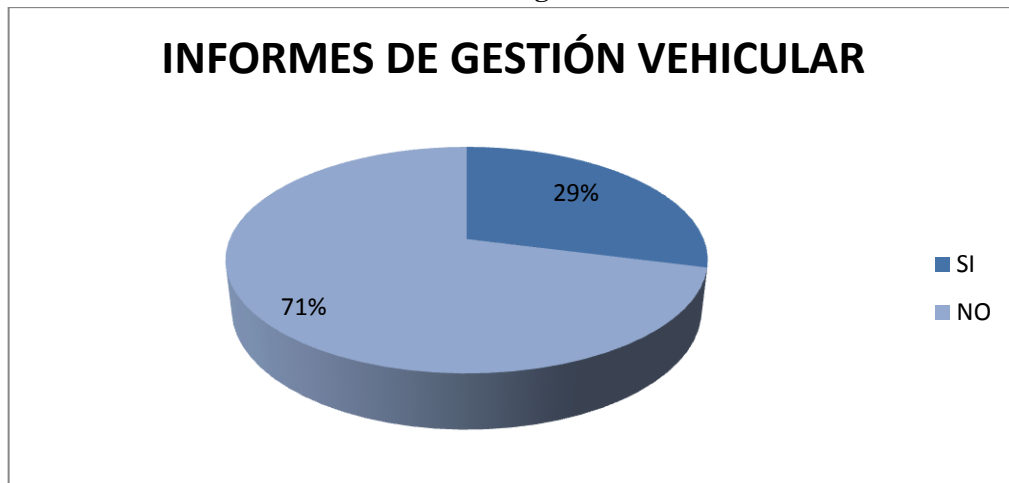
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	2	29 %
NO	5	71 %
TOTAL	7	100%

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 7.

Título: Informes de gestión vehicular



Fuente: Tabla 10

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN:

El 71 % de los encuestados dice que al momento de realizar sus informes la información no está disponible en forma inmediata, ni se la elabora en forma instantánea; lo cual implica pérdida de tiempo e informes no confiables, lo cual indica la necesidad de la implementación de un sistema de gestión vehicular, el 29 % de los mismo mencionan que al momento de ser requerido un informe lo hacen sin problema

PREGUNTA N° 6

¿Cree Ud. que el mantenimiento y los procesos de gestión vehicular actualmente son los adecuados?

Tabla 11.

Título: Mantenimiento y procesos de gestión vehicular

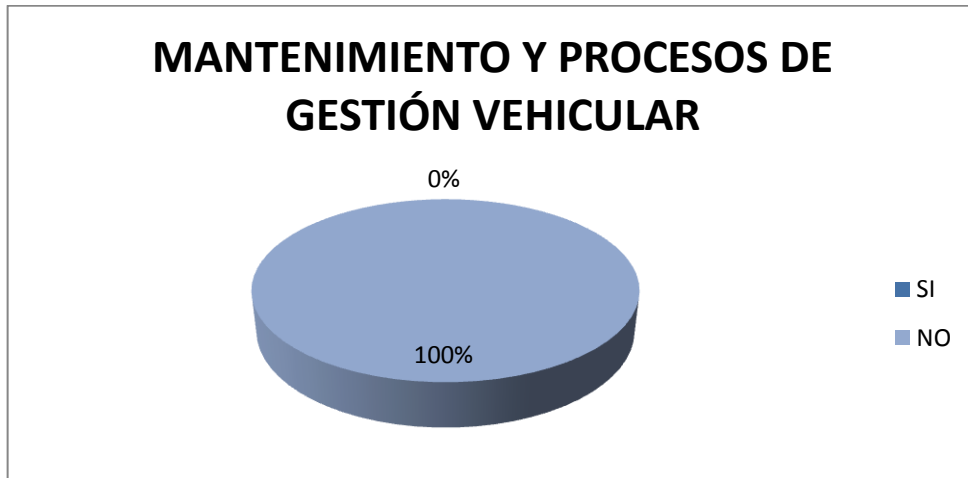
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	0	0 %
NO	7	100 %
TOTAL	7	100%

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 8.

Título: Mantenimiento y procesos de gestión vehicular



Fuente: Tabla 11

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN:

El 100% de los encuestados dicen que los procesos de mantenimiento y gestión vehicular no son los adecuados.

PREGUNTA N° 7

¿Cree Ud. que se debería implementar un sistema informático en la Escuela de Conducción de la UNACH para gestionar de mejor forma la gestión vehicular?

Tabla 12.

Título: Implementación de un sistema informático

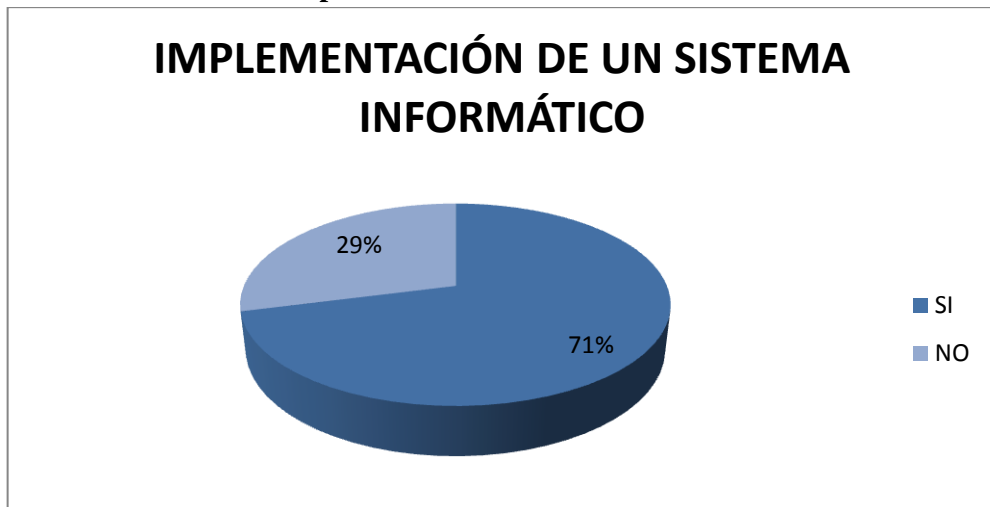
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	5	71 %
NO	2	29 %
TOTAL	7	100%

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 9.

Título: Implementación de un sistema informático



Fuente: Tabla 12

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN:

El 71 % manifiesta que es necesario la implementación de un sistema informático de gestión de procesos vehiculares.

PREGUNTA N° 8

¿Piensa Ud. que es importante la utilización de un sistema informático en la Escuela de Conducción de la UNACH para gestionar de mejor forma el control vehicular?

Tabla 13.

Título: Importancia del sistema informático

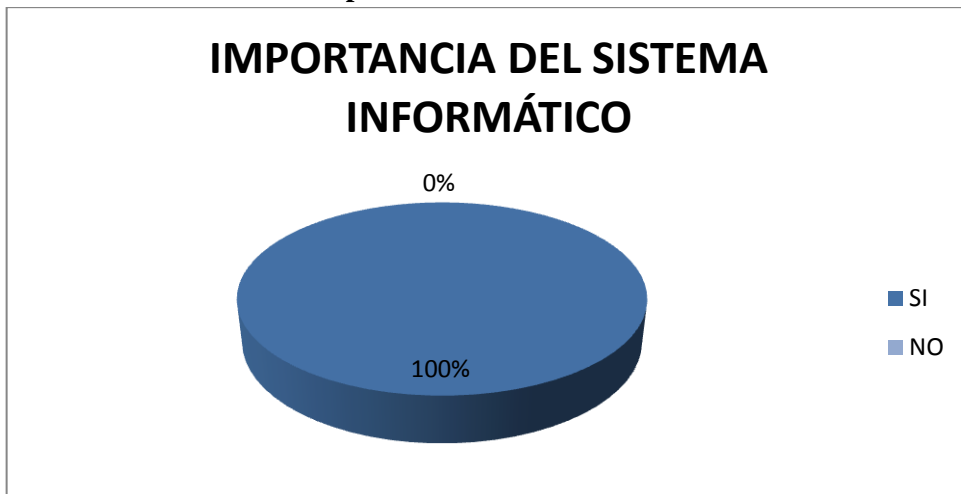
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	6	86 %
NO	1	14 %
TOTAL	7	100%

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 10.

Título: Importancia del sistema informático



Fuente: Tabla 13

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN:

El 14 % de los encuestados indican que un sistema informático no es tan necesario, mientras que el 86 % de los mismos dicen que es sumamente importante y necesario dicho sistema.

PREGUNTA 9

¿Cree Ud. que la implementación de un sistema informático de gestión vehicular sería eficiente para la administración actual e instructores de conducción para llevar de mejor forma todos los procesos involucrados en el tema?

Tabla 14.

Título: Eficiencia del sistema de gestión vehicular

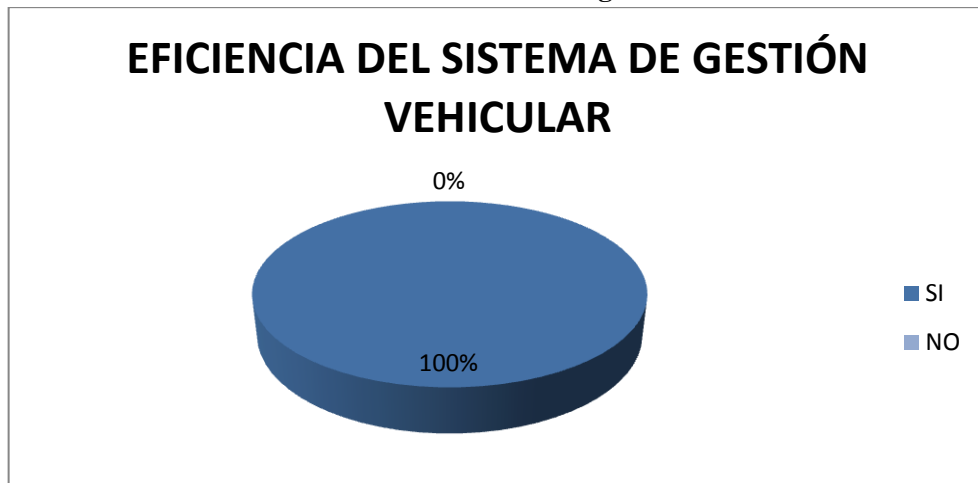
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	7	100 %
NO	0	0 %
TOTAL	7	100%

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 11.

Título: Eficiencia del sistema de gestión vehicular



Fuente: Tabla 14

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN:

El 100 % de los encuestados, dicen que denotaría mayor eficacia en el personal administrativo e instructores la implementación de un sistema de gestión vehicular en la Escuela de Conducción de la UNAH, en todo lo referente al tema.

PREGUNTA N° 10

¿Cree Ud. que está en capacidad de manejar y adaptarse a la utilización de un sistema informático de gestión vehicular?

Tabla 15.

Título: Utilización y adaptación al nuevo sistema informático

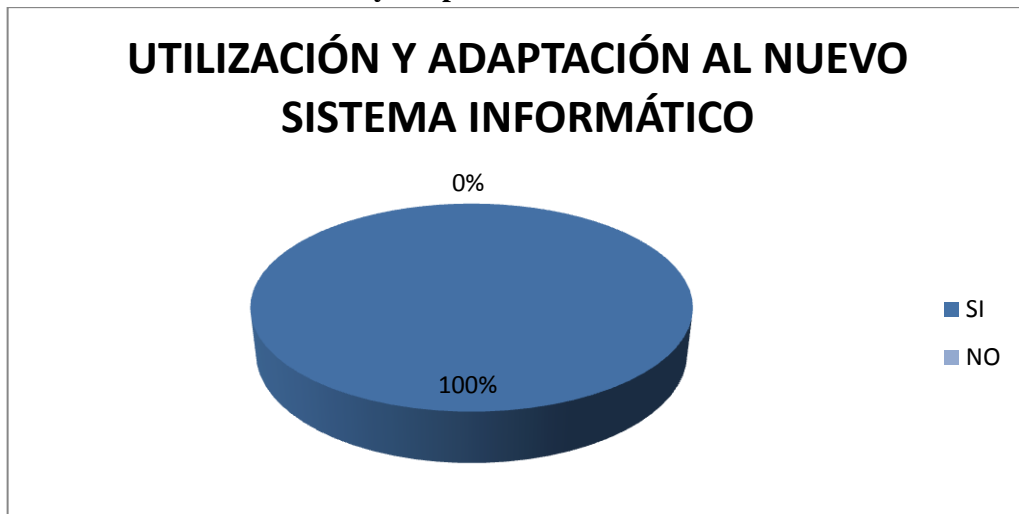
RESPUESTA	TABULACIÓN	PORCENTAJE
SI	7	100 %
NO	0	0 %
TOTAL	7	100%

Fuente: Encuesta realizada a los instructores de la Escuela de Conducción

Elaborado por: Hugo Flores Dávila

Figura 12.

Título: Utilización y adaptación al nuevo sistema informático



Fuente: Tabla 15

Elaborado por: Hugo Flores Dávila

ANÁLISIS E INTERPRETACIÓN

El 100 % de los encuestados dicen que se adaptarían y que manejarían el nuevo sistema informático, con la debida capacitación del proponente del proyecto.

3.5.2. Análisis de los resultados de la entrevista realizada al director y secretaria de la Escuela de Conducción de la Universidad Nacional de Chimborazo.

En la entrevista realizada al Director, Secretaria e Inspectora de la Escuela de Conducción se obtuvo las siguientes conclusiones de las preguntas planteadas.

Pregunta 1: ¿Como Directora Administrativa de la Escuela de Conducción de la UNACH Ud. confirma que al ser solicitados los informes de gestión vehicular por parte de la autoridad superior, existe retraso en la entrega de los mismos?

La entrevistada afirma que al ser solicitado un reporte en relación al tema no se lo puede realizar en forma inmediata, ya que la información esta archivada y se la debe buscar minuciosamente.

Pregunta 2: ¿Cree Ud. que almacenar la información de administración, control, consumo de combustible, mantenimiento y todo lo referente a la gestión vehicular en papel y carpetas es un método adecuado?

En relación al tema la entrevistada manifiesta que llevar un control en esta forma no es apropiado.

Pregunta 3: ¿Cómo Directora de la Escuela de Conducción considera factible la creación e implementación de un sistema de información para la gestión de mantenimiento de los vehículos de práctica de conducción?

Respondiendo a esta pregunta la entrevistada dice que es muy apropiada la implementación de este software para llevar un control eficiente en lo referente a gestión vehicular en la Escuela de Conducción.

Pregunta 4: Como Secretaria de la Escuela de Conducción, cree Ud. que la implementación de un sistema de gestión vehicular le sería útil para llevar de mejor manera su trabajo y realizar de mejor forma cualquier informe solicitado y en forma oportuna?

La Secretaria de la Escuela de Conducción de la UNACH responde a esta pregunta diciendo que la implementación de un software de gestión vehicular optimizará su trabajo en relación al tema.

Pregunta 5: ¿Cómo Secretaria de la Escuela de Conducción estaría dispuesta a trabajar conjuntamente con el Director en la administración del nuevo sistema?

La entrevistada manifiesta que da su compromiso total de encargarse de la administración del nuevo sistema propuesto, y de vigilar su correcto funcionamiento.

Luego de haber analizado todas las respuestas que se dieron a las preguntas sugeridas, se decide que la implementación del sistema de gestión vehicular es factible de realizar y se lo pone en marcha.

CAPITULO IV

4. PROCESO DE INGENIERIA DE SOFTWARE

4.1. Ingeniería de la información

4.1.1. Definición del Ámbito del Problema

Es esta etapa es importante determinar cuáles son las necesidades de los usuarios. Como resultado de esta fase se puede determinar la base de la cuantificación de los requisitos.

En la actualidad en la Escuela de Conducción de la UNACH, el mantenimiento y control de la flota vehicular se realiza de manera deficiente, lo que ocasiona un desconocimiento de la situación real de los vehículos.

4.1.2. Investigar la organización.

4.1.2.1. Identificación de objetivos.

Los objetivos de la Escuela de Conducción de la UNACH son los siguientes:

- Proporcionar a la sociedad una nueva clase de conductores profesionales con licencia tipo C y D.
- Fomentar en los alumnos de la Escuela de Conducción de la UNACH los conocimientos certeros de leyes de tránsito, seguridad vial, mecánica automotriz y demás cátedras que les servirán al momento d desempeñarse como conductores profesionales.
- La Escuela de Conducción de la UNACH busca ser una de las mejores Escuelas de Conducción del país, fomentando una nueva forma de impartir conocimientos y automatizando sus procesos de gestión vehicular y práctica de conducción.

4.1.2.2. Identificación de estructura orgánico funcional

Es primordial saber cómo está organizada funcionalmente la Escuela de Conducción ya que de esta manera podemos respaldar la solución software.

4.1.2.3. Aplicar técnicas de investigación de campo

Para tener una idea más clara de lo que debe realizar la solución software se hace necesario que tanto el cliente (Escuela de Conducción) como el desarrollador tengan la misma idea de lo que debe realizar la solución software , para el efecto se ha realizado reuniones con el Director de la Escuela de Conducción y los Instructores.

4.1.3. Definición de Escenarios Problema.

Es trascendental dividir y categorizar los problemas de la Escuela de Conducción para poder deducirlos y formularlos de forma clara para una mejor definición.

Los escenarios que se han detectado se detallan a continuación:

- Deficiente mantenimiento y control de vehículos de práctica de Conducción.
- Inapropiado mantenimiento preventivo y mantenimiento correctivo.
- Reposición de piezas averiadas.
- Control de gastos por combustible y mantenimientos deficiente
- Desconocimiento de kilometrajes recorridos por los vehículos por período de instrucción.
- Falta de un historial exacto de cambios o composturas de accesorios de los vehículos.
- No existe un historial certero de accidentes sucedidos por unidad vehicular.
- Descoordinación de todas estas actividades
- Ineficaz control de la mayoría de procesos.

4.1.4. Definición de alternativas de solución.

Es preciso que se detallan procedimientos que permitan dar soluciones o respuestas a los escenarios problema que se han detallado en el punto anterior; en tal virtud se desarrollará un sistema informático para automatizar todo el proceso de mantenimiento vehicular.

4.2. Estimaciones

El sistema informático, se desarrollará basado en las estimaciones de proyectos generales. Se ha seleccionado el modelo COCOMO como técnica para las estimaciones. Para esto es necesario analizar el número de instrucciones fuente (KLDC) que se utilizará en la obtención de la aplicación, para ello se sustentará en el número de KLDC para proyectos estándares, para el proyecto consideramos 3000 LDC. Una vez que se ha definido los KLDC se procederá a calcular: esfuerzo (E hombres-mes), tiempo de desarrollo de software (D meses), productividad (P KLDC / hombres-mes), cantidad de hombres (CH hombres).

$$\begin{aligned}
 E &= 2.4 (\text{KLDC})^{1.05} & D &= 2.5 (E)^{0.38} \\
 E &= 2.4 (3000)^{1.05} & D &= 2.5 (7.61)^{0.38} \\
 E &= 7.61 \text{ hombres - mes} & D &= 5.41 \text{ meses} \approx 5 \text{ meses} \\
 \\ \\
 CH &= \frac{E}{D} & P &= \frac{(\text{KLDC} * 1000)}{E} \\
 CH &= \frac{3.6}{4.1} & P &= \frac{(3 * 1000)}{7.61} \\
 CH &= 1.4 \text{ hombres} \approx 1 \text{ hombre} & P &= \frac{394.22 \text{ LDC}}{\text{hombres - mes}}
 \end{aligned}$$

4.3. Planificación

Mediante un seguimiento por tareas se ha logrado establecer que el sistema informático tendrá una duración de 120 días equivalente a 4 meses, mediante la utilización de algunos recursos tales como: Entrevistador, Analista, Planificador, Diseñador, Programador y Usuarios.

Tabla 16. Planificación en tiempo del sistema

ACTIVIDAD	DURACIÓN
Formulación	5 días
Planificación	10 días
Análisis	30 días
Modelización	60 días
Pruebas	15 días
TOTAL	120 días

4.4. Análisis de Riesgos

Es necesario determinar los riesgos que se pueden presentar en el proyecto, para lo cual se le debe asignar un peso o porcentaje de que éste se presente y tenga incidencia, así como una descripción del mismo.

Entre los riesgos que se pueden presentar tenemos los siguientes:

- Errores en la estimación del presupuesto.
- Se pueden presentar requerimientos que no han sido considerados al momento de presentar las necesidades por parte de los representantes de la Escuela de Conducción de la UNACH.
- Desconocimiento del desarrollador en el manejo de herramientas o la inexperiencia influye en la implementación del proyecto.
- Pueden presentarse inconvenientes en la seguridad del software.
- Cambio de políticas de gestión.

4.4.1. Identificar los Riesgos

- Riesgos del Proyecto
 - Costo de Producción
- Riesgos Técnicos
 - Imprecisión en la Especificación de Requerimientos de Software
 - Problemas de diseño
 - Problemas en Verificación & Validación

- Riesgos de Negocios
 - Riesgo de gestión

4.4.2. Gestión y Supervisión del Riesgo

Después que han sido determinados los riesgos, es necesario gestionarlos de manera que provoquen el menor impacto posible en el proyecto, para esto utilizamos formatos en los que se detallan las causas y un plan de prevención para contrarrestar los posibles riesgos.

DOCUMENTO RIESGO 1
<p>1. IDENTIFICACIÓN</p> <p>Errores en la estimación del presupuesto</p> <p>2. TIPO</p> <p>Riesgo del Proyecto</p> <p>3. SUBTIPO</p> <p>Costo de producción</p> <p>4. CAUSAS</p> <p>Errores en los cálculos, no estimar bien los factores que influyen en el cálculo (archivos, funciones, etc.) o el caso contrario la sobre estimación.</p> <p>5. PLAN DE AVERSIÓN</p> <p>Buscar los recursos que sean necesarios para terminar el proyecto a tiempo, y no acumular actividades de tareas al personal.</p>

DOCUMENTO RIESGO 2
<p>1. IDENTIFICACIÓN</p> <p>Se pueden presentar requerimientos que no han sido considerados al momento de presentar las necesidades por parte de los representantes de la Escuela de Conducción de la UNACH.</p> <p>2. TIPO</p> <p>Riesgo Técnico</p> <p>3. SUBTIPO</p> <p>Imprecisión en la Especificación de Requerimientos de Software</p>

4. CAUSAS

Información deficiente.
Poca experiencia.
Documentos no existentes.

5. PLAN DE AVERSIÓN

Buena información
Crear documentos

DOCUMENTO RIESGO 3**1. IDENTIFICACIÓN**

Desconocimiento en el manejo de herramientas o la inexperiencia por parte del desarrollador influye en la implementación del proyecto.

2. TIPO

Riesgo Técnico

3. SUBTIPO

Problemas de diseño

4. CAUSAS

El equipo de desarrollo puede no contar con la experiencia suficiente en cuanto a utilización de las herramientas de desarrollo, implementación.

5. PLAN DE AVERSIÓN

Buscar material, manuales, talleres sobre las diferentes herramientas a ser utilizadas en el proyecto.

DOCUMENTO RIESGO 4**1. IDENTIFICACIÓN**

Pueden presentarse inconvenientes en la seguridad del sitio.

2. TIPO

Riesgo Técnico

3. SUBTIPO

Problemas en Verificación & Validación

4. CAUSAS

Falta de experiencia del personal del proyecto en las cuestiones relacionadas a Seguridad Web, control de ingreso malicioso (hackers), seguridad de acceso físico a los equipos, seguridad del software de aplicación, falta de instalación y actualizaciones de programas de seguridad

5. PLAN DE AVERSIÓN

Utilizar funciones para encriptar información.

DOCUMENTO RIESGO 5

1. IDENTIFICACIÓN

Cambio de políticas de gestión

2. TIPO

Riesgo de Negocios

3. SUBTIPO

Riesgo de gestión

4. CAUSAS

El proyecto puede sufrir retrasos, cuanto más avanzado este el desarrollo del mismo más crítico será implementar los cambios, y generación de nuevos requisitos.

5. PLAN DE AVERSIÓN

Cumplir las actividades dentro del tiempo establecido para cada una de ellas, y obtener los resultados programados en lo que se refiere al producto final.

4.5. Factibilidad

Luego de estipular los requisitos en la primera parte de la fase inicial es necesario verificar la factibilidad del proyecto.

4.5.1. Factibilidad Operativa

La Escuela de Conducción de la UNACH no posee un Sistema Informático que logre llevar de manera adecuada todos los procesos de gestión vehicular es por eso que se ha visto la necesidad de implementar el Sistema de Gestión Vehicular con la finalidad de tener un control de todo lo referente a los vehículos.

4.5.2. Factibilidad Técnica

- **Recursos Humanos**

Para el desarrollo del software intervienen las siguientes personas:

- Hugo Flores – Desarrollador
- Ing. Alfredo Colcha – Asesor de Tesis
- Lic. Jorge Silva – Director Administrativo Escuela de Conducción

Este es el equipo que llevará a cabo el desarrollo e implementación del software.

- **Recursos Hardware**

Para el desarrollo del software se utilizará un computador con las siguientes características:

- Procesador: Intel® CoreI3
- Memoria RAM: 2048 Mb
- Tarjeta de Red: 10/100 Mb
- Sistema Operativo: Windows

Para la implementación:

- Procesador: Intel CORE 2 DUO 2.3 Gb
- Memoria RAM: 1024 Mb
- Tarjeta de Red: 10/100 Mb
- Sistema Operativo: Windows XP SP3

- **Recursos Software**

Para la implementación se la aplicación web se ha utilizado las siguientes herramientas:

- Servidor Web: Apache
- Lenguaje de Programación: PHP versión 5.3
- Herramienta para diseño de páginas web: Dreamweaver CS6
- Base de datos: MySQL 5

En base a los recursos humanos, hardware, software con los que se cuenta para desarrollar el software, se puede concluir que desde el punto de vista técnico el

software para Gestión Vehicular de la Escuela de Conducción de la UNACH si es factible de realizarlo.

4.5.3. Factibilidad Económica

El desarrollador cubrirá los gastos que genere el desarrollo del software, para la implementación; sin embargo la Escuela de Conducción de la UNACH proporcionará los recursos para la implementación del mismo.

- **Análisis Costo – Beneficio**

- Programación	\$ 700
- Literatura relacionados al tema	\$ 300
- Internet	\$ 250
- Varios	\$ 250
COSTO TOTAL	\$ 1100

- **Determinación de Beneficios**

Cuando el software sea completado se podrá ver beneficios tangibles e intangibles para los usuarios finales (personal de la Escuela de Conducción de la UNACH).

- **Beneficios Tangibles**

- Gestión del mantenimiento y control de vehículos de la Escuela de Conducción de la UNACH.

- **Beneficios Intangibles**

- Automatizar todos los procesos para tener un control adecuado para el mantenimiento y control de los vehículos de la Escuela de Conducción de la UNACH.
- Manejo adecuado a la información.

4.6. Fase de Análisis

4.6.1. Jerarquía de usuarios

Se han establecido varios niveles de usuarios que han sido considerados en la fase de especificación de requisitos de software.

Estos usuarios están relacionados directamente en el funcionamiento de la aplicación web y se detallan a continuación:

- Usuario Administrador: Es el encargado de manipular información en la aplicación en lo referente a ingreso, búsqueda, eliminación de información para mantenimiento de los vehículos.
- Usuario Instructor: Es el usuario especial que utilizan el contenido y funcionalidad de la aplicación, a medida que ingresan la información del mantenimiento de los vehículos.

4.7. DISEÑO E IMPLEMENTACION DE LA APLICACIÓN

En esta etapa se definirán un conjunto de técnicas y características incluyendo las ventajas y desventajas para tener una idea amplia del software que se va a desarrollar. El producto final permitirá a los instructores de la Escuela de Conducción de la Universidad Nacional de Chimborazo que ingresen información del estado de sus vehículos.

4.7.1. Definición de la Metodología

Existen varias metodologías para el desarrollo de software, sin embargo se ha tomado en cuenta una metodología que tienda a disminuir riesgos y permita obtener un software de calidad en los plazos o tiempos establecidos. RUP (Proceso Unificado) hace posible a través de pruebas y retroalimentación que el producto a obtenerse sea de calidad y enmarcado en estándares que se ven reflejados en un software que cumple a cabalidad sus requerimientos.

4.8. Descripción General del Sistema

El sistema permitirá a los instructores de la Escuela de Conducción de la Universidad Nacional de Chimborazo que ingresen y manipulen información de los diferentes tipos de mantenimientos que se efectúa a los vehículos que utilizan para realizar las clases prácticas de conducción.

La aplicación web está desarrollada con herramientas de software libre y para que se ejecute sin inconvenientes sea en plataforma Windows en sus diferentes distribuciones como Windows XP, Windows 7 o Windows 8.

4.9. Definición del Prototipo

En esta fase se establecen los requerimientos que demanda la aplicación por parte del cliente; por tal motivo se realizó una reunión con el Director de la Escuela de Conducción de la Universidad Nacional de Chimborazo en la que se establecieron los lineamientos a los que se debe ajustar la aplicación web en lo que se refiere a usuario – aplicación.

4.10. Definición de Requisitos Generales del Sistema

4.10.1. Requisitos funcionales relacionados con Vehículos

- **Agregar vehículo**
El sistema ingresará datos de vehículos, llenando toda la información a través de un formulario.
- **Modificar vehículo**
El sistema permitirá realizar modificaciones en los datos de los vehículos utilizando como identificador el código de cada uno.
- **Eliminar vehículo**
El sistema podrá eliminar los datos de vehículos a través de su identificador.

4.10.2. Requisitos funcionales relacionados con Mantenimiento Preventivo

- **Agregar Mantenimiento Preventivo**
El sistema ingresará datos de mantenimientos preventivos efectuados a los vehículos, llenando la información a través de un formulario.
- **Modificar Mantenimiento Preventivo**

El sistema permitirá realizar modificaciones en los datos de mantenimientos preventivos utilizando como identificador el código de cada uno.

- **Eliminar Mantenimiento Preventivo**

El sistema podrá eliminar los datos de mantenimientos preventivos a través de su identificador respectivo.

4.10.3. Requisitos funcionales relacionados con Mantenimiento Correctivo

- **Agregar Mantenimiento Correctivo**

El sistema ingresará datos de mantenimientos correctivos efectuados a los vehículos, llenando la información a través de un formulario.

- **Modificar Mantenimiento Correctivo**

El sistema permitirá realizar modificaciones en los datos de mantenimientos correctivos utilizando como identificador el código de cada uno.

- **Eliminar Mantenimiento Correctivo**

El sistema podrá eliminar los datos de mantenimientos correctivos a través de su identificador respectivo.

4.10.4. Requisitos funcionales relacionados con Taller

- **Agregar Taller**

El sistema ingresará datos de los talleres, llenando toda la información a través de un formulario.

- **Modificar Taller**

El sistema permitirá realizar modificaciones en los datos de los talleres utilizando como identificador el código de cada uno.

- **Eliminar Taller**

El sistema podrá eliminar los datos de talleres a través de su identificador.

4.10.5. Requisitos funcionales relacionados con Instructores

- **Agregar Instructores**

El sistema ingresará datos de los instructores, llenando toda la información a través de un formulario.

- **Modificar Instructores**

El sistema permitirá realizar modificaciones en los datos de los instructores utilizando como identificador el código de cada uno.

- **Eliminar Instructores**

El sistema podrá eliminar los datos de instructores a través de su identificador.

4.10.6. Requisitos funcionales relacionados con Accesorios

- **Agregar Accesorios**

El sistema ingresará datos de los accesorios, llenando toda la información a través de un formulario.

- **Modificar Accesorios**

El sistema permitirá realizar modificaciones en los datos de los accesorios utilizando como identificador el código de cada uno.

- **Eliminar Accesorios**

El sistema podrá eliminar los datos de accesorios_ a través de su identificador.

4.10.7. Requisitos funcionales relacionados con Accidentes

- **Agregar Accidentes**

El sistema ingresará datos de los accidentes, llenando toda la información a través de un formulario.

- **Modificar Accidentes**

El sistema permitirá realizar modificaciones en los datos de los accidentes utilizando como identificador el código de cada uno.

- **Eliminar Accidentes**

El sistema podrá eliminar los datos de accidentes_ a través de su identificador.

4.11. Requisitos de rendimiento

4.11.1. Estáticos

El sistema está diseñado para ser instalado en una estación de trabajo (que esté conectado a la intranet de la Escuela de Conducción de la Universidad Nacional de Chimborazo) bajo un servidor web, y un motor de base de datos. El acceso dependerá del perfil que posea el usuario.

4.12. Atributos del sistema software

4.12.1. Fiabilidad

En base a los requisitos que se han establecido con las autoridades de la Escuela de Conducción se prevé que el sistema se ajustará a las necesidades descritas y realizará las acciones esperadas.

4.12.2. Disponibilidad

El sistema estará disponible para proporcionar información en cualquier momento por su diseño modular, lo cual implica que si se debe realizar alguna modificación no afectará al sistema.

4.12.3. Seguridad

La seguridad es un aspecto importante, por tal motivo se ha implementado el manejo de sesiones, así como evitar ataques por inyección de SQL.

4.12.4. Portabilidad

En esta fase no está contemplado ningún requerimiento de portabilidad para este sistema informático.

4.13. Requisitos no funcionales (Normas y Estándares)

- Mediante la metodología RUP se desarrollará el sistema.
- Las interfaces que proporciona deben ser claras e intuitivas.

4.14. Requisitos no funcionales – Seguridad

Se usarán sesiones para lo cual debe ingresar el nombre de usuario y la contraseña para ser verificados en la base de datos y según el perfil de usuario se proporcionará las opciones establecidas para cada uno.

4.15. Requisitos no funcionales – Organización

Para el desarrollo del sistema deben utilizarse herramientas apropiadas que faciliten su construcción, así como conocimiento adecuado de las mismas.

4.16. Catálogo de Usuarios

- Administrador: es el usuario que tiene el control del sistema y el encargado de administrarlo.
- Instructor: es el usuario que va a ingresar información de los mantenimientos vehiculares.

4.17. Modelo de Negocio

Esta fase hace referencia a los procesos de uno o más elementos organizacionales que se los representarán mediante los Casos de Uso.

4.18. Objetivos y Alcance del Sistema

El sistema permitirá ingresar información de los mantenimientos correctivos o preventivos de los vehículos, esta información se almacenará en una base de datos y estarán disponibles para ser manipulados o generar reportes.

4.19. Establecimiento de requisitos

4.19.1. Especificación de Casos de Uso

Los casos de uso permiten de manera detallada especificar los procesos que se deben realizar de manera prioritaria y la interacción que existe entre el software y los usuarios que están directamente relacionados para la manipulación de la información. La aplicación maneja dos módulos, uno para el administrador y otro para los Instructores que son quienes manejan los automóviles de la Escuela de Conducción de la UNACH. El módulo administrador realizará la gestión de mantenimiento; mientras que el módulo instructor solo manipulará la información que se ha ingresado a través del módulo de Administrador.

4.20. Casos de Uso

Es necesario El diagrama de casos de uso se ilustra a continuación.

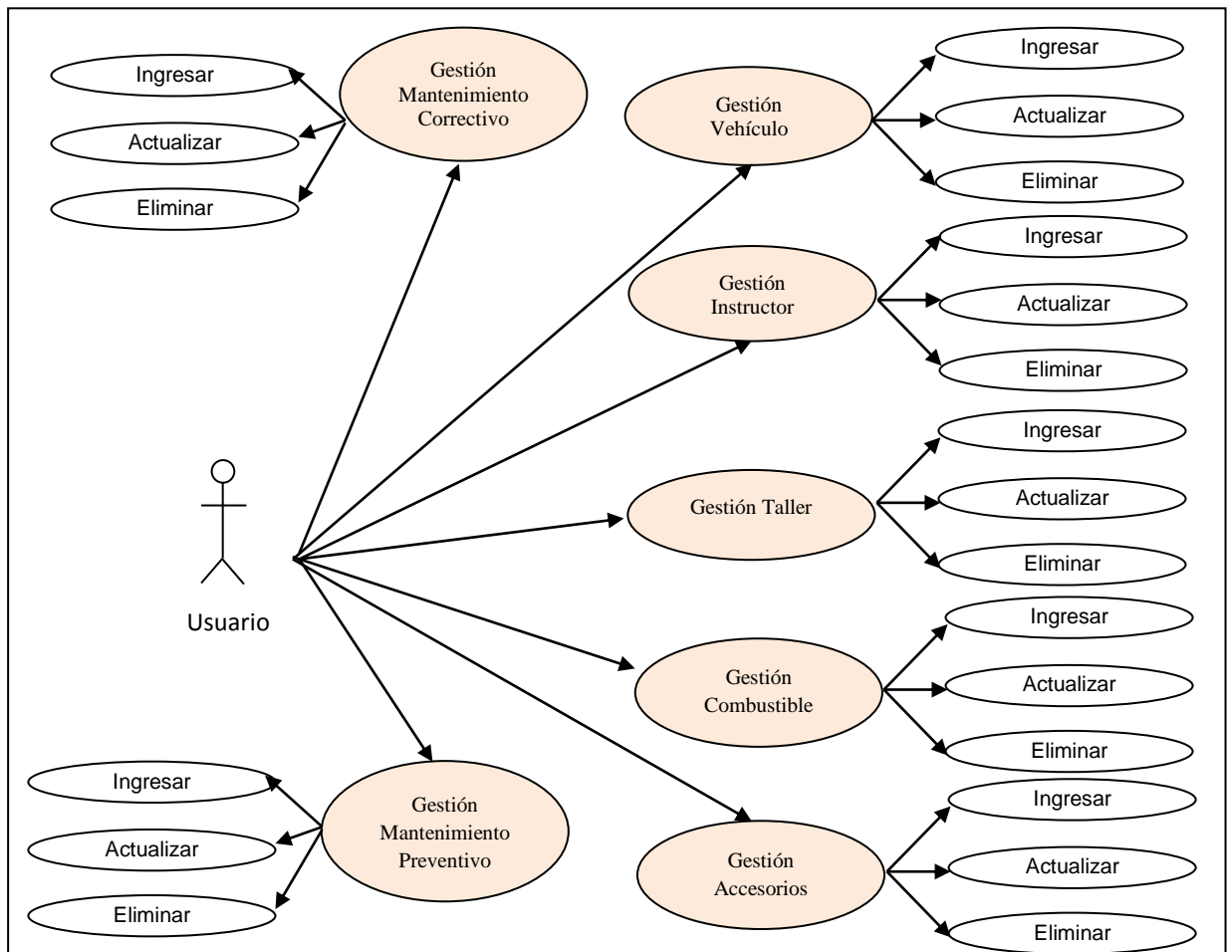


Figura 13. Diagrama de casos de uso

4.20.1. Detalle de los casos de uso

Caso de Uso (CU-1): Validación de usuario

Actores: Administrador, Instructor, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el usuario pueda ingresar su usuario y clave para ser validados con la base de datos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario ingresa su identificador de usuario y la clave respectiva.	2. La aplicación valida los datos proporcionados con los que están almacenados en la base de datos. 2.1. Si los datos son correctos le proporciona una interfaz para que interactúe con los menús. 2.2. Si los datos son incorrectos le presenta un mensaje de error en la validación.

Caso de Uso (CU-2): Ingresar Vehículo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el usuario pueda ingresar y registrar los datos de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú VEHICULO en la opción Nuevo	2. La aplicación le proporciona una interfaz para que ingrese la información
3. El actor ingresa: Número, Tipo, Marca, Modelo, Año de Fabricación, Color, Número de chasis, Número motor, Peso, Cilindraje, Número de Ocupantes, Placa, Transmisión, Ejes,	4. El sistema almacena la información ingresada.

Neumáticos, Estado, Matriculado, Tipo combustible	
--	--

Caso de Uso (CU-3): Actualiza Vehículo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el usuario de la aplicación pueda actualizar los datos de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario ingresa al menú VEHICULO, en la opción Actualizar	2. La aplicación le proporciona una interfaz con un listado de los vehículos almacenados en la base de datos
3. El usuario da clic en el vehículo que desea modificar la información	4. El sistema le proporciona una interfaz con los datos correspondientes al vehículo seleccionado.
5. El actor Administrador actualiza la siguiente información: Número, Tipo, Marca, Modelo, Año de Fabricación, Color, Número de chasis, Número motor, Peso, Cilindraje, Número de Ocupantes, Placa, Transmisión, Ejes, Neumáticos, Estado, Matriculado, Tipo combustible .	6. El sistema almacena la información modificada.

Caso de Uso (CU-4): Elimina Vehículo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú VEHICULO, en la opción Eliminar	2. La aplicación le proporciona una interfaz para que elimine la información
3. El usuario da clic en el vehículo que desea modificar la información	4. El sistema elimina la información de acuerdo al código del auto seleccionado.

Caso de Uso (CU-5): Ingresar Instructor

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar los datos de los instructores.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a la opción INSTRUCTOR opción Nuevo	2. La aplicación le proporciona una interfaz para que ingrese la información
3. El actor Administrador ingresa: Código, Nombres, Apellidos, Cédula, Dirección domicilio, Tipo licencia, Número teléfono, Nivel instrucción, Correo	4. El sistema almacena la información ingresada.

Caso de Uso (CU-6): Actualiza Instructor

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar los datos de los instructores.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú INSTRUCTORES, opción Actualizar	2. La aplicación le proporciona una interfaz con un listado de los instructores almacenados en la base de datos

3. El usuario da clic en el instructor que desea modificar la información	4. El sistema le proporciona una interfaz con los datos correspondientes al instructor seleccionado.
5. El actor Administrador actualiza la siguiente información: Código, Nombres, Apellidos, Cédula, Dirección domicilio, Tipo licencia, Número teléfono, Nivel instrucción, Correo	6. El sistema almacena la información modificada.

Caso de Uso (CU-7): Elimina Instructor

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los datos de los instructores.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú INSTRUCTOR, opción Eliminar	2. La aplicación le proporciona una interfaz con un listado de los instructores almacenados en la base de datos
3. El usuario da clic en el instructor que desea eliminar la información	4. El sistema registra la eliminación

Caso de Uso (CU-8): Ingresar Taller

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar los datos del taller.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a la opción TALLER opción Nuevo	2. La aplicación le proporciona una interfaz para que ingrese la

	información
3. El actor Administrador ingresa: Nombre, Propietario, Dirección, Teléfono, Dirección, Ciudad, Correo	4. El sistema almacena la información ingresada.

Caso de Uso (CU-9): Actualiza Taller

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar los datos de los instructores.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú TALLER, opción Actualizar	2. La aplicación le proporciona una interfaz con un listado de los talleres almacenados en la base de datos
3. El usuario da clic en el taller que desea modificar la información	4. El sistema le proporciona una interfaz con los datos correspondientes al taller seleccionado.
5. El actor Administrador actualiza la siguiente información: Nombre, Propietario, Dirección, Teléfono, Dirección, Ciudad, Correo	6. El sistema almacena la información modificada.

Caso de Uso (CU-10): Elimina Taller

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los datos del taller.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú TALLER, opción Eliminar	2. La aplicación le proporciona una interfaz con un listado de los talleres almacenados en la base de datos

3. El usuario da clic en el taller que desea eliminar la información	4. El sistema registra la eliminación
--	---------------------------------------

Caso de Uso (CU-11): Ingresar Mantenimiento Preventivo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar el mantenimiento de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a la opción MANTENIMIENTO PREVENTIVO opción Nuevo	2. La aplicación le proporciona una interfaz para que ingrese la información
3. El actor Administrador ingresa: Tipo de mantenimiento, Programa de mantenimiento, Fecha, Vehículo, Placa, Kilometraje, Taller, Precio, Número de factura, Próximo Mantenimiento	4. El sistema almacena la información ingresada.

Caso de Uso (CU-12): Actualiza Mantenimiento Preventivo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar el mantenimiento de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú MANTENIMIENTO PREVENTIVO, opción Actualizar	2. La aplicación le proporciona una interfaz con un listado de los mantenimientos preventivos almacenados en la base de datos
3. El usuario da clic en el mantenimiento que desea modificar la información	4. El sistema le proporciona una interfaz con los datos correspondientes al mantenimiento

	seleccionado.
5. El actor Administrador actualiza la siguiente información: Tipo de mantenimiento, Programa de mantenimiento, Fecha, Vehículo, Placa, Kilometraje, Taller, Precio, Número de factura, Próximo Mantenimiento	6. El sistema almacena la información modificada.

Caso de Uso (CU-13): Elimina Mantenimiento Preventivo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los mantenimientos de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú MANTENIMIENTO, opción Eliminar	2. La aplicación le proporciona una interfaz con un listado de los mantenimientos preventivos almacenados en la base de datos
3. El usuario da clic en el mantenimiento que desea eliminar la información	4. El sistema registra la eliminación

Caso de Uso (CU-14): Ingresar Mantenimiento Correctivo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar el mantenimiento de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a la opción MANTENIMIENTO CORRECTIVO opción Nuevo	2. La aplicación le proporciona una interfaz para que ingrese la información

3. El actor Administrador ingresa: Fecha, Detalle, Precio, Factura/Proforma, Kilometraje, Vehículo, Placa, Próximo mantenimiento, Taller	4. El sistema almacena la información ingresada.
--	--

Caso de Uso (CU-15): Actualiza Mantenimiento Correctivo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar el mantenimiento de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú MANTENIMIENTO CORRECTIVO, opción Actualizar	2. La aplicación le proporciona una interfaz con un listado de los mantenimientos correctivos almacenados en la base de datos
3. El usuario da clic en el mantenimiento que desea modificar la información	4. El sistema le proporciona una interfaz con los datos correspondientes al mantenimiento seleccionado.
5. El actor Administrador actualiza la siguiente información: Fecha, Detalle, Precio, Factura/Proforma, Kilometraje, Vehículo, Placa, Próximo mantenimiento, Taller	6. El sistema almacena la información modificada.

Caso de Uso (CU-16): Elimina Mantenimiento Correctivo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los mantenimientos de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú MANTENIMIENTO, opción Eliminar	2. La aplicación le proporciona una interfaz con un listado de los mantenimientos correctivos almacenados en la base de datos
3. El usuario da clic en el mantenimiento que desea eliminar la información	4. El sistema registra la eliminación

Caso de Uso (CU-17): Ingresar Combustible

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar la tanqueada de combustible de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a la opción COMBUSTIBLE opción Nuevo	2. La aplicación le proporciona una interfaz para que ingrese la información
3. El actor Administrador ingresa: Tipo, Fecha, Vehículo, Placa, Kilometraje, Cantidad Galones, Valor Total	4. El sistema almacena la información ingresada.

Caso de Uso (CU-18): Actualiza Mantenimiento Correctivo

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar la tranqueada de combustible de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú COMBUSTIBLE, opción Actualizar	2. La aplicación le proporciona una interfaz con un listado de los registros de combustible

	almacenados en la base de datos
3. El usuario da clic en el combustible que desea modificar la información	4. El sistema le proporciona una interfaz con los datos correspondientes al registro de combustible seleccionado.
5. El actor Administrador actualiza la siguiente información: Tipo, Fecha, Vehículo, Placa, Kilometraje, Cantidad Galones, Valor Total	6. El sistema almacena la información modificada.

Caso de Uso (CU-19): Elimina Combustible

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja la tanqueada de combustible de los mantenimientos de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú COMBUSTIBLE, opción Eliminar	2. La aplicación le proporciona una interfaz con un listado de las tanqueadas de combustibles almacenados en la base de datos
3. El usuario da clic en el mantenimiento que desea eliminar la información	4. El sistema registra la eliminación

Caso de Uso (CU-20): Ingresar Accesorio

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar accesorios de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a la opción ACCESORIO opción Nuevo	2. La aplicación le proporciona una interfaz para que ingrese la información
3. El actor Administrador ingresa: Fecha, Nombre del accesorio, Estado, Precio, Factura/Proforma, Vehículo, Kilometraje, Placa, Taller	4. El sistema almacena la información ingresada.

Caso de Uso (CU-21): Actualiza Accesorio

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar los accesorios de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú ACCESORIO, opción Actualizar	2. La aplicación le proporciona una interfaz con un listado de los accesorios almacenados en la base de datos
3. El usuario da clic en el accesorio que desea modificar la información	4. El sistema le proporciona una interfaz con los datos correspondientes al registro de combustible seleccionado.
5. El actor Administrador actualiza la siguiente información: Fecha, Nombre del accesorio, Estado, Precio, Factura/Proforma, Vehículo, Kilometraje, Placa, Taller	6. El sistema almacena la información modificada.

Caso de Uso (CU-22): Elimina Accesorio

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los accesorios de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa al menú ACCESORIO, opción Eliminar	2. La aplicación le proporciona una interfaz con un listado de los accesorios almacenados en la base de datos
3. El usuario da clic en el accesorio que desea eliminar la información	4. El sistema registra la eliminación

Caso de Uso (CU-23): Ingresar Accidente

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar accidentes de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a la opción ACCIDENTE opción Nuevo	2. La aplicación le proporciona una interfaz para que ingrese la información
3. El actor Administrador ingresa: Fecha, Detalle, Ubicación, Precio, Factura, Cubierto por aseguradora, Taller, Vehículo, Placa, Kilometraje	4. El sistema almacena la información ingresada.

Caso de Uso (CU-24): Actualiza Accidente

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar los accidentes de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a al menú ACCIDENTE, opción Actualizar	2. La aplicación le proporciona una interfaz con un listado de los accidentes almacenados en la base

	de datos
3. El usuario da clic en el accesorio que desea modificar la información	4. El sistema le proporciona una interfaz con los datos correspondientes al registro de combustible seleccionado.
5. El actor Administrador actualiza la siguiente información: Fecha, Detalle, Ubicación, Precio, Factura, Cubierto por aseguradora, Taller, Vehículo, Placa, Kilometraje	6. El sistema almacena la información modificada.

Caso de Uso (CU-25): Elimina Accidente

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los accidentes de los vehículos.

Tipo: Primario.

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa al menú ACCIDENTE, opción Eliminar	2. La aplicación le proporciona una interfaz con un listado de los accidentes almacenados en la base de datos
3. El usuario da clic en el accidente que desea eliminar la información	4. El sistema registra la eliminación

4.21. Modelo Conceptual

Las clases identificadas son las siguientes:

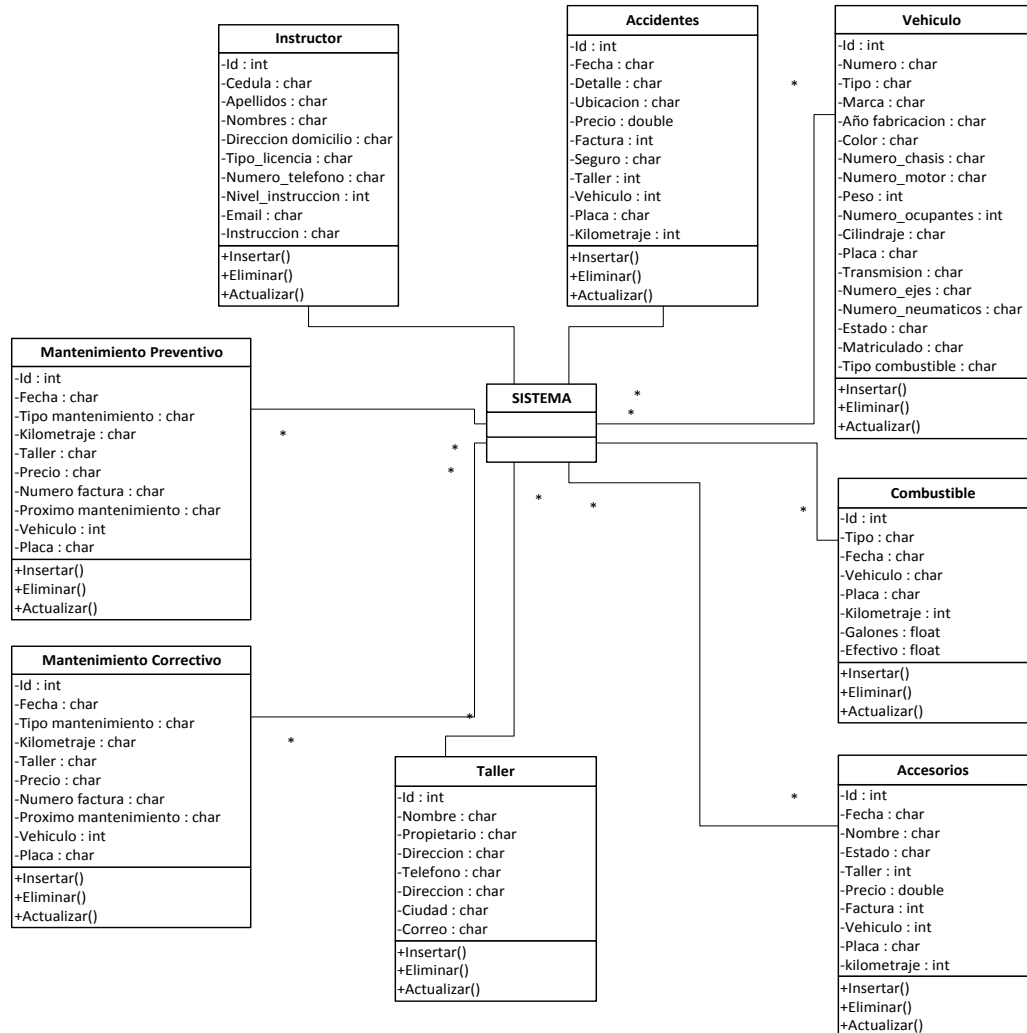


Figura 14. Clases de la base de datos

4.22. Secuencia de Actividades entre actores

El diagrama de secuencia muestra todos los eventos generados por actores externos, su orden y los eventos internos del sistema y que fluyen de los actores hacia el sistema.

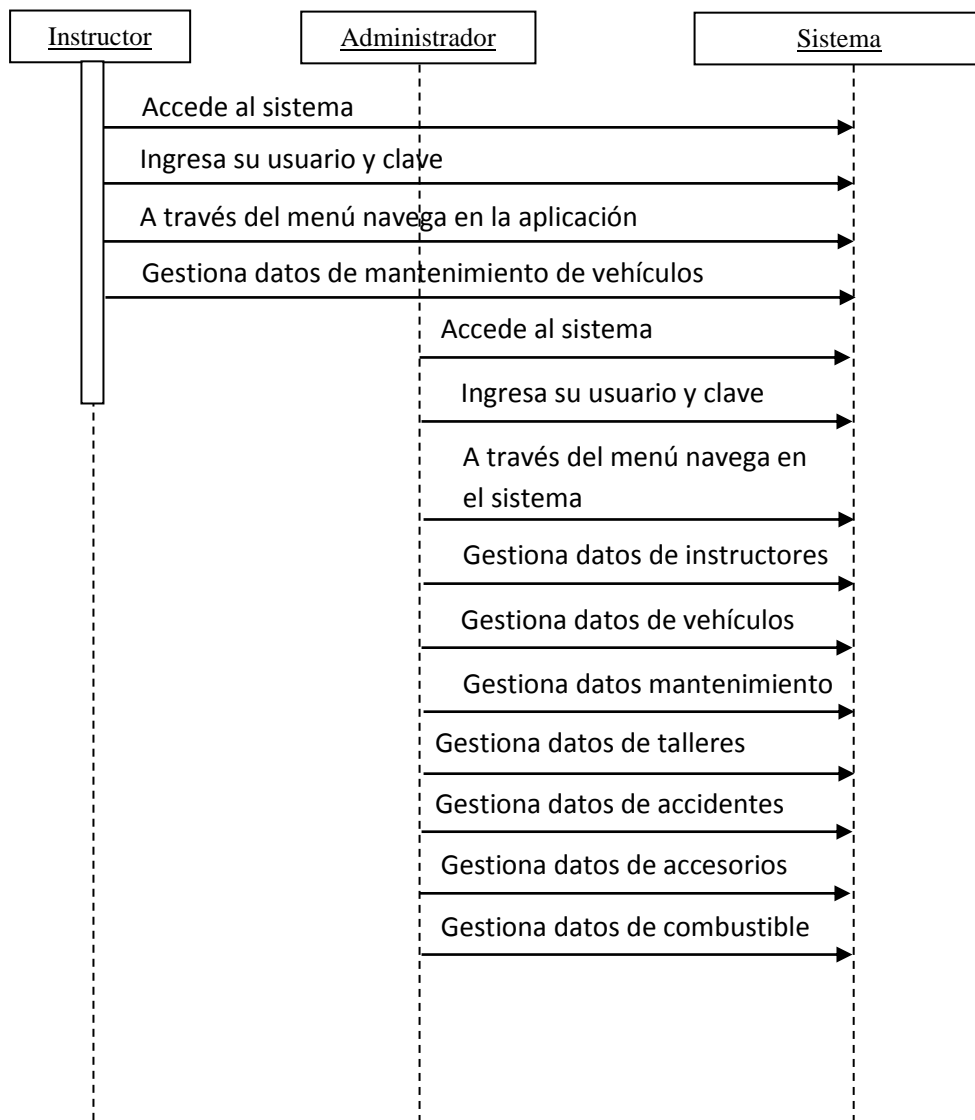


Figura 15. Secuencia de actividades entre actores y el sistema

4.23. Definición de la Interfaz

Mediante formularios y a través de un lenguaje de programación estandarizado para obtener un rendimiento óptimo en la información que interactúa entre los usuarios y el sistema, por lo que se puede aseverar que el rendimiento del sistema no dependerá de la arquitectura en la que ésta se implemente sino de los navegadores web y el sistema operativo desde donde se acceda.

Figura 16. Interfaz para validación de usuarios

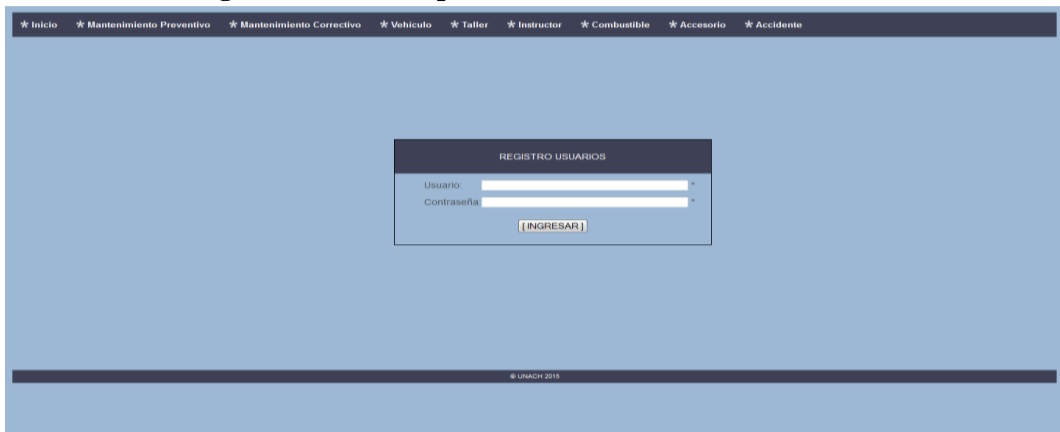


Figura 17. Interfaz con menú principal



Figura 18. Interfaz para registrar un nuevo mantenimiento

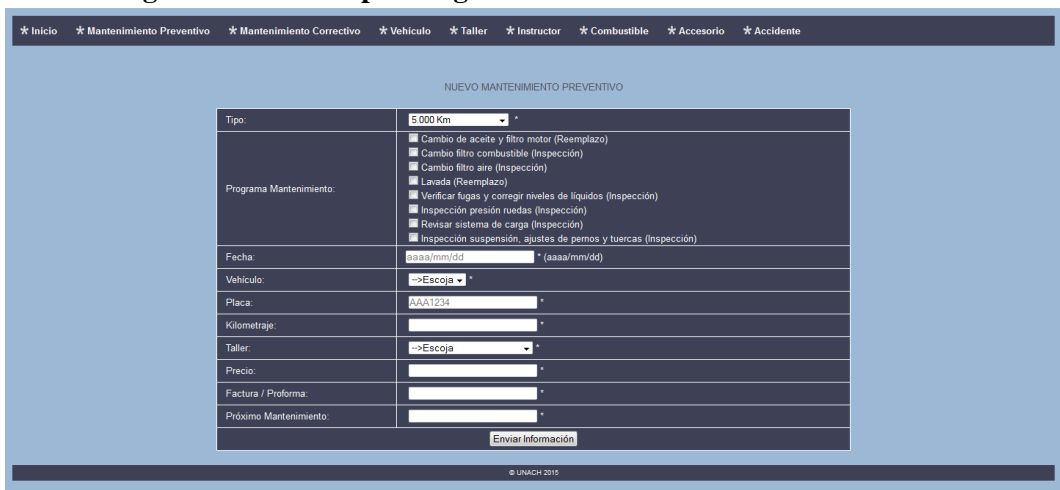


Figura 19. Interfaz para eliminar un mantenimiento

[Inicio](#)
[Mantenimiento Preventivo](#)
[Mantenimiento Correctivo](#)
[Vehículo](#)
[Taller](#)
[Instructor](#)
[Combustible](#)
[Accesorio](#)
[Accidente](#)

BIENVENIDO: 1 SALIR

ELIMINAR MANTENIMIENTO PREVENTIVO

FECHA	TIPO	VEHICULO	TALLER	FACTURA	
2012-01-11	5000 km	HEI1142	TALLERES AMERICANO	1233211234	Eliminar
2012-01-10	5000 km	HEI1136	LLANTA SIERRA S.A	1234567890	Eliminar
2012-01-10	5000 km	HEI1144	LLANTA SIERRA S.A	1234567890	Eliminar
2012-01-04	5000 km	HEI1140	TALLERES AMERICANO	12345678	Eliminar
2012-07-02	10000 km	HEI1136	TALLERES AMERICANO	123456789	Eliminar
2012-07-02	10000 km	HEI1142	LLANTA SIERRA S.A	12345	Eliminar
2012-07-04	10000 km	HEI1139	LLANTA SIERRA S.A	12347	Eliminar
2012-07-05	10000 km	HEI1144	LLANTA SIERRA S.A	12344	Eliminar
2012-07-05	10000 km	HEI1140	LLANTA SIERRA S.A	12345	Eliminar
2014-12-01	25000 km	HEI1136	LLANTA SIERRA S.A	1234780	Eliminar
2014-12-01	25000 km	HEI1142	LLANTA SIERRA S.A	123908	Eliminar
2014-12-02	25000 km	HEI1139	LLANTA SIERRA S.A	123657	Eliminar
2014-12-04	25000 km	HEI1144	LLANTA SIERRA S.A	345612	Eliminar
2014-12-05	25000 km	HEI1140	LLANTA SIERRA S.A	6780341	Eliminar

© UNACH 2015

Figura 20. Interfaz para modificar un mantenimiento

[Inicio](#)
[Mantenimiento Preventivo](#)
[Mantenimiento Correctivo](#)
[Vehículo](#)
[Taller](#)
[Instructor](#)
[Combustible](#)
[Accesorio](#)
[Accidente](#)

BIENVENIDO: 1 SALIR

ACTUALIZAR MANTENIMIENTO CORRECTIVO

FECHA:	2014-12-03
DETALLE:	REPARACION número DEL MOTO
TALLER:	TALLERES AMERICANO
VEHICULO:	2
PLACA:	HEI1142
KILOMETRAJE:	26567
PRECIO:	\$ 765.34
FACTURA:	1234567
PROXIMO MANTENIMIENTO:	NO HAY FECHA NI KILOMETRAJE

ACTUALIZAR

© UNACH 2015

Figura 21. Interfaz para buscar un mantenimiento

[Inicio](#)
[Mantenimiento Preventivo](#)
[Mantenimiento Correctivo](#)
[Vehículo](#)
[Taller](#)
[Instructor](#)
[Combustible](#)
[Accesorio](#)
[Accidente](#)

BIENVENIDO: 1 SALIR

BUSCAR MANTENIMIENTO PREVENTIVO

PLACA HEI1142

FECHA	TIPO	VEHICULO	PLACA	KILOMETRAJE	TALLER	PRECIO	FACTURA	PROX. MANTEN.	
2012-01-11	5000	2	HEI1142	5019 km	TALLERES AMERICANO	\$ 78.00	1233211234	10000 km	Eliminar
2012-07-02	10000	2	HEI1142	10123 km	LLANTA SIERRA S.A	\$ 123.45	12345	15000 km	Eliminar
2014-12-01	25000	2	HEI1142	25003 km	LLANTA SIERRA S.A	\$ 54.45	123908	30000 km	Eliminar

© UNACH 2015

Figura 22. Interfaz para generar reporte de un mantenimiento

ID	FECHA	TIPO	MANTENIMIENTO	VEHICULO	PLACA	KILOMETRAJE	TALLER	EFECTIVO	FACTURA	PUNTO
1	2012-01-11	5000 km	Revisión de aceite y filtros	2	HEI 112	2019 km	TALLER ABERCANO	\$ 12.00	03211231	10000 km
2	2012-01-10	8000 km	Cambio de aceite y filtros	1	HEI 120	5000 km	LLANTA DEPPA S.A	\$ 90.45	123456789	10000 km
3	2012-01-10	8000 km	Cambio de aceite y filtros	1	HEI 114	5 600 km	LLANTA DEPPA S.A	\$ 90.09	123456789	10000 km
4	2012-01-04	8000 km	Cambio de aceite y filtros	3	HEI 140	5 013 km	TALLER ABERCANO	\$ 76.24	12345678	10000 km
5	2012-01-02	10000 km	Cambio de aceite y filtros	1	HEI 130	6000 km	TALLER ABERCANO	\$ 128.24	123456789	10000 km
6	2012-01-02	10000 km	Cambio de aceite y filtros	2	HEI 142	6123 km	LLANTA DEPPA S.A	\$ 128.48	12345	10000 km
7	2012-01-04	10000 km	Cambio de aceite y filtros	3	HEI 139	6000 km	LLANTA DEPPA S.A	\$ 97.01	12345	10000 km

4.2.4. Diseño arquitectónico

El entorno WEB propuesto posee seguridad a nivel de arquitectura de aplicación como se puede ver en la figura a continuación:

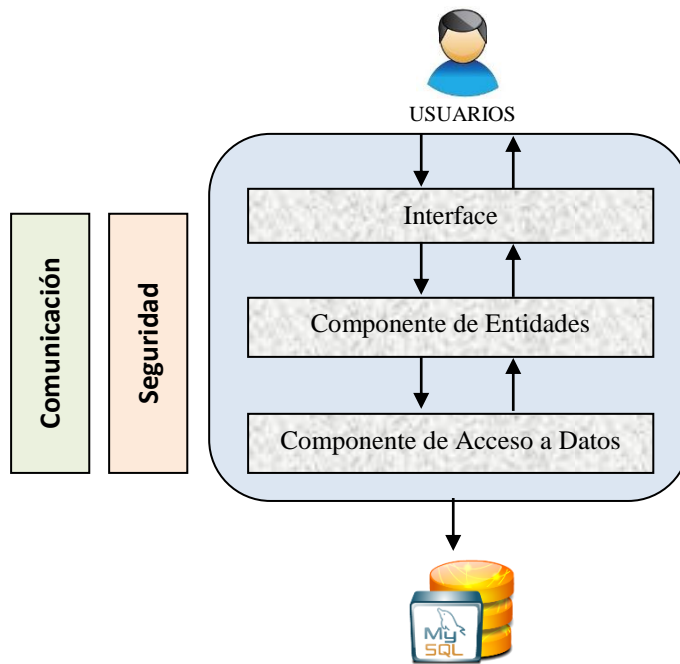


Figura 23. Arquitectura del sistema.

La comunicación entre componentes se realiza siguiendo un esquema de protección o seguridad de acceso por concepción natural de la aplicación y en cuanto se refiere a interfaz, está diseñada para facilitar la utilidad por parte del usuario pero también para evitar que se viole la integridad del sistema mediante la

introducción de código malicioso sean estos exploits, SQL Injections, entre otros. El uso de sesiones también contribuye a un aumento considerable de medidas de prevención y seguridad.

4.25. Diseño de la base de datos

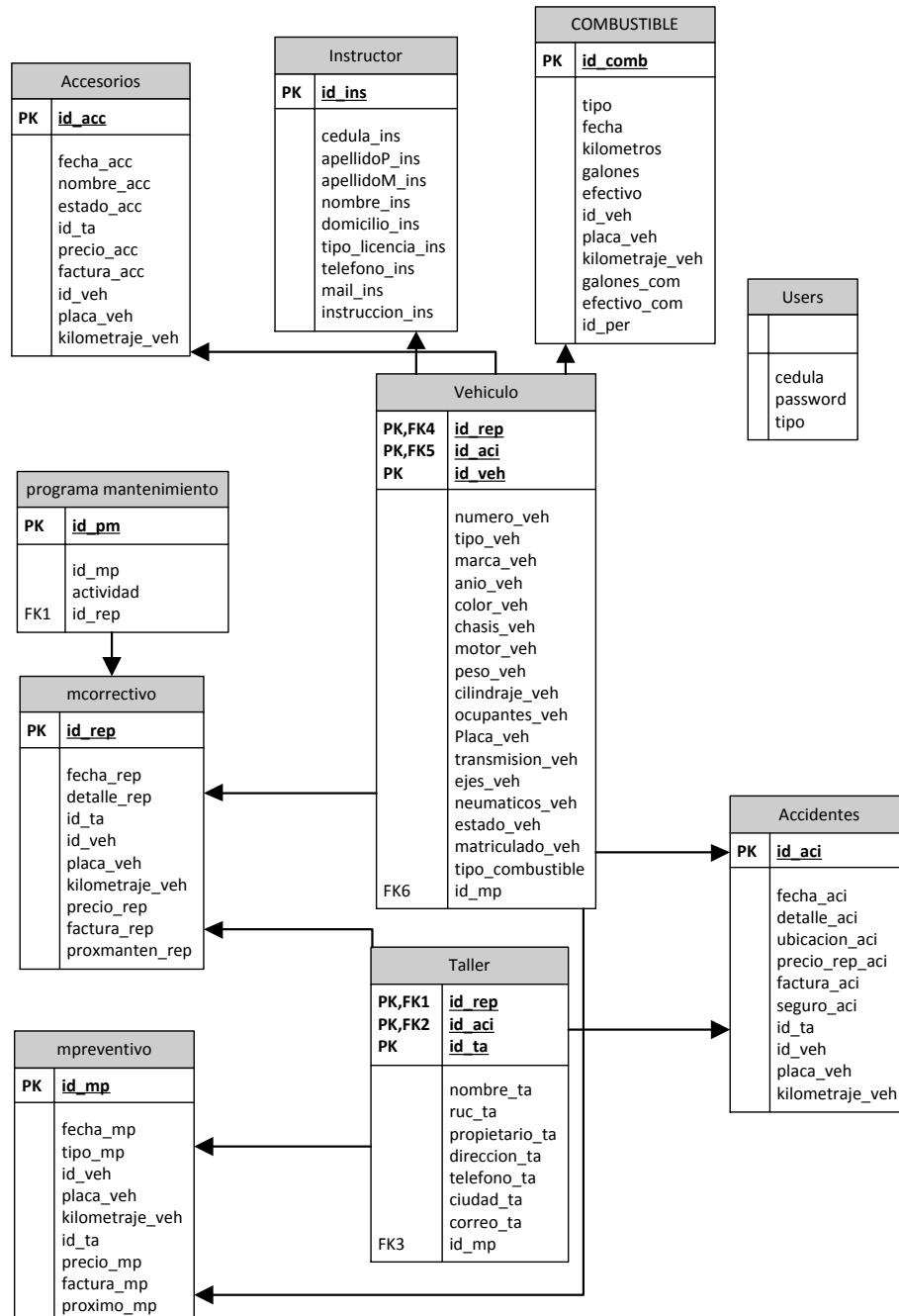


Figura 24. Estructura de la base de datos

4.26. Diseño de procedimientos

4.26.1. Clase Sistema

```
class Sistema extends CI_Controller
{
    var $fecha_acc;
    var $nombre_acc;
    var $estado_acc;
    var $precio_acc;
    var $factura_acc;
    var $cedula_ins;
    var $apellidoP_ins;
    var $apellidoM_ins;
    var $nombre_ins;
    var $domicilio_ins;
    var $tipo_licencia_ins;
    var $telefono_ins;
    var $mail_ins;
    var $instruccion_ins;
    var $tipo_com;
    var $fecha_com;
    var $kilometros_com;
    var $galones_com;
    var $efectivo_com;
    var $id_mp;
    var $actividad;
    var $numero_veh;
    var $tipo_veh;
    var $marca_veh;
    var $anio_veh;
    var $color_veh;
    var $chasis_veh;
    var $motor_veh;
    var $peso_veh;
    var $cilindraje_veh;
    var $ocupantes_veh;
    var $placa_veh;
    var $transmision_veh;
    var $ejes_veh;
    var $neumaticos_veh;
    var $estado_veh;
    var $matriculado_veh;
    var $kilometraje_veh;
    var $tipo_combustible;
    var $nombre_ta;
    var $ruc_ta;
    var $propietario_ta;
    var $direccion_ta;
    var $telefono_ta;
```

```
var $ciudad_ta;  
var $correo_ta;  
var $fecha_aci;  
var $detalle_aci;  
var $ubicacion_aci;  
var $precio_aci;  
var $factura_aci;  
var $seguro_aci;  
var $fecha_rep;  
var $detalle_rep;  
var $precio_rep;  
var $factura_rep;  
var $proxmaten_rep;  
var $fecha_mp;  
var $tipo_mp;  
var $precio_mp;  
var $factura_mp;  
var $proximo_mp;
```

```
function constructor()  
{ parent::__construct(); }
```

```
function insertar_accesorio();  
function eliminar_accesorio();  
function actualizar_accesorio();  
function buscar_accesorio();
```

```
function insertar_taller();  
function eliminar_taller ();  
function actualizar_taller ();  
function buscar_taller ();
```

```
function insertar_vehiculo ();  
function eliminar_vehiculo ();  
function actualizar_vehiculo ();  
function buscar_vehiculo ();
```

```
function insertar_instructor();  
function eliminar_instructor ();  
function actualizar_instructor ();  
function buscar_instructor ();
```

```
function insertar_combustible();  
function eliminar_combustible ();  
function actualizar_combustible ();  
function buscar_combustible ();
```

```
function insertar_accidente();  
function eliminar_accidente ();  
function actualizar_accidente ();
```

```

function buscar_accidente ();

function insertar_mantenimiento_preventivo ();
function eliminar_mantenimiento_preventivo ();
function actualizar_mantenimiento_preventivo ();
function buscar_mantenimiento_preventivo ();

function insertar_mantenimiento_correctivo ();
function eliminar_mantenimiento_correctivo ();
function actualizar_mantenimiento_correctivo ();
function buscar_mantenimiento_correctivo ();

}

```

4.26.2. Crear Mantenimiento Correctivo

4.26.2.1. Controlador

```

class Reparacion_nuevo extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('reparacion_nuevo_model');
    }
    function index()
    {
        $this->load->view('header_view');
        $data['dataveh'] = $this->reparacion_nuevo_model->getvehiculos();
        $data['datat'] = $this->reparacion_nuevo_model->gettalleres();
        $this->load->view('reparacion_nuevo_view', $data);
        $this->load->view('footer_view');
    }
    public function logout_user()
    {
        $this->session->sess_destroy();
        $this->load->view('header_view');
        $this->load->view('welcome_message');
        $this->load->view('footer_view');
    }

    //funcion para procesar el formulario
    function insertar_reparacion()
    {
        //si se ha pulsado el botón submit validamos el formulario con codeIgniter
        if($this->input->post('submit'))
        {
            //hacemos las comprobaciones que deseemos en nuestro
            formulario

```

```

        $this->form_validation-
>set_rules('fecha','fecha','trim|required|xss_clean');
        $this->form_validation-
>set_rules('detalle','detalle','trim|required|xss_clean');
        $this->form_validation-
>set_rules('precio','precio','trim|required|xss_clean');
        $this->form_validation-
>set_rules('factura','factura','trim|required|xss_clean');

        $this->form_validation-
>set_rules('vehiculo','vehiculo','trim|required|xss_clean');
        $this->form_validation-
>set_rules('placa','placa','trim|required|xss_clean');
        $this->form_validation-
>set_rules('kilometraje','kilometraje','trim|required|xss_clean');
        $this->form_validation-
>set_rules('taller','taller','trim|required|xss_clean');

        $this->form_validation->set_message('required', 'Campo %s es
obligatorio');

        if (!$this->form_validation->run())
        {
            //si no pasamos la validación volvemos al formulario
mostrando los errores
            $this->index();
        }
        //si pasamos la validación correctamente pasamos a hacer la
inserción en la base de datos
        else
        {

            $fecha = $this->input->post('fecha');
            $detalle = $this->input->post('detalle');
            $taller = $this->input->post('taller');

            $vehiculo = $this->input->post('vehiculo');
            $placa = $this->input->post('placa');
            $kilometraje = $this->input->post('kilometraje');
            $precio = $this->input->post('precio');
            $factura = $this->input->post('factura');
            $mantenimiento = $this->input->post('mantenimiento');

            //ahora procesamos los datos hacía el modelo que
debemos crear

            $nueva_insercion = $this->reparacion_nuevo_model-
>nueva_reparacion(
                $fecha,
                $detalle,

```

```

                $staller,
                $vehiculo,
                $placa,
                $kilometraje,
                $precio,
                $factura,
                $mantenimiento
            );
            //redirect(base_url("index.php/repuracion_nuevo"),
"refresh");
            redirect(base_url("index.php/"), "refresh");
        }
    }
}
}

```

4.26.2.2. Vista

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ingresar Reparacion</title>
<link rel="stylesheet" href="http://localhost/vehiculo/scripts/tablas.css">
</head>
<body>
    <? if($this->session->userdata('usuario')!="")
        {?>
    <?php echo form_open(base_url('index.php/repuracion_nuevo/insertar_reparacion'));
    //aqui se procesará nuestro formulario, controlador comentarios, función
    insertar_instructor
    //creamos los arrays que compondran nuestro formulario

    $fecha = array(
        'name' => 'fecha',
        'id' => 'fecha',
        'size' => '10',
        'style' => 'width:50%',
            'placeholder'=>'aaaa/mm/dd',
        'value' => set_value('fecha') // con esto al procesar el formulario de forma
    incorrecta quedará guardado el dato que le habíamos puesto
    );

    $detalle = array(
        'name' => 'detalle',
        'id' => 'detalle',
        'size' => '50',
        'style' => 'width:50%',

```

```
'value' => set_value('detalle') // con esto al procesar el formulario de forma incorrecta quedará guardado el dato que le habíamos puesto
);
```

```
$taller = array(
  'name' => 'taller',
  'id' => 'taller',
  'size' => '50',
  'style' => 'width:50%',
  'value' => set_value('taller') // con esto al procesar el formulario de forma incorrecta quedará guardado el dato que le habíamos puesto
);
```

```
$precio = array(
  'name' => 'precio',
  'id' => 'precio',
  'size' => '50',
  'style' => 'width:50%',
  'value' => set_value('precio') // con esto al procesar el formulario de forma incorrecta quedará guardado el dato que le habíamos puesto
);
```

```
$factura = array(
  'name' => 'factura',
  'id' => 'factura',
  'size' => '50',
  'style' => 'width:50%',
  'value' => set_value('factura') // con esto al procesar el formulario de forma incorrecta quedará guardado el dato que le habíamos puesto
);
```

```
$vehiculo = array(
  'name' => 'vehiculo',
  'id' => 'vehiculo',
  'size' => '50',
  'style' => 'width:50%',
  'value' => set_value('vehiculo') // con esto al procesar el formulario de forma incorrecta quedará guardado el dato que le habíamos puesto
);
```

```
$mantenimiento = array(
  'name' => 'mantenimiento',
  'id' => 'mantenimiento',
  'size' => '50',
  'style' => 'width:50%',
  'value' => set_value('mantenimiento') // con esto al procesar el formulario de forma incorrecta quedará guardado el dato que le habíamos puesto
);
```

```
$placa = array(
```



```

}
//creamos la funcion nuevo comentario que será la que haga la inserción a la base
//de datos pasándole los datos a introducir en forma de array, siempre al estilo ci
function getvehiculos() {
    $this->load->database();
    $datav = array();
    $query = $this->db->get('vehiculos');
    if ($query->num_rows() > 0) {
        foreach ($query->result_array() as $row){
            $datav[] = $row;
        }
    }
    $query->free_result();
    return $datav;
}

function gettalleres() {
    $this->load->database();
    $datat = array();
    $query = $this->db->get('talleres');
    if ($query->num_rows() > 0) {
        foreach ($query->result_array() as $row){
            $datat[] = $row;
        }
    }
    $query->free_result();
    return $datat;
}

function
nuevo_mantenimiento($fecha,$tipo,$vehiculo,$placa,$kilometraje,$staller,$precio,$fact
ura,$proximo)
{
    $data = array(
        'id_mp' => "",
        'fecha_mp' => $fecha,
        'tipo_mp' => $tipo,
        'id_veh' => $vehiculo,
        'placa_veh' => $placa,
        'kilometraje_veh' => $kilometraje,
        'id_ta' => $staller,
        'precio_mp' => $precio,
        'factura_mp' => $factura,
        'proximo_mp' => $proximo,
    );
    //aqui se realiza la inserción, si queremos devolver algo deberíamos usar delante
return
    //y en el controlador despues de $nueva_insercion poner lo que queremos hacer,
redirigir,

```

```

        $this->db->insert('mpreventivo',$data);
    }
}

```

4.26.3. Eliminar Mantenimiento Correctivo

4.26.3.1. Controlador

```

class Reparacion_eliminar extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('reparacion_eliminar_model');
    }

    function index()
    {
        $this->load->view('header_view');
        //con este array enviamos los datos a mostrar a la vista
        $data['data'] = $this->reparacion_eliminar_model->getReparaciones();
        $this->load->view('reparacion_eliminar_view', $data);
        $this->load->view('footer_view');
    }

    public function logout_user()
    {
        $this->session->sess_destroy();
        $this->load->view('header_view');
        $this->load->view('welcome_message');
        $this->load->view('footer_view');
    }

    //función para eliminar una reparación dependiendo del id
    function borrar_reparacion()
    {
        $id = $this->uri->segment(3);
        $delete = $this->reparacion_eliminar_model->eliminar_reparacion($id);
        redirect(base_url().'index.php/reparacion_eliminar');
    }
}

```

4.26.3.2. Vista

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Eliminar Reparación</title>
<link rel="stylesheet" href="http://localhost/vehiculo/scripts/tablas.css">
<script type="text/javascript">

```

```

function confirma(){
    if (confirm("¿Realmente desea eliminarlo?")){
        alert("El registro ha sido eliminado.") }
        else {
            return false
        }
    }
}
</script>
</head>
<body>
<? if($this->session->userdata('usuario')!="")
    {?>
<div align="center"><BR><BR>ELIMINAR MANTENIMIENTO
CORRECTIVO<BR><BR></div>
<table width="70%" cellspacing="10" cellpadding="10">
    <tr>
        <th align="left">&nbsp;&nbsp;&nbsp;FECHA</th>
        <th align="left">DETALLE</th>
            <th align="left">FACTURA</th>
        <th align="left">PRECIO</th>
            <th align="left">&nbsp;&nbsp;&nbsp;</th>
    </tr>
    <?php
    if ($datar)
    {
        // $cont=1;
        foreach($datar as $reparacion)
        {
            // $color = (($cont % 2) == 0) ? '#dae5fd' : '#b9cdfa'; ?>
            <tr style="background-color:#OFF">
                <td>&nbsp;&nbsp;&nbsp;&nbsp;<?=$reparacion->fecha_rep?></td>
                <td>&nbsp;&nbsp;&nbsp;&nbsp;<?=$reparacion->detalle_rep?></td>
                <td>&nbsp;&nbsp;&nbsp;&nbsp;<?=$reparacion->factura_rep?></td>
                <td>&nbsp;&nbsp;&nbsp;&nbsp;<?=$reparacion->precio_rep?></td>
                <td>&nbsp;&nbsp;&nbsp;&nbsp;<a onclick="if(confirma() == false) return false"
href="<?=$base_url()?>index.php/reparacion_eliminar/borrar_reparacion/<?=$reparacion->id_rep?>">Eliminar</a></td>
            </tr>
        <?php
        }
    }else{
    ?>
    <tr>
        <td colspan="7" align="center">No hay mantenimientos preventivos</td>
    </tr>
    <?php
    }
    ?>
</table>
<?php

```

```

        echo form_close();
        }
        else{
            echo"<p align='center'><br><br>NO TIENE PERMISO PARA ACCEDER A LA
APLICACIÓN<br><br></p>";
        }
    ?>
</body>
</html>

```

4.26.3.3. Modelo

```

class Reparacion_eliminar_model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
    }
    //obtenemos reparaciones para poder eliminar
    function getReparaciones()
    {
        $this->load->database();
        $datarep = array();
        $query = $this->db->get('mcorrectivo');
        if($query->num_rows()>0)
        {
            foreach ($query->result() as $fila)
            {
                $datarep[] = $fila;
            }
            return $datarep;
        }
    }

    //function para eliminar la reparacion
    function eliminar_reparacion($id)
    {
        $this->db->where('id_rep',$id);
        return $this->db->delete('mcorrectivo');
    }
}

```

4.26.4. Modificar Mantenimiento Correctivo

4.26.4.1. Controlador

```

class Reparacion_actualizar extends CI_Controller
{
    function __construct()
    {

```

```

        parent::__construct();
        $this->load->model('reparacion_actualizar_model');
    }

    function index()
    {
        $this->load->view('header_view');
        //con este array enviamos los datos a mostrar a la vista
        $data['data'] = $this->reparacion_actualizar_model->getReparaciones();
        $this->load->view('reparacion_actualizar_view', $data);
        $this->load->view('footer_view');
    }

    public function logout_user()
    {
        $this->session->sess_destroy();
        $this->load->view('header_view');
        $this->load->view('welcome_message');
        $this->load->view('footer_view');
    }

    //funcion para mostrar los datos dependiendo del id
    function mostrar_datos()
    {
        $id = $this->uri->segment(3);
        $this->load->view('header_view');
        //con este array enviamos los datos a mostrar a la vista
        $data['data'] = $this->reparacion_actualizar_model-
>getReparacionesSeleccionado($id);
        $data['dataveh'] = $this->preventivo_nuevo_model->getvehiculos();
        $data['data'] = $this->preventivo_nuevo_model->gettalleres();
        $this->load->view('reparacion_actualizar_datos_view', $data);
        $this->load->view('footer_view');
    }

    //función para actualizar una reparación dependiendo del id
    function editar_reparacion()
    {
        $id = $this->input->post('id_rep');
        $fecha = $this->input->post('fecha');
        $detalle = $this->input->post('detalle');
        $taller = $this->input->post('taller');

        $vehiculo = $this->input->post('vehiculo');
        $placa = $this->input->post('placa');
        $kilometraje = $this->input->post('kilometraje');
        $precio = $this->input->post('precio');
        $factura = $this->input->post('factura');
        $mantenimiento = $this->input->post('proximo');
    }

```

```

//ahora procesamos los datos hacia el modelo que
debemos crear
$editar_datos = $this->reparacion_actualizar_model-
>editar_reparacion(
    $id,
    $fecha,
    $detalle,
    $taller,
    $vehiculo,
    $placa,
    $kilometraje,
    $precio,
    $factura,
    $mantenimiento
);

// $update = $this->reparacion_actualizar_model->editar_reparacion($id);

//cargamos el modelo y llamamos a la función update()
// $this->load->model('mantenimiento_model');
// $this->mantenimiento_model->update($data);

    redirect(base_url().'index.php/reparacion_actualizar');
}
}

```

4.26.4.2. Vista

```

<?
require("../conec.php");
    $link=Conectarse();

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Actualizar Mantenimiento Correctivo</title>
<link rel="stylesheet" href="http://localhost/vehiculo/scripts/tablas.css">
<script type="text/javascript">
function confirma(){
    if (confirm("¿Realmente desea modificar los datos?")){
        }
    else {
        return false
    }
}
}

```



```

        return $datar;
    }
}

function getReparacionesSeleccionado($id)
{
    $this->load->database();
    $datar = array();
    $this->db->where('id_rep',$id);
    $query = $this->db->get('mcorrectivo');
    if($query->num_rows()>0)
    {
        foreach ($query->result() as $fila)
        {
            $datar[] = $fila;
        }
        return $datar;
    }
}

function getvehiculos() {
    $this->load->database();
    $datav = array();
    $query = $this->db->get('vehiculos');
    if ($query->num_rows() > 0) {
        foreach ($query->result_array() as $row){
            $datav[] = $row;
        }
    }
    $query->free_result();
    return $datav;
}

function gettalleres() {
    $this->load->database();
    $datat = array();
    $query = $this->db->get('talleres');
    if ($query->num_rows() > 0) {
        foreach ($query->result_array() as $row){
            $datat[] = $row;
        }
    }
    $query->free_result();
    return $datat;
}

//function para actualizar la reparacion
function
editar_reparacion($id,$fecha,$detalle,$taller,$vehiculo,$placa,$kilometraje,$precio,$fac
tura,$mantenimiento)
{

```

```

        $this->db->set('fecha_rep', $fecha);
        $this->db->set('detalle_rep', $detalle);
        $this->db->set('id_ta', $taller);
        $this->db->set('id_veh', $vehiculo);
        $this->db->set('placa_veh', $placa);
        $this->db->set('kilometraje_veh', $kilometraje);
        $this->db->set('precio_rep', $precio);
        $this->db->set('factura_rep', $factura);
        $this->db->set('proxmanten_rep', $mantenimiento);
        $this->db->where('id_rep',$id);
        $this->db->update('mcorrectivo');
    }
}

```

4.26.5. Buscar Mantenimiento Correctivo

4.26.5.1. Controlador

```

class Reparacion_buscar extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('reparacion_buscar_model');
    }

    function index()
    {
        $this->load->view('header_view');
        //con este array enviamos los datos a mostrar a la vista
        $this->load->view('reparacion_buscar_view');
        $this->load->view('footer_view');
    }

    public function logout_user()
    {
        $this->session->sess_destroy();
        $this->load->view('header_view');
        $this->load->view('welcome_message');
        $this->load->view('footer_view');
    }

    //funcion para mostrar los datos dependiendo la fecha
    function listar_datos()
    {
        $placa = $this->input->post('placa');
        $this->load->view('header_view');
    }
}

```

```

        $this->load->view('reparacion_buscar_view');
        $data['datar'] = $this->reparacion_buscar_model-
>getReparacionesListado($placa);
        $this->load->view('reparacion_buscar_listado_view', $data);
        $this->load->view('footer_view');
    }
    //función para actualizar una reparación dependiendo del id
    function mostrar_datos()
    {
        $id = $this->uri->segment(3);
        $this->load->view('header_view');
        $this->load->view('reparacion_buscar_view');
        //con este array enviamos los datos a mostrar a la vista
        $data['datar'] = $this->reparacion_buscar_model-
>getReparacionesBuscado($id);
        $data['dataveh'] = $this->reparacion_buscar_model->getvehiculos();
        $data['datat'] = $this->reparacion_buscar_model->gettalleres();
        $this->load->view('reparacion_buscar_datos_view', $data);
        $this->load->view('footer_view');
    }
}

```

4.26.5.2. Vista

```

<? require("../conec.php");
    $link=Conectarse();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Buscar Mantenimiento Correctivo</title>
<link rel="stylesheet" href="http://localhost/vehiculo/scripts/tablas.css">
<script type="text/javascript">
function confirma(){
    if (confirm("¿Realmente desea modificar los datos?")){
        }
        else {
            return false
        }
    }
}
</script>
</head>
<body>
<? if($this->session->userdata('usuario')!="")
    {?>
<div align="center"><BR><BR>BUSCAR MANTENIMIENTO CORRECTIVO<BR><BR></div>

```

```

<form name="buscar"
action="http://localhost/vehiculo/index.php/reparacion_buscar/listar_datos"
method="POST">
<table width="20%" cellspacing="10" cellpadding="10">
  <tr>
    <td align="left">&nbsp;  PLACA</td>
    <th align="left">
      &nbsp;  <select name="placa">
        <option-->Escoja</option>
        <? $SQL = "SELECT DISTINCT (placa_veh) FROM
mcorrectivo";
        $busca = mysql_query($SQL,$link);
        while($placa=mysql_fetch_object($busca))
        {
          echo'<option value="" . $placa->placa_veh.' . $placa-
>placa_veh.'</option>';
        }?>
      </select>
      &nbsp;  <input type="submit" value="BUSCAR" /></th>
    </tr>
  </table>
</form>
<?php }
      else{
        echo"<p align='center'><br><br>NO TIENE PERMISO PARA
ACCEDER A LA APLICACIÓN<br><br></p>";
      }
    ?>
  </body>
</html>
<?
  $link=Conectarse();

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Buscar Mantenimiento Correctivo</title>
<link rel="stylesheet" href="http://localhost/vehiculo/scripts/tablas.css">
<script type="text/javascript">
function confirma(){
  if (confirm("Se está realizando la búsqueda")){
    }
    else {
      return false
    }
  }
}
</script>

```



```

        <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<?=$reparacion->proxmanten_rep?> km</td>
            <td>&nbsp;&nbsp;&nbsp;&nbsp;<a onclick="if(confirm() == false) return false"
href="<?=base_url()?>index.php/reparacion_buscar/mostrar_datos/<?=$reparacion-
id_rep?>">Detalle</a></td>
        </tr>
        <?php
            //$cont++;
        }
    }else{
        ?>
    <tr>
        <td colspan="10" align="center">No hay mantenimientos
preventivos</td>
    </tr>
    <?php
    }
    ?>
</table>
<?php
    echo form_close();
    }
    else{
        echo"<p align='center'><br><br>NO TIENE PERMISO PARA
ACCEDER A LA APLICACIÓN<br><br></p>";
    }
    ?>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Buscar Mantenimiento Correctivo</title>
<link rel="stylesheet" href="http://localhost/vehiculo/scripts/tablas.css">
</head>
<body>
<? if($this->session->userdata('usuario')!=")
    {?>
<div align="center"><BR><BR><BR><BR></div>
<table width="60%" cellspacing="10" cellpadding="10">
    <?php
        if ($datar)
        {
            //$cont=1;
            foreach($datar as $reparacion)
            {
                //$color = (($cont % 2) == 0) ? '#dae5fd' : '#b9cdfa'; ?>
                <tr style="background-color:#OFF">

```



```

</tr>
    <?php
    }
    }else{
    ?>
    <tr>
        <td colspan="7" align="center">No hay mantenimientos correctivos</td>
    </tr>
    <?php
    }
    ?>
</table>
<?php
    echo form_close();
    }
    else{
    echo"<p align='center'><br><br>NO TIENE PERMISO PARA ACCEDER A LA
    APLICACIÓN<br><br></p>";
    }
    ?>
</body>
</html>

```

4.26.5.3. Modelo

```

class Reparacion_buscar_model extends CI_Model
{

    function __construct()
    {
        parent::__construct();
    }
    //obtenemos mantenimientos correctivos para poder actualizar

    function getReparacionesListado($id)
    {
        $this->load->database();
        $datar = array();
        $this->db->where('placa_veh',$id);
        $query = $this->db->get('mcorrectivo');
        if($query->num_rows()>0)
        {
            foreach ($query->result() as $fila)
            {
                $datar[] = $fila;
            }
        }
        return $datar;
    }
}

```

```

    }

    function getReparacionesBuscado($id)
    {
        $this->load->database();
        $datar = array();
        $this->db->where('id_rep',$id);
        $query = $this->db->get('mcorrectivo');
        if($query->num_rows()>0)
        {
            foreach ($query->result() as $fila)
            {
                $datar[] = $fila;
            }
        }
        return $datar;
    }
}

```

```

function getvehiculos() {
    $this->load->database();
    $datav = array();
    $query = $this->db->get('vehiculos');
    if ($query->num_rows() > 0) {
        foreach ($query->result_array() as $row){
            $datav[] = $row;
        }
    }
    $query->free_result();
    return $datav;
}

```

```

        function gettalleres() {
            $this->load->database();
            $datat = array();
            $query = $this->db->get('talleres');
            if ($query->num_rows() > 0) {
                foreach ($query->result_array() as $row){
                    $datat[] = $row;
                }
            }
            $query->free_result();
            return $datat;
        }
    }
}

```

4.27. Pruebas de Software

Para que las pruebas de software sean más efectivas, las realizaron los usuarios y el fabricante del sistema, es así que se realizó las siguientes pruebas:

- **Pruebas Unitarias:** Empezamos por realizar pruebas desde lo más pequeño hacia lo más grande o complejo del software. Estas pruebas se centran en el correcto funcionamiento de la lógica del procesamiento interno. De tal manera que se realizaron pruebas sobre cada uno de los métodos de las clases, estos deben realizar el proceso para el cual fueron creados; para el efecto se utilizó la herramienta PHPUNIT la cual analizó la funcionalidad de cada uno de los métodos. Luego de esto se realizó pruebas prácticas sobre los siguientes módulos: modulo mantenimiento preventivo, mantenimiento correctivo, vehículos, talleres, combustible, accesorios y accidentes, sobre cada uno de estos se realizó pruebas de inserción, eliminación y actualización de información; obteniéndose resultados satisfactorios en cada uno de ellos. Cada vez que se realizó alguna modificación se ejecutaron nuevamente las pruebas con el propósito de no perder ninguna funcionalidad. De esta forma las pruebas unitarias resultaron ser satisfactorias.
- **Pruebas de Integración:** Una vez que se realizaron las pruebas unitarias se verificó que los módulos que pasaron dichas pruebas interactúen entre sí de manera correcta; esta etapa sirvió para detectar errores de interfaces desarrolladas y se procedió a corregirlas, así se procedió a combinar componentes, se probó el programa como un todo. Para el efecto se realizaron los respectivos ensayos en estos módulos, realizando búsquedas y reportes; dependiendo del requerimiento, de igual forma se obtuvo resultados precisos.
- **Pruebas de validación.-** Todas y cada una de las pruebas antes mencionadas satisficieron las necesidades del cliente. Es decir fueron validadas como efectivas.
- **Pruebas del Sistema:** Se probó el sistema como un todo, verificando que se cumplan los requisitos de calidad estipulados; para el efecto se validan los datos de los formularios y que cada entrada haga lo que debe, por ejemplo

que se realicen las inserciones, eliminaciones, actualizaciones y búsquedas correctamente en la base de datos. Además se realizaron pruebas de seguridad, de resistencia y de desempeño del sistema. Realizadas estas pruebas el sistema trabaja perfectamente, sus módulos interactúan unos con otros en forma efectiva y la información es precisa y clara. Se concluye diciendo que el sistema es rápido, seguro, resistente y su funcionalidad es óptima.

4.28. Resultados de la aplicación del software

Es necesario poner de manifiesto la diferencia marcada que existe entre los tiempos de respuesta y el uso de recursos en el manejo de información de mantenimiento vehicular en la Escuela de Conducción de la Universidad Nacional de Chimborazo, este proceso lo realizaron la Secretaria, así como los Instructores; al final se realizó una sumatoria del tiempo que le tomó a cada uno realizar el proceso de registro de mantenimiento vehicular, como se detalla a continuación:

Tabla 17. Medición de la manipulación de información referente a mantenimiento vehicular manual y mediante la utilización del sistema

Tipo de mantenimiento	Tiempo manual (minutos)					Tiempo sistema (minutos)				
	EVENTOS					EVENTOS				
	Inserción	Eliminación	Actualización	Búsqueda	Reporte	Inserción	Eliminación	Actualización	Búsqueda	Reporte
Preventivo	12	5	10	25	60	2	0.1	1	0.3	0.5
Correctivo	10	5	8	20	60	1.5	0.1	1	0.3	0.5
Suma total	22	10	18	45	120	3.5	0.2	2	0.6	1
Promedio	11	5	9	22.5	60	1.75	0.1	1	0.3	0.5

En base a los criterios anteriormente analizados se puede determinar que el sistema proporciona información correcta, cierta y de acuerdo a lo que el usuario requiere en un tiempo rápido.

Una vez que el sistema ha sido desarrollado e implementado, es ineludible determinar si éste cumple con el objetivo para el cual fue creado. Para esto se detalló en el Capítulo 2 el framework en plataforma de software libre adecuado siendo Codeigniter el que presentaba las mejores características para aprovecharlas en el sistema.

El sistema permite acceder a información veraz, completa y precisa de mantenimiento vehicular, de vehículos, talleres, instructores y tanqueada de combustible, tomando en cuenta que el tiempo desde que el usuario procede a realizar una petición (consulta a la base de datos) y visualizar el respectivo resultado (información proporcionada desde la base de datos) no sobrepasa los 5 segundos.

Además se puede observar aspectos como:

- El límite de tiempo para que una persona mantenga su atención en una página mientras espera es de unos 4 segundos.
- El tiempo para que los usuarios sientan que una página web reacciona instantáneamente es de una décima (0.1) de segundo.
- Teniendo en cuenta que una gran parte de los usuarios no siempre van a acceder desde computadores con características óptimas, las páginas web no exceden los 40 Kb lo que, para un módem básico de 56 kbps, representa unos 5 segundos de tiempo de descarga en el browser.

Sin embargo, también se debe tomar en cuenta aspectos como características hardware del computador (especialmente cantidad de memoria RAM, así como velocidad del procesador), o también la velocidad de conexión a internet (por ejemplo 56 kbps, 128kbps, 256 kbps, 512kbps) son aspectos que inciden directamente en la forma de acceder a los datos del sistema y la obtención de los respectivos resultados.

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Existen varios frameworks de código abierto que se pueden utilizar dependiendo de las necesidades de un sistema informático, para la gestión del mantenimiento vehicular de la Escuela de Conducción de la Universidad Nacional de Chimborazo, se necesitaba eficiencia en la ejecución del código, así como la reutilización y facilidad de edición del código, por lo que Codeigniter proporcionaba estas características. Además tiene una característica especial que es estructurar los archivos en base a modelos, controladores y vistas siendo casi en su totalidad orientada a objetos.
- Los procesos de gestión y mantenimiento vehicular como son: mantenimiento preventivo, mantenimiento correctivo, consumo de combustible, reposición de accesorios, accidentes sucedidos en los vehículos, búsqueda de información y creación de reportes basados en los parámetros anteriores; se los ha venido realizando de manera manual, esto influye en la subutilización de recursos tanto materiales, humanos como tecnológicos. Cada uno de estos parámetros en forma individual tomaban un tiempo en promedio de 21.5 minutos. Con la implementación del sistema se reduce considerablemente estos tiempos de respuesta a 0.73 minutos; por consiguiente se disminuye la carga de trabajo humano, se optimizan tiempos y se hace uso de la plataforma tecnológica disponible en la Escuela de Conducción de la Universidad Nacional de Chimborazo.
- Para el desarrollo e implementación del sistema se han utilizado varias herramientas open source como son un servidor web Apache, motor de base de datos MYSQL, lenguaje de programación PHP, plataforma de framework Codeigniter, para cada una de estas herramientas es necesario aplicar los principios de la ingeniería de software para incorporarlos a los requerimientos de la Escuela de Conducción.

5.2. Recomendaciones

- Se recomienda dar periódicamente el mantenimiento respectivo al sistema; para que su eficiencia y eficacia sean óptimos. Se debe tener muy en cuenta la forma en que trabaja el framework Codeigniter siendo los controladores los que manejan tanto las vistas (la interfaces que visualizan los usuarios) así como los modelos (los que ejecutan las sentencias SQL o los archivos que se conectan a la base de datos).
- Se recomienda el uso permanente del sistema de gestión vehicular ya que será de gran ayuda para optimizar el trabajo de todo el personal, además se debe utilizar un terminal con características suficientes para la adecuada ejecución del sistema informático, en lo que se refiere a memoria RAM principalmente, siendo lo sugerido 2 Gb.
- Se recomienda tener conocimiento de las herramientas open source necesarias para implementar un sistema informático, desde el servidor web adecuado así como el motor de base de datos ya que de eso depende una respuesta eficiente y eficaz del sistema informático, recalcando que debe ser independiente de la plataformas y se debe complementar el sistema informático de gestión de mantenimiento vehicular con un sistema que administre las clases de manejo de los estudiantes.

5.3. BIBLIOGRAFÍA

Bermúdez, G., Delgado, C., & Gonzáles , A. (Agosto de 2014). *Google Académico*. Recuperado el 26 de Septiembre de 2014, de <http://rus.ucf.edu.cu/index.php/rus/article/view/351/202>

Boucly, F. (1988). *Gestión del mantenimiento*. Madrid-España: Asociación Española de Normalización y Certificación (AENOR), p. 19.

Cáceres, P., & Marcos, E. (s.f.). www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf. Recuperado el 19 de Agosto de 2014, de <http://www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf>

Cobo, A., Gómez, P., Pérez, D., & Rocha, R. (2005). *PHP Y MySQL Tecnologías para el desarrollo de aplicaciones web*. España: Díaz de Santos.

Coronel, C., Morris, S., & Rob, P. (2011). *BASE DE DATOS Diseño, implementación y administración* (Novena ed.). México D.F: CENGAJE Learnig.

Chiavenato, I. (s.f.). *Promonegocios*. Recuperado el 3 de Julio de 2014, de <http://odauts.com/blogsuts/admongeneral/files/2013/10/Proceso-Administrativo.pdf>

Durán, L. (2007). *El gran libro del PC interno*. Barcelona: MARCOMBO, S.A .

Ferrater, J. (1979). *Sistema Diccionario de filosofía* . Recuperado el 7 de Julio de 2014, de <http://www.filosofia.org/enc/fer/sistema.htm>

GLADYS, C. d. (1996). *Análisis y Administración Financiera* (Cuarta ed.). Santa Fé de Bogotá, D.C.- Colombia:Edit. Corcas Editores Ltda. p. 3, 4, 5, 6, 10

Kendall, K., & Kendall, J. (2011). *Análisis y diseño de sistemas*. México: Pearson.

Maigua, G., & López, E. (2012). *BUENAS PRACTICAS EN LA GESTION Y DIRECCION DE PROYECTOS INFORMATICOS*. Argentina: edUTecNe.

Navarro, L., & otros. (1997). *Gestión integral de mantenimiento*. Barcelona, España: MARCOMBO S.A, p. 5-6.

Navarro, L., & otros, Op. Cit., p. 11-12; Boucly, F, Op. Cit., p. 22-23; Tavares, L., *Administración moderna del mantenimiento*, Formato PDF, Rio de Janeiro-Brasil-2013, p.27.

Pavón, J. (2011). *Creación de un portal con PHP y MySQL*. México: Alfaomega Grupo editor S.A.

Sánchez, J. (2007). *Sistema de gestión de base de datos*. Lima: RitisGraff.

Silberschat, A., Korth, H., & Sudarshan, S. (2002). *FUNDAMENTOS DE BASES DE DATOS*. Madrid: McGRAW-HILL.

Stallman, R. M. (2004). *Software libre para una sociedad libre*. (J. Rowan, D. Sanz, & L. Trinidad, Trads.) Madrid: Traficantes de sueños.

Tamayo, M., Lemes, J., & Naranjo, T. (s.f.). *Google Accadémico*. Recuperado el 26 de Septiembre de 2014, de <http://jornada2011.sld.cu/index.php/jornada/2011/paper/viewFile/105/39>

Vaswani, V. (2010). *FUNDAMENTOS DE PHP*. MÉXICO: McGraw Hill.

5.3.1. WEBGRAFIA

www.etnassoft.com/

www.mcgraw-hill.es

<http://www.mcgraw-hill.es/bcv/guide/capitulo/8448148797.pdf>

<http://www.mclibre.org/descargar/docs/libros/libro-blanco-sl-iii-200702.pdf>

<http://desarrollomovilmultiplataforma.blogspot.com/2012/08/aspectos-teoricos-framework>.

<https://www.anf.es/es/pdf/bbs-open-source.pdf>

<http://mundoinformatica.portalmundos.com/herramientas-opensource-para-el-desarrollo-y-diseño-de-paginas-web/>

<http://opensource.org/>

<http://opensource.org/licenses>

<https://www.apachefriends.org/es/>

<https://ellislab.com/codeigniter>

<http://www.php.net>

<http://repositorio.utn.edu.ec/handle/123456789/1693>


<http://repositorio.utn.edu.ec/handle/123456789/1693>

www.udemy.com

http://recursos.fundacionesplai.org/intranet/dinamizadores/recursos_tecnologicos/ejercicios_de_dreamweaver/materiales/Manual%20de%20Dreamweaver.pdf

5.4. ANEXOS


5.4.1. Control de Mantenimiento de vehículos.



UNIVERSIDAD NACIONAL DE CHIMBORAZO

ESCUELA DE FORMACION DE CONDUCTORES PROFESIONALES

CONTROL DE MANTENIMIENTO DE VEHICULOS




MANTENIMIENTO N° **0101**

FECHA DE EMISIÓN: _____

TIPO: SEDAN AÑO: 2012 N° SERIE: EL4TD51Y0C0118039 N° MOTOR: F15S34015331 N° PLACA: HEJ 4139 N° CODIGO: _____

NOMBRE DEL INSTRUCTOR: Inj. Juan Carlos Cordova


FECHA	KILOMETRAJE ACTUAL	ESPECIFICACIONES DEL MANTENIMIENTO Y/O REPARACION	DETALLE DE LOS REPUESTOS UTILIZADOS	NOMBRE DEL TALLER	VALOR FACTURA	KILOMETRAJE PROXIMO CHEQUEO
14-02-2013	15104	Mantenimiento de los 15000km Revisión de los niveles de fluidos Cambio de Filtro de Combustible Cambio Express Revisión APC Frenos.	4 Litros de Aceite 1 Filtro de Aceite 1 Filtro de Combustible	SUPERILUMINA - TECNICENTRO	9240	


 FIRMA AUTORIZADA Y SELLO

RECIBI CONFORME _____

IMPRENTA UNIVERSITARIA 2013

5.4.2. Reporte de carga de combustible


 **UNIVERSIDAD NACIONAL DE CHIMBORAZO**
ORDEN DE PROVISIÓN DE COMBUSTIBLE VEHÍCULOS N° 004613

Vehículo: 1 Tipo combustible: Gasolina Extra
Placas: HE11136 Cantidad: 91.244
Kilometraje: 34467 Valor total: \$ 1220
(AL PONER COMBUSTIBLE)
Conductor responsable: Gabriel Santos
Cédula identidad N°: 000303155-0
Lugar de destino: Ruta #3
Objeto de la movilización: Práctica de Conducción
Miembros de la Comisión: _____

IMPRESA UNIVERSITARIA 2014

Riobamba, 1 de ABRIL de 2015

Autorizado,



Dra. María Angélica Barba M.
RECTORA

5.4.3. Reporte de consumo de combustible



ESCUELA DE FORMACIÓN Y CAPACITACIÓN DE CONDUCTORES PROFESIONALES
DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO

Riobamba 30 de Abril

INFORME DE KILOMETRAJE Y CARGA DE COMBUSTIBLE EN EL SEXTO PERIODO DE
ACTIVIDADES EN INSTRUCCIÓN DE PRACTICA DE CONDUCCIÓN VEHICULAR DEL 01 DE
NOVIEMBRE 2014 AL 30 DE ABRIL 2015

VEHÍCULO:#.....02....

CUSTODIO DOCENTE DE INSTRUCCIÓN PRÁCTICA: Segundo Roberto Santos García

KILOMETROS RECORRIDOS

Kilómetros a la fecha de inicio :01 de Nov. 2014	Kilómetros a la fecha de culminación: 30 de Abr. 2015	Total de Kilómetros recorridos

GARGA DE COMBUSTIBLE

No.	Fecha de la carga de combustible	Galones de combustible	Cantidad en Dólares
1	10 de Noviembre 2014	10,13	15,00
2	21 de Noviembre 2014	9,80	14,50
3	3 de Diciembre 2014	8,97	13,27
4	11 de Diciembre 2014	8,11	12,00
5	7 de Enero 2015	9,11	13,51
6	14 de Enero 2015	8,98	13,80
7	21 de Enero 2015	8,68	12,85
8	11 de Febrero 2015	8,52	12,60
9	27 de Febrero 2015	8,15	12,06
10	11 de Marzo 2015	8,71	12,90
11	20 de Marzo 2015	7,09	10,50
12	1 de Abril 2015	7,55	11,17
13			
14			
15			
TOTALES= 12 cargues de combustible		103,80 Galones / Extra	\$ 154,16

Es todo cuanto puedo informar,

Atentamente

Roberto Santos
DOCENTE DE INSTRUCCIÓN PRACTICA

**5.4.4. Certificado de la implementación del Sistema de Gestión Vehicular en
la Escuela de Conducción de la Universidad Nacional de Chimborazo.**

Riobamba, 28 de julio de 2015
Oficio.279-EFCP-UNACH-2015


Ingeniero
Alfredo Colcha
**DOCENTE TUTOR DEL PROYECTO DE INVESTIGACIÓN
“IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN VEHICULAR
UTILIZANDO HERRAMIENTAS OPEN SOURCE PARA LA
ESCUELA DE CONDUCCIÓN DE LA UNIVERSIDAD NACIONAL
DE CHIMBORAZO**
Presente

De mi consideración:

Con fecha 29 de julio del presente año, el Sr. Hugo Flores Egresado de la Facultad de Ingeniería, Carrera Ingeniería en Computación, realizó la implementación del sistema de gestión vehicular utilizando herramientas OPEN SOURCE, en la Escuela de Formación de Conductores Profesionales de la UNACH, dejándolo en pleno funcionamiento y con los requerimientos especificados en su trabajo de investigación.

Es todo cuanto puedo certificar en honor a la verdad, facultado al Sr. Hugo Flores hacer uso del presente certificado en las gestiones de graduación.

Atentamente,



Ing. Diana Erazo Riera
**DIRECTORA ADMINISTRATIVA
ESCUELA DE CONDUCCIÓN UNACH**



Elaborado por: Ing. Gina Zabala Ramirez