



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN VEHICULAR CON JAVA  
SERVER FACE PARA EL DEPARTAMENTO DE MANTENIMIENTO  
VEHICULAR DE LA ESPOCH”**

Trabajo de Titulación presentado para optar al grado académico de:

**INGENIERA EN SISTEMAS INFORMÁTICOS**

AUTORES:

**Patricia Alexandra Herrera Saltos**

**Vanessa Margarita Vargas Guambo**

**RIOBAMBA – ECUADOR**

**2016**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN SISTEMAS**

El Tribunal de Tesis certifica que: El trabajo de investigación: IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN VEHICULAR CON JAVA SERVER FACE PARA EL DEPARTAMENTO DE MANTENIMIENTO VEHICULAR DE LA ESPOCH, de responsabilidad de Patricia Alexandra Herrera Saltos y Vanessa Margarita Vargas Guambo, ha sido minuciosamente revisado por los Miembros del Tribunal de Tesis, quedando autorizada su presentación.

<b>NOMBRES</b>	<b>FIRMAS</b>	<b>FECHA</b>
Ing. Washington Luna. <b>DECANO DE LA FACULTAD DE INFORMÁTICA Y ELETRÓNICA</b>	_____	_____
Ing. Julio Santillán <b>DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS</b>	_____	_____
Ing. Raúl Rosero <b>DIRECTOR DE TESIS</b>	_____	_____
Ing. Eduardo Villa <b>MIEMBRO DE TESIS</b>	_____	_____

## **RESPONSABILIDAD DEL AUTOR**

Nosotras, Patricia Alexandra Herrera Saltos y Vanessa Margarita Vargas Guambo, somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo

---

PATRICIA ALEXANDRA HERRERA

SALTOS

---

VANESSA MARGARITA VARGAS

GUAMBO

## DEDICATORIA

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme haber llegado hasta este momento tan importante de mi formación profesional. A mis padres por ser el pilar más importante, por su apoyo y amor incondicional. Y a mi esposo por su gran amor y apoyo constante para este gran paso y a la mayor motivación de mi vida mi hijo Carlos Andrés.

Patricia Herrera.

A Dios por la vida y salud que nos da cada día, a mi madre por su apoyo incondicional, a mi familia que siempre han estado a mi lado, a mi hija y esposo que han sido mi fortaleza para cumplir esta importante etapa de mi vida.

Vanessa Vargas.

## **AGRADECIMIENTO**

El más sincero agradecimiento a la Escuela Superior Politécnica de Chimborazo, por darnos la oportunidad de obtener una profesión y ser una ayuda para la sociedad.

A nuestra familia por su apoyo y comprensión.

Patricia y Vanessa

## INDICE DE ABREVIATURAS

<b>ESPOCH</b>	Escuela Superior Politécnica de Chimborazo
<b>IDE</b>	Entorno de Desarrollo Integrado
<b>IDP</b>	Incremento del Producto (SCRUM)
<b>JSF</b>	Java Server Faces
<b>MVC</b>	Modelo Vista Controlador
<b>SISCOVE</b>	Sistema de Control Vehicular
<b>SRS</b>	Especificación de Requerimientos de Software

## TABLA DE CONTENIDOS

<b>RESPONSABILIDAD DEL AUTOR</b> .....	iii
<b>DEDICATORIA</b> .....	iv
<b>AGRADECIMIENTO</b> .....	v
<b>INDICE DE ABREVIATURAS</b> .....	vi
<b>TABLA DE CONTENIDOS</b> .....	7
<b>INDICE DE TABLAS</b> .....	9
<b>INDICE DE FIGURAS</b> .....	10
<b>RESUMEN</b> .....	12
<b>SUMMARY</b> .....	13
<b>INTRODUCCIÓN</b> .....	14
<b>CAPÍTULO I</b> .....	15
<b>1. MARCO REFERENCIAL</b> .....	15
<b>1.1. Planteamiento del Problema</b> .....	15
<b>1.1.1. Antecedentes</b> .....	15
<b>1.1.2. Formulación del problema</b> .....	15
<b>1.1.3. Sistematización del problema</b> .....	16
<b>1.2. Justificación del trabajo de grado</b> .....	16
1.2.1. Justificación teórica.....	16
1.2.2. Justificación aplicativa .....	17
<b>1.3. Objetivos</b> .....	19
1.3.1. Objetivo general.....	19
1.3.2. Objetivos específicos.....	19
<b>1.4. Métodos y técnicas</b> .....	19
1.4.1. Métodos .....	19
1.4.2. Técnicas de investigación.....	20
<b>CAPÍTULO II</b> .....	22
<b>2. MARCO TEÓRICO</b> .....	22
<b>2.1. Lenguaje de programación Java</b> .....	22
<b>2.2. Entorno de Desarrollo Java - IDE Netbeans</b> .....	23
<b>2.3. JavaServer Faces</b> .....	25
<b>2.4. Primefaces</b> .....	27
<b>2.5. Modelo Vista Controlador (MVC)</b> .....	28
<b>2.6. PostgreSQL</b> .....	29

2.7.	<b>SCRUM</b> .....	30
	<b>CAPÍTULO III</b> .....	34
3.	<b>INGENIERÍA DE SOFTWARE</b> .....	34
3.1.	<b>Especificación de Requerimientos de Software (SRS)</b> .....	34
3.1.1.	Introducción .....	34
3.1.2.	Descripción general.....	36
3.1.3.	Requisitos Específicos.....	38
3.2.	<b>Implementación del Sistema</b> .....	44
3.2.1.	Fase I: Planificación.....	45
3.2.2.	Fase II: Desarrollo de sprints .....	45
3.2.3.	Fase III: Cierre .....	47
3.2.4.	Software utilizado en la implementación .....	51
3.2.5.	Implementación del Módulo de Diagnóstico Vehicular .....	51
	<b>CAPÍTULO IV</b> .....	57
4.	<b>EVALUACIÓN DE LA EFICIENCIA Y EFICACIA DEL SOFTWARE DESARROLLADO</b> .....	57
	<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	61
	<b>CONCLUSIONES</b> .....	61
	<b>RECOMENDACIONES</b> .....	62
	<b>BIBLIOGRAFÍA</b> .....	63
	<b>ANEXOS</b> .....	65
	<b>Anexo I: Pila del Producto e Historias de Usuarios</b> .....	65
	<b>Anexo II: Diagramas de Modelado UML</b> .....	77
	<b>Anexo III: Arquitectura del Sistema e Interfaz Gráfica</b> .....	96
	<b>Anexo IV: Estructuras de Datos – Base de Datos SISCOVE</b> .....	99
	<b>Anexo V: Encuesta para desarrollo de Módulo de Diagnóstico Vehicular</b> .....	113
	<b>Anexo VI: Manual Técnico</b> .....	¡Error! Marcador no definido.
	<b>Anexo VII: Manual de Usuario</b> .....	114

## INDICE DE TABLAS

<b>Tabla 1-3.</b> SRS-Funciones del producto.....	37
<b>Tabla 2-3.</b> Documentación técnica desarrollada – SISCOVE .....	45
<b>Tabla 3-3.</b> Equipo SCRUM – roles y asignaciones.....	45
<b>Tabla 4-3.</b> Planificación de sprints.....	46
<b>Tabla 5-3.</b> Matriz de Seguimiento - Sprint Backlog .....	47
<b>Tabla 6-3.</b> Pruebas de integración del sistema.....	48
<b>Tabla 7-3.</b> Desarrollo de la metodología SCRUM – tiempos y responsables.....	49
<b>Tabla 8-3.</b> Software utilizado en la implementación del Sistema SISCOVE .....	51
<b>Tabla 9-3.</b> Estudiantes matriculados en la Escuela de Ingeniería Automotriz.....	52
<b>Tabla 10-3.</b> Pregunta 1 - Tabulación.....	53
<b>Tabla 11-3.</b> Pregunta 2 - Tabulación.....	54
<b>Tabla 12-3.</b> Pregunta 3 - Tabulación.....	54
<b>Tabla 13-3.</b> Validación de preguntas y lógica de negocios – módulo de diagnóstico vehicular .....	56
<b>Tabla 1-4.</b> Evaluación de Eficiencia – Procesos del Taller .....	57
<b>Tabla 2-4.</b> Evaluación de Eficiencia – Reportes del Sistema .....	58
<b>Tabla 3-4.</b> Evaluación de Eficiencia – Formatos .....	58
<b>Tabla 4-4.</b> Evaluación de Eficiencia – Diagnóstico vehicular .....	58
<b>Tabla 5-4.</b> Evaluación de Eficiencia – Resumen .....	59
<b>Tabla 1-I.</b> Pila del producto, SPRINT 1- Requerimientos funcionales .....	65
<b>Tabla 2-I.</b> Pila del producto, SPRINT 2- Requerimientos funcionales .....	70
<b>Tabla 1-IV.</b> Diccionario de datos SISCOVE.....	101
<b>Tabla 2-IV.</b> Relaciones entre tablas .....	104
<b>Tabla 3-IV.</b> Script de la base de datos .....	104

## INDICE DE FIGURAS

<b>Figura 1-1:</b> Formulación del problema .....	15
<b>Figura 1-2:</b> Funcionamiento de JSF.....	26
<b>Figura 2-2:</b> Modelo Vista Controlador .....	29
<b>Figura 3-2:</b> Metodología SCRUM.....	31
<b>Figura 1-3:</b> SRS - Organización Jerárquica de los perfiles de usuario .....	38
<b>Figura 1-4:</b> Evaluación de Eficiencia – procesos manuales vs ejecución del sistema SISCOVE .....	60
<b>Figura 1-II:</b> Diagrama de Casos de Uso – Proceso: Gestión de información general .....	77
<b>Figura 2-II:</b> Diagrama de Casos de Uso – Proceso: Gestión de bodega .....	78
<b>Figura 3-II:</b> Diagrama de Casos de Uso – Proceso: Reparación Vehicular .....	79
<b>Figura 4-II:</b> Diagrama de Casos de Uso – Proceso: Gestión de reportes del sistema.....	80
<b>Figura 5-II:</b> Diagrama de Casos de Uso – Proceso: Gestión de formatos .....	81
<b>Figura 6-II:</b> Diagrama de Casos de Uso – Proceso: Diagnóstico Vehicular.....	82
<b>Figura 7-II:</b> Diagrama de Secuencias – Caso de Uso: Gestionar tablas dependientes .....	83
<b>Figura 8-II:</b> Diagrama de Secuencias – Caso de Uso: Gestionar tablas independientes .....	84
<b>Figura 9-II:</b> Diagrama de Secuencias – Caso de Uso: Gestionar inventario.....	85
<b>Figura 10-II:</b> Diagrama de Secuencias – Caso de Uso: Gestionar egreso de bodega.....	86
<b>Figura 11-II:</b> Diagrama de Secuencias – Caso de Uso: Gestionar reparación vehicular .....	87
<b>Figura 12-II:</b> Diagrama de Secuencias – Caso de Uso: Gestionar trabajos mecánicos de una reparación vehicular.....	88
<b>Figura 13-II:</b> Diagrama de Secuencias – Caso de Uso: Gestionar mecánicos de una reparación vehicular.....	89
<b>Figura 14-II:</b> Diagrama de Secuencias – Caso de Uso: Generar reportes de vehículos .....	90
<b>Figura 15-II:</b> Diagrama de Secuencias – Caso de Uso: Generar reportes de inventarios .....	91
<b>Figura 16-II:</b> Diagrama de Secuencias – Caso de Uso: Generar reportes de reparación vehicular - inventario .....	91
<b>Figura 17-II:</b> Diagrama de Secuencias – Caso de Uso: Generar reportes de reparación vehicular - mecánica .....	92
<b>Figura 18-II:</b> Diagrama de Secuencias – Caso de Uso: Generar reporte de reparación vehicular individual.....	92
<b>Figura 19-II:</b> Diagrama de Secuencias – Caso de Uso: Generar formatos de documentos administrativos .....	93

<b>Figura 20-II:</b> Diagrama de Secuencias – Caso de Uso: Generar informe técnico de vehículos con datos preimpresos .....	93
<b>Figura 21-II:</b> Diagrama de Secuencias – Caso de Uso: Generar informes de reparación vehicular con datos preimpresos .....	94
<b>Figura 22-II:</b> Diagrama de Secuencias – Caso de Uso: Diagnosticar fallas mecánicas.....	95
<b>Figura 1-III:</b> Arquitectura del Sistema – Sistema SISCOVE en el Servidor Web .....	96
<b>Figura 2-III:</b> Logotipo del sistema SISCOVE .....	96
<b>Figura 3-III:</b> Maquetación del sistema SISCOVE .....	97
<b>Figura 4-III:</b> Diseño preliminar de la interfaz de usuario del Sistema SISCOVE .....	98
<b>Figura 1-IV:</b> Diagrama de Clases – Vehiculo, tipo_vehiculo, departamento y chofer .....	99
<b>Figura 2-IV:</b> Diagrama de Clases – Mecánico, Reparación Vehicular, Usuario_Sistema y Trabajos_Mecánicos .....	99
<b>Figura 3-IV:</b> Diagrama de Clases – Reparación e Inventario .....	100
<b>Figura 4-IV:</b> Diagrama de Clases – Diagnóstico de fallas mecánicas .....	100
<b>Figura 5-IV:</b> Diagrama de Clases – Tablas adicionales no relacionadas .....	101

## RESUMEN

El presente trabajo de titulación se ejecutó con la finalidad implementar el sistema de gestión vehicular con Java Server Faces para el Departamento de Mantenimiento Vehicular (SISCOVE) de la Escuela Superior Politécnica de Chimborazo. Se desarrolló mediante método cualitativo figurando entrevistas en profundidad obteniendo requerimientos funcionales, además la observación participativa para obtener información de procesos manuales, así como también tiempos que toma a cada uno cumplir con dichos procesos, esto permite aumentar validez y fiabilidad de resultados que fueron analizados una vez finalizado el desarrollo del sistema SISCOVE. Las herramientas tecnológicas utilizadas para el desarrollo del sistema son en hardware: una computadora de escritorio, software: Netbeans como Entorno de Desarrollo Integrado, PostgreSQL como motor de base de datos y Glassfish como servidor de páginas web. Mediante el proceso denominado Especificación de Requerimientos de Software (SRS) se implementaron los siguientes módulos: Información General, Procesos del Taller, Reportes del Sistema, Formatos y Diagnóstico Vehicular, gestionado así 62 vehículos que posee la institución. Para definir la lógica de negocios del último módulo se requirió el apoyo de profesionales mecánicos y estudiantes de la Escuela de Ingeniería Automotriz. A fin de establecer el impacto de la utilización del nuevo software, se midió su eficiencia respecto a los procesos manuales, analizando los tiempos globales, un proceso manejado por el sistema se realiza en un tiempo promedio de 24.38seg, si se compara con los 1129.50seg en promedio que toma la realización de un proceso manual, tras los resultados obtenidos se concluye que el sistema mejoró la eficiencia del trabajo del taller mecánico, reduciendo los tiempos de ejecución en un 98%. Se recomienda al jefe del departamento de mantenimiento vehicular de la ESPOCH, solicite realizar la configuración como cliente-servidor de la aplicación, para permitir el acceso general a su módulo de diagnóstico vehicular del sistema SISCOVE.

### **Palabras Clave:**

<SISTEMA DE GESTIÓN VEHICULAR> <ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE> <METODOLOGÍA CUALITATIVA> <SOFTWARE (NETBEANS)> <SOFTWARE (POSTGRESQL)> <SOFTWARE (GLASSFISH)> <SISTEMAS INFORMÁTICAS>.

## SUMMARY

This degree work has been conducted in order to implement the management vehicle system by using java server Faces for the Vehicle maintenance Department (SISCOVE) from the Escuela Superior Politécnica de Chimborazo. This work was developed by means of the qualitative method by arranging deep interviews, which made possible to collect the functional requirements, in addition to this process; the participative observation was applied in order to collect information related to the manual processes as well as the times used by each one to accomplish them, which enables to increase the validity and reliability of the results put under analysis after finishing the development of SISCOVE system. The technological tools used for the development of the system are hardware and software which includes: Netbeans as Integrated Development Environment, PostgreSQL as motor of data base and Glassfish as web pages server. Modules like General information, workshop processes, system reports, Formats and vehicle diagnosis were implemented by means of the process called Software Requirement Specifications (SRS), which enabled to deal with 62 vehicles owned by the institution. In order to define the business logic of the last module it was necessary the support of professional mechanics as well as the support of students from the Automotive engineering school. Also, It was identified the impact of the usage of the new software by evaluating its efficiency regarding to manual processes by checking global times, a process conducted by the system is performed in an average time of 24.38 seconds, compared with the 1129.50 seconds in average taken to accomplish a manual process. After this analysis the findings lead to the conclusion that the system improved the efficiency of performance of the mechanical workshop reducing run times in a 98%. It is recommended that the chief of the Vehicle Maintenance Department at ESPOCH asks for the configuration as client-server of the application to enable the general Access to the module of vehicle diagnosis of SISCOVE system.



## INTRODUCCIÓN

Actualmente, en el taller de mecánica de la ESPOCH se gestiona la información de manera manual con el apoyo de software ofimático, lo cual ha conllevado varios problemas, tales como: duplicación de datos, errores en la digitación e ineficiencia en el procesamiento de datos. Considerando además que la administración central de la institución solicita a este departamento la frecuente entrega de informes de gestión, sus empleados han manifestado la necesidad de automatizar o informatizar sus procesos. Ante este requerimiento, el presente trabajo de titulación describe el desarrollo del Sistema de Control Vehicular (SISCOVE).

Las aplicaciones web han revolucionado la manera en como las empresas funcionan hoy en día. Cualidades como: compatibilidad multiplataforma, acceso universal (red de internet), mantenimiento y escalabilidad centralizados, así como requerimientos mínimos de hardware y software para su utilización, han permitido que éstas vayan desplazando cada vez más a las típicas aplicaciones de escritorio. Considerando todos estos beneficios, además de la existencia de una infraestructura tecnológica de gran nivel en la institución, se propuso la elaboración del sistema con código libre para su funcionamiento en la web, mediante JavaServer Faces. Al utilizar el patrón de arquitectura Modelo Vista Controlador (MVC) se facilitará la escalabilidad de sus funcionalidades y su reingeniería.

En el **Capítulo I** se describe el marco referencial de la tesis, el cual incluye el planteamiento del problema, justificación, objetivos, métodos y técnicas de investigación. El **Capítulo II** presenta el marco conceptual o teórico, mediante una descripción general de los siguientes temas: Netbeans, JSF, Primefaces, Java, PostgreSQL, SCRUM. En el **Capítulo III** se describe el proceso de ingeniería de software, el cual contiene el SRS y una descripción del desarrollo de la metodología SCRUM. El **Capítulo IV** se presentan los resultados de evaluación de eficiencia del sistema. Los **Anexos I, II, III y IV** corresponden a la ingeniería del sistema (resultados de la fase de diseño). En el **Anexo V** contiene la encuesta aplicada a los estudiantes de la ESPOCH para el desarrollo del módulo de diagnóstico vehicular. El **Anexo VI** contiene las páginas del manual de usuario entregado a los usuarios del sistema.

# CAPÍTULO I

## 1. MARCO REFERENCIAL

### 1.1. Planteamiento del Problema

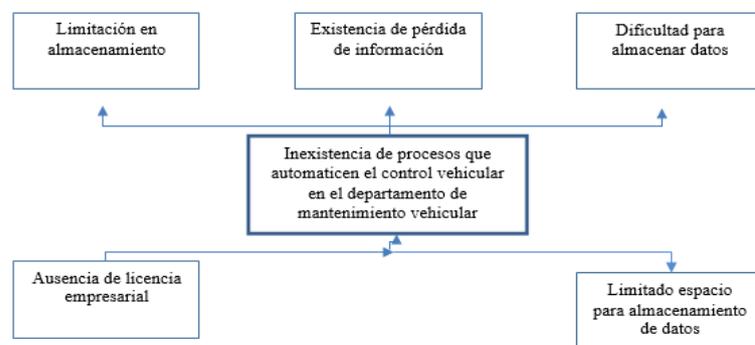
#### 1.1.1. Antecedentes

En la actualidad, toda empresa se encuentra en un escenario cada vez más competitivo, que las obliga a desarrollar y mantener procesos de mejoramiento continuo para disminuir tiempos, costos, riesgos, y mejorar el servicio, ya sea en el ámbito público o privado se requiere almacenar la información de sus actividades para obtener un manejo eficiente de sus datos y transacciones, facilitando así todo proceso.

En el Departamento de Mantenimiento Vehicular de la ESPOCH, se ha hecho cada vez más necesario la automatización de todo proceso aún más en lo que tiene que ver al control de todo lo relacionado con los vehículos que posee dicho Departamento.

Los Sistemas de Gestión de Vehicular, permiten controlar todo lo relacionado con un vehículo ya sea su marca, matrícula, chasis, placa, el nombre del chofer encargado de dicho vehículo, etc.

#### 1.1.2. Formulación del problema



**Figura 1-1:** Formulación del problema

Fuente: Patricia Herrera y Vanessa Vargas

El Departamento de Mantenimiento Vehicular de la ESPOCH, se administran los datos de vehículos manualmente, lo que provoca una gran pérdida de tiempo, pérdida de información, limitación en el almacenamiento y dificultad en la adquisición de información actualizada, se proyecta que el incremento de sus datos sobrepasaran los niveles como para que se pueda seguir ingresando datos manualmente por lo que se ha generado la necesidad de automatizar dichos procesos en un Sistema que realice todo el proceso de control vehicular.

### **1.1.3. Sistematización del problema**

¿Por qué incorporar un Sistema de Gestión Vehicular en el Departamento de Mantenimiento Vehicular de la ESPOCH?

¿En que mejorará la Implementación de un Sistema en el Departamento de Mantenimiento Vehicular de la ESPOCH?

## **1.2. Justificación del trabajo de grado**

### **1.2.1. Justificación teórica**

En la implementación de un sistema que permita la Gestión Vehicular, los programadores necesitan utilizar plataformas, lenguajes específicos y herramientas de interfaz, se debe tener conocimiento acerca de lineamientos y recomendaciones. Actualmente el uso de herramientas para la integración de procesos en el desarrollo de sistemas informáticos, han permitido construir software de calidad por ese motivo se realizó una revisión previa para elegir las más adecuadas.

Se escogió como herramienta de desarrollo la plataforma Java junto con Postrest SQL para el almacenamiento de datos. De igual forma la librería PrimeFaces es una librería de componentes visuales de código abierto para el conjunto Java Server Faces 2.0.

La metodología es esencial en un proyecto y el paso inicial, para el desarrollo de un producto software de calidad depende de las actividades y etapas que permiten su realización, donde elegir la mejor metodología para un determinado proyecto es trascendental para el éxito del mismo, se ha elegido la metodología ágil SCRUM aplicable al desarrollo de software ya que ofrecen una rápida respuesta a cambios de requisitos gracias a su proceso iterativo, permitiendo el desarrollo, testeo y correcciones rápidas que facilita la entrega de productos de calidad a tiempo.

### **1.2.2. Justificación aplicativa**

Se implantará un Sistema de Gestión Vehicular ya que éste permitirá automatizar todo proceso, dicha automatización ayuda a optimizar recursos tangibles e intangibles, un recurso intangible y de gran valor es el tiempo, ya que este avanza sin importar lo que ocurra sin poderlo recuperar, por tal razón en el departamento de Mantenimiento de la ESPOCH se pretende mejorar el servicio a través de la implementación de un sistema para Gestionar el control vehicular, el desarrollo de la aplicación está enfocado en brindar menor tiempo de respuesta a los trámites que se realizan en esta unidad a través de la digitalización de procesos

El sistema estará compuesto por 4 módulos generales:

#### **Módulo de Información General**

En este módulo se registrará la información básica del sistema:

- Mecánicos
- Usuarios del Sistema
- Dependencias
- Vehículos
- Repuestos
- Choferes

### **Módulo de Taller**

En este módulo se registrarán los procesos del taller relacionados al mantenimiento vehicular y gestión de bodega:

- Gestión de mantenimientos vehiculares
- Asignación de repuestos y mecánicos al mantenimiento de cada vehículo.
- Administración del ingreso y egreso de repuestos a utilizar en el mantenimiento de cada vehículo.

### **Módulo de Reportes**

En este módulo se presentarán los siguientes reportes :

- Reportes de mantenimiento vehicular
- Reportes de bodega (existencias)
- Reportes de vehículos
- Formatos preimpresos
- Listados de información general

### **Módulo Diagnóstico**

Se presentara un prototipo de un módulo de diagnóstico de fallas mecánicas en vehículos. Esto será útil para técnicos mecánicos, practicantes y personas con conocimientos básicos de mecánica automotriz.

### **1.3. Objetivos**

#### **1.3.1. Objetivo general**

Implementar un sistema de gestión vehicular con la utilización de java server face para el departamento de mantenimiento vehicular de la ESPOCH.

#### **1.3.2. Objetivos específicos**

- Analizar los requerimientos de usuario y analizarlos para plantear alternativas de solución.
- Implementar el sistema de gestión vehicular en el departamento de mantenimiento de la ESPOCH.
- Evaluar la eficiencia y eficacia del Software desarrollado.

### **1.4. Métodos y técnicas**

#### **1.4.1. Métodos**

La elección del método se ha realizado de acuerdo con las necesidades de la investigación para la realización del proyecto se ha establecido aplicar el método cualitativo, ya que figuran la etnografía, los estudios de caso, las entrevistas en profundidad y la observación participativa, esto permite aumentar la validez y fiabilidad de los resultados.

Existen cuatro fases fundamentales en el proceso de investigación cualitativa:

- 1. Preparatoria:** En esta fase inicial de la investigación cualitativa se pueden diferenciar dos grandes etapas: reflexiva y diseño. En la primera etapa el investigador, tomando como base su propia formación investigadora, sus conocimientos y experiencias sobre los fenómenos educativos y, claro está, su

propia ideología, intentará establecer el marco teórico-conceptual desde el que parte la investigación. En la etapa de diseño, se dedicará a la planificación de las actividades que se ejecutarán en las fases posteriores.

2. **Trabajo de campo:** permitirá realizar la recolección de datos e información mediante la aplicación estructurada de preguntas y cuestionarios.
3. **Fase Analítica:** Se establecen tareas como: a) reducción de datos b) Disposición y transformación de datos y c) obtención de resultados y verificación de conclusiones.
4. **Informativa:** El proceso de investigación culmina con la presentación y difusión de los resultados, mediante el desarrollo de un informe cualitativo

En cuanto tiene que ver a la metodología a utilizar será la metodología ágil SCRUM aplicable al desarrollo de software ya que ofrecen una rápida respuesta a cambios de requisitos gracias a su proceso iterativo, permitiendo el desarrollo, testeo y correcciones rápidas que facilita la entrega de productos de calidad a tiempo.

#### **1.4.2. Técnicas de investigación**

La diversidad de fuentes se encuentra a través del internet permitiendo el fácil acceso a la información necesaria respecto a la investigación que se desarrolla. La variedad de las herramientas técnicas que permite la recopilación de información enfocado a las TICs, Para la recopilación de la información necesaria que sustente el presente trabajo de investigación, se ha establecido como técnicas las siguientes:

- Análisis Documental
- Observación
- Pruebas

- Encuestas.
- Estadística Descriptiva.

Análisis documental, permite la recolección de información tales como: manuales, reportes, hojas de codificación, estadísticas y documentos, con este material se puede obtener una imagen de lo que se hizo, se hace y espera hacer.

La observación es el acto de presenciar los acontecimientos en el momento y lugar en que suceden, con el propósito de identificar, aclarar y/o confirmar ciertos aspectos de un ambiente determinado.

Encuesta es una técnica de recopilación de información de un grupo significativo que permite a los estudiar las actitudes, creencias, comportamiento acerca de los problemas de estudio

## CAPÍTULO II

### 2. MARCO TEÓRICO

Java es uno de los lenguajes de programación más utilizados en el mundo. Es así que IDEs como Eclipse y Netbeans permiten su codificación, compilación y ejecución. En el caso específico de Netbeans, es un IDE muy completo y adaptable, con capacidad de integrar una gran cantidad de plugins.

SISCOVE se ha concebido como una aplicación web opensource en java, por lo cual será programado mediante JavaServer Faces (JSF) en el IDE Netbeans. El almacenamiento de datos se realizará en PostgreSQL, motor de base de datos robusto y de gran rendimiento. Considerando además la imperiosa necesidad que tienen los potenciales usuarios en utilizar el sistema, se estimó necesaria aplicación de SCRUM como metodología de desarrollo.

A fin de dar al lector una noción de todas estas tecnologías empleadas, se describirán a continuación sus generalidades.

#### 2.1. Lenguaje de programación Java

##### **Definición**

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de la década de 1990. Este lenguaje toma la estructura de sintaxis (escritura de código fuente) de C y C++ incorporando un modelo de objetos más simples y eliminando las herramientas de bajo nivel (que suele inducir muchos errores como la manipulación de punteros en C y C++). (Peñarrieta, 2011)

## **Características**

Según Peñarrieta (2011) y Belmonte (2005), las principales características de Java son:

- La principal característica de Java es la de ser un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código que se genera bytecode es interpretado por una máquina virtual. De este modo se consigue la independencia de la máquina, el código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma.
- Java es un lenguaje orientado a objetos de propósito general. Aunque Java comenzará a ser conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto.
- Su sintaxis es muy parecida a la de C y C++ pero hasta ahí llega el parecido. Java no es una evolución ni de C++ ni un C++ mejorado.
- Los programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware (multiplataforma)
- Permite muchas funciones simultáneas en una aplicación (multihilo). Los hilos son procesos o piezas independientes dentro de un gran proceso; al ser éstos hilos contruidos en el mismo lenguaje, son más fáciles de usar y más robustos respecto a otros lenguajes.
- Es simple; reduce hasta en un 50% los errores más comunes de programación respecto a otros lenguajes.
- Proporciona librerías y herramientas para que los programas se ejecuten en varias máquinas e interactuando entre ellas (distribuido).

### **2.2. Entorno de Desarrollo Java - IDE Netbeans**

Los IDEs (Integrated Development Environment), tal y como su nombre indica, son entornos de desarrollo integrados. En un mismo programa es posible escribir el código

Java, compilarlo y ejecutarlo sin tener que cambiar de aplicación. Existen distintos programas comerciales que permiten desarrollar código Java. La compañía Sun, creadora de Java, distribuye gratuitamente el Java(tm) Development Kit (JDK). Se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en Java. Incorpora además la posibilidad de ejecutar parcialmente el programa, deteniendo la ejecución en el punto deseado y estudiando en cada momento el valor de cada una de las variables (es el denominado Debugger). Cualquier programador con un mínimo de experiencia sabe que una parte muy importante (muchas veces la mayor parte) del tiempo destinado a la elaboración de un programa se destina a la detección y corrección de errores. Existe también una versión reducida del JDK, denominada JRE (Java Runtime Environment) destinada únicamente a ejecutar código Java (no permite compilar). (García, y otros, 1999)

### **Definición**

Netbeans es un poderoso entorno de desarrollo que permite desarrollar aplicaciones complejas con interacción web, UML, base de datos, aplicaciones para telefonía móvil e inclusive Inteligencia Artificial (IA). Java no cuenta con un entorno de desarrollo propio, por esa razón, se puede utilizar desde un bloc de notas hasta entornos de desarrollo avanzados como NetBeans. (Peñarrieta, 2011)

### **Beneficios**

Según Gimeno & González (2011), los beneficios que ofrece la utilización de Netbeans a un programador son los siguientes:

- Simplifica alguna de las tareas que, sobre todo en proyectos grandes, son tediosas.
- Nos asiste (parcialmente) en la escritura de código, aunque no nos libera de aprender el lenguaje de programación.
- Nos ayuda en la navegación de las clases predefinidas en la plataforma.
- Aunque puede ser costoso su aprendizaje, los beneficios superan las dificultades.

### 2.3. JavaServer Face

JavaServer Faces (JSF) es un framework basado en el patrón MVC (Modelo Vista Controlador) para aplicaciones Java basadas en web, el cual simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente suelen incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (Álvarez Teruelo, Barcenilla Mancha, Rodríguez Gallego, & Salas Fernández, 2008)

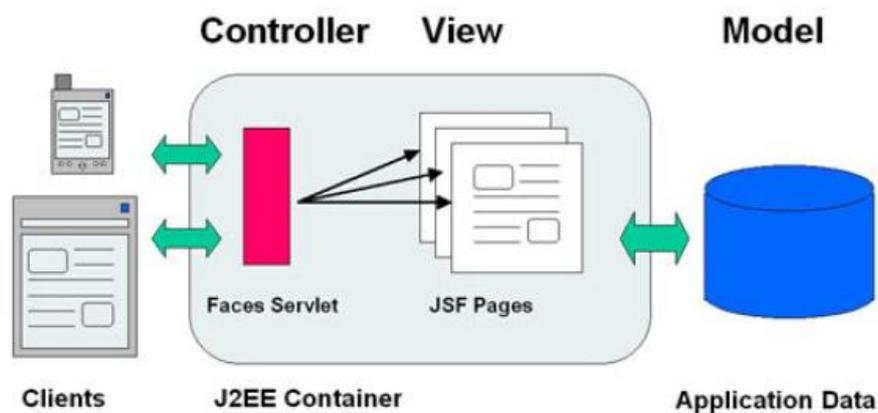
#### Características

Entre sus principales características se encuentran las siguientes (Álvarez Teruelo, Barcenilla Mancha, Rodríguez Gallego, & Salas Fernández, 2008):

- Se ejecuta del lado del servidor
- La especificación JSF fue diseñada según los siguientes principios:
  - Amigable
  - Independiente del protocolo y del dispositivo del cliente
  - Posibilidad de utilizarse con tecnologías JSP
  - Independencia de tecnologías JSP
  - Escalable
  - Posibilidad de utilizarse con navegadores Web actuales
- Principales Componentes (Ferro, Vargas, & Franco, 2006)
  - API e implementación de referencia para representar componentes de interfaz de usuario y manejar su estado, manejo de eventos, validación del lado del servidor y conversión de datos; también permite definir las reglas de navegación entre páginas.
  - Librería de etiquetas JavaServer Pages, personalizadas para dibujar componentes de interfaz dentro de una página JSP.
- Con estos elementos se puede:
  - Conectar eventos generados en el cliente con código de la aplicación en el servidor

- Mapear componentes UI a la página de datos del servidor
- Construir un UI con componentes reutilizables y extensibles
- Grabar y restaurar el estado del UI más allá de la vida de las peticiones del servidor

JSF pretende normalizar y estandarizar el desarrollo de aplicaciones web. Ofrece una clara separación entre el comportamiento y la presentación, lo que permite a cada miembro del equipo de desarrollo de una aplicación Web enfocarse en su parte del proceso de desarrollo, y proporciona un sencillo modelo de programación para enlazar todas las piezas. (Álvarez Teruelo, Barcenilla Mancha, Rodríguez Gallego, & Salas Fernández, 2008)



**Figura 1-2:** Funcionamiento de JSF

Fuente: (Álvarez Teruelo, Barcenilla Mancha, Rodríguez Gallego, & Salas Fernández, 2008)

### Estructura de las aplicaciones JSF

Una aplicación desarrollada con JavaServer Faces maneja generalmente la siguiente estructura (Ferro, Vargas, & Franco, 2006):

- **/JavaSource:** Fuentes de las clases Java
- **/ant:** Scripts de creación (build.xml)
- **/WebContent:** Archivos de la aplicación web usados por el servidor de aplicaciones
- **/Web-INF:** archivos que se utilizan como parte de la rutina de la aplicación web:

- /classes
  - /lib: jsf-impl.jar, jsf-api.jar
  - web.xml
  - faces-config.xml
- **/pages:** archivos jsp para la presentación de la interfaz

## **Ventajas**

Entre sus ventajas se encuentran (Álvarez Teruelo, Barcenilla Mancha, Rodríguez Gallego, & Salas Fernández, 2008):

- Clara separación entre el comportamiento y la presentación
- La separación de la lógica permite a cada miembro del equipo de desarrollo encargarse de su cometido y posteriormente enlazar las partes
- Mejora los conceptos familiares de componente-UI y capa-Web sin limitarnos a tecnología de script particular o lenguaje de marcas
- Aunque incluye librería de etiquetas JSP personalizadas, los APIs de JSF se han creado sobre el API JavaServlet: usar tecnología presentación distinta de JSP, crear componentes personalizados, generar salida para diferentes dispositivos cliente
- Proporciona una rica arquitectura para manejar el estado de los componentes, procesar datos, validar la entrada del usuario y manejar eventos

## **2.4. Primefaces**

### **Definición**

PrimeFaces es una librería de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. PrimeFaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar PrimeFaces, es que permite la integración con otros componentes como por ejemplo RichFaces. (Egas, 2015)

## Propiedades

Para Egas (2015), PrimeFaces tiene las siguientes propiedades:

- Conjunto de componentes (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros).
- Soporte de Ajax con despliegue parcial, lo que permite controlar qué componentes de la página actual se actualizarán y cuáles no.
- Componente para desarrollar aplicaciones web para teléfonos móviles, especiales para iPhone, Palm, Android y teléfonos móviles Nokia.

## 2.5. Modelo Vista Controlador (MVC)

### Definición

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software el cual separa los datos, la lógica de negocio y la interfaz de usuario. MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador; por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Se basa en las ideas de reutilización de código y la separación de conceptos, a fin de facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento (Creative Commons Attribution-Share Alike, 2014).

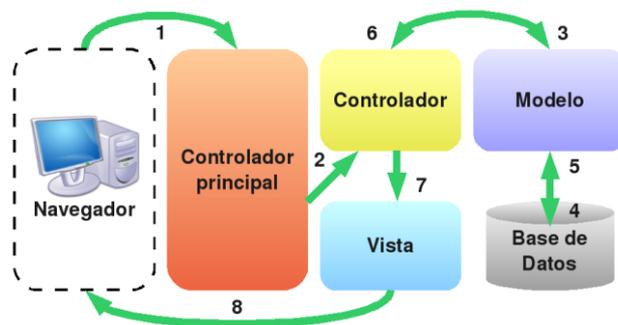
### Descripción del patrón

Los componentes de MVC se pueden definir como sigue (Creative Commons Attribution-Share Alike, 2014):

**Modelo:** representación de la información con la cual el sistema opera; gestiona todos los accesos a dicha información, tanto consultas como actualizaciones. Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

**Controlador:** responde a eventos (acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información. Puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo', por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

**Vista:** presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.



**Figura 2-2:** Modelo Vista Controlador

Fuente: (ForoNSN, 2013)

## 2.6. PostgreSQL

### Definición

El sistema de base de datos PostgreSQL es un administrador de base de datos que a su vez es capaz de soportar un modelo de datos que consiste en una colección de relaciones con nombre los cuales suelen contar con atributos específicos. (Pool Bautista, 2010)

### Ventajas

Entre sus ventajas se pueden señalar (Pool Bautista, 2010):

- Por su diseño, arquitectura y escala va muy bien al aumentarle el número de CPUs y así mismo más de memoria RAM.
- También es capaz de soportar claves ajenas con comprobaciones de integridad referencial.
- Cuenta con un buen y mejor rendimiento para soportar triggers y procedimientos en el servidor.
- Soporta un subconjunto de SQL92 mayor que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.

### **Inconvenientes**

Postgresql tiene algunos inconvenientes en su uso, de los cuales se citan los siguientes (Pool Bautista, 2010):

- Consume BASTANTES más recursos y carga más el sistema.
- Tiene un límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).
- Entre 2 y 3 veces más lento que MySQL.
- Posee menos funciones en PHP.

## **2.7. SCRUM**

### **Definición**

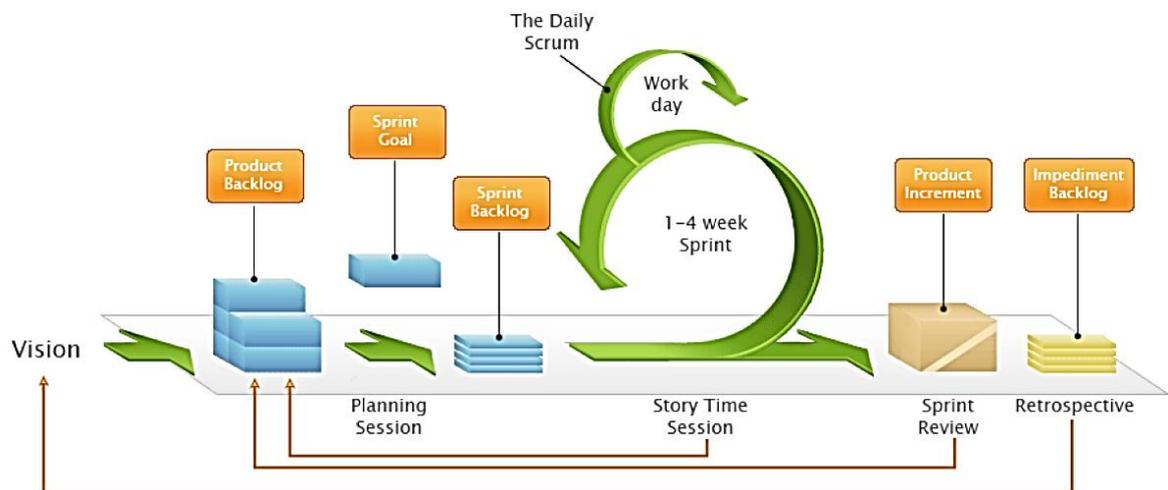
Scrum es una metodología de desarrollo ágil que tiene como base la idea de creación de ciclos breves para el desarrollo (Trigas Gallego, 2012).

Scrum se basa en la teoría de control de procesos empírica o empirismo, la cual asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo (Sutherland & Schwaber, 2013).

### Características de la metodología

SCRUM es un modelo de desarrollo ágil caracterizado por (Palacio, 2014):

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto-organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.



**Figura 3-2:** Metodología SCRUM

Fuente: (Trigas Gallego, 2012)

### Marco Técnico:

Dentro de los aspectos técnicos de esta tecnología, destacan los siguientes (Palacio, 2014):

- **Roles:**

- **Equipo SCRUM:** grupo de profesionales que realizan el incremento de cada sprint.
- **Dueño del producto:** cliente del sistema
- **Scrum Master:** responsable del cumplimiento de las reglas de un marco de SCRUM técnico, asegurando que se entienden en la organización, y se trabaja conforme a ellas. Proporciona la asesoría y formación necesaria al propietario del producto y al equipo. Realiza su trabajo con un modelo de liderazgo servil.

- **Artefactos:**

- **Pila del producto:** (product backlog) lista de requisitos de usuario, que a partir de la visión inicial del producto, crece y evoluciona durante el desarrollo.
- **Pila del sprint:** (sprint backlog) lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- **Sprint:** nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso de avance por tiempos prefijados (time boxing).
- **Incremento:** resultado de cada sprint.

- **Eventos**

- **Reunión de planificación del sprint:** reunión de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el objetivo del sprint y las tareas necesarias para conseguirlo.
- **Scrum diario:** breve reunión diaria del equipo, en la que cada miembro responde a tres cuestiones:

1.- El trabajo realizado el día anterior.

2.- El que tiene previsto realizar.

3.- Cosas que puede necesitar o impedimentos que deben eliminarse para poder realizar el trabajo.

Cada persona actualiza en la pila del sprint el tiempo o esfuerzo pendiente de sus tareas.

- **Revisión del sprint:** análisis e inspección del incremento generado, y adaptación de la pila del producto si resulta necesario.
- **Retrospectiva del sprint:** revisión de lo sucedido durante el Sprint. Reunión en la que el equipo analiza aspectos operativos de la forma de trabajo y crea un plan de mejoras para aplicar en el próximo sprint.

## CAPÍTULO III

### 3. INGENIERÍA DE SOFTWARE

#### 3.1. Especificación de Requerimientos de Software (SRS)

##### 3.1.1. Introducción

###### A. Propósito del documento

La Especificación de Requerimientos de Software (SRS) es la fase fundamental en el proceso de desarrollo de cualquier aplicación. Su objetivo fundamental es comprender el problema a automatizar, partiendo de la indagación de requerimientos específicos, la identificación de los potenciales usuarios del sistema y la determinación de la factibilidad técnica del sistema en torno a la realidad tecnológica del cliente (Méndez, 2008). El presente documento detalla técnicamente la lógica del negocio que pretende informatizarse, a fin de generar una base documentada para futuras reingenierías del sistema SISCOVE. Debe considerarse que, al aplicar la metodología de desarrollo SCRUM, los requerimientos funcionales se describen en función a la “pila del producto” y los futuros “sprints” o iteraciones a efectuarse se detallarán en etapas posteriores de implementación del sistema.

###### B. Alcance del producto

El sistema descrito en el presente documento se identifica con el nombre de Sistema de Control Vehicular (SISCOVE). El sistema limita su funcionalidad a las siguientes actividades:

- Gestionar información general, como: *mecánicos, choferes, vehículos, repuestos/lubricantes/aditivos, usuarios del sistema y dependencias.*
- Gestionar los procesos básicos del taller: *control de bodega y reparación vehicular*
- Generar reportes del sistema, respecto a: *vehículos, bodega y reparación vehicular*

- Generar formatos de impresión, tales como: *formatos generales y formatos con datos preimpresos*
- *Diagnóstico básico de fallas mecánicas.*

El sistema excluye de sus funcionalidades aquellas detalladas a continuación:

- Control de talento humano
- Interfaces hardware-software para el uso de periféricos de control de inventario (lectores de códigos de barra) o de impresión en formatos especiales (adhesivos de códigos de barra).

### **C. Definiciones, Acrónimos y Abreviaturas**

**ADMINISTRADOR:** usuario del sistema que gestiona el acceso de otros tipos de usuarios al sistema y sus roles y tiene acceso total a todos los módulos.

**FORMATOS:** usuario del sistema encargado de la generación e impresión de formatos.

**INVITADO:** usuario general del sistema; tiene acceso únicamente al módulo de diagnóstico vehicular.

**GESTIÓN:** en términos de bases de datos, ejecución de acciones de inserción, modificación y eliminación de registros a través de la interfaz gráfica de un sistema informático.

**INVENTARIO:** lista de bienes que posee la institución en general, o por departamentos.

**MODULO:** parte de un sistema que cumple una funcionalidad macro (como por ejemplo, diagnóstico vehicular, formatos), y es accesible a determinados tipos de usuario.

**PILA DEL PRODUCTO:** formato de inventario de funcionalidades del sistema, utilizado en el desarrollo de sistemas mediante SCRUM.

**SCRUM:** metodología de desarrollo ágil de aplicaciones, basado en sprints o iteraciones y en la generación temprana de versiones entregables del sistema para su validación (Palacio, 2014).

**SPRINT:** iteración o ciclo en el cual se incrementan funcionalidades a un sistema en desarrollo. Término utilizado en la metodología SCRUM (Palacio, 2014).

**USUARIO CORPORATIVO DEL SISTEMA:** personal del Taller Mecánico de la ESPOCH, que tienen acceso al sistema.

## **D. Referencias**

Méndez, G. (2008). *Especificación de Requisitos según el estándar de IEEE 830*. Recuperado de: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

Palacio, J. (2014). *Gestión de proyectos Scrum Manager (Scrum Manager I y II)*. Recuperado de: [http://www.scrummanager.net/files/sm\\_proyecto.pdf](http://www.scrummanager.net/files/sm_proyecto.pdf)

## **E. Descripción del resto del documento**

La sección dos del presente documento refleja una breve descripción del software SISCOVE en torno a su funcionalidad, usuarios y potencial evolución. La sección tres detalla los requerimientos funcionales (pila del producto) y no funcionales del sistema.

### **3.1.2. Descripción general**

#### **A. Perspectiva del producto**

SISCOVE es un sistema independiente en relación a otros sistemas existentes en la organización.

## B. Funciones del producto

**Tabla 1-3.** SRS-Funciones del producto

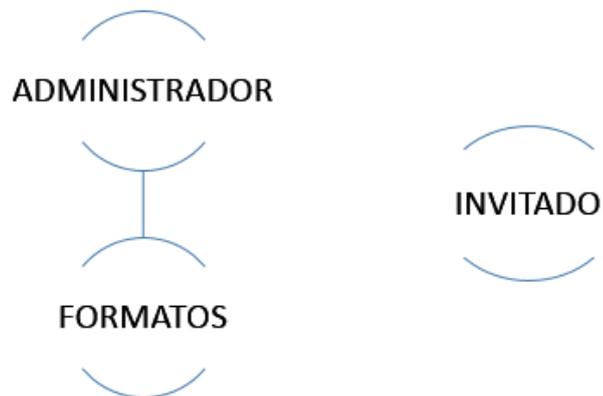
ACTORES Y CARACTERÍSTICAS DEL SISTEMA	FUNCIONES
ADMINISTRADOR	<ul style="list-style-type: none"> <li>• Gestión de Información General</li> <li>• Gestión de Procesos de Taller</li> <li>• Gestión de Reportes</li> <li>• Gestión de Formatos.</li> </ul>
FORMATOS	<ul style="list-style-type: none"> <li>• Gestión de formatos</li> </ul>
INVITADO	<ul style="list-style-type: none"> <li>• Diagnóstico de fallas mecánicas</li> </ul>

**Fuente:** Patricia Herrera y Vanessa Vargas

## C. Características del usuario

Las organización jerárquica de los potenciales usuarios del sistema pueden definirse tal y como lo demuestra la Figura 1-3.

El acceso al sistema se gestionará en función a los tres perfiles de usuario: administrador, formatos, e invitado. Cada usuario podrá realizar únicamente las funciones relacionadas a su perfil (ver Tabla 1-3).



### **Figura 1-3: SRS - Organización Jerárquica de los perfiles de usuario**

Fuente: Patricia Herrera y Vanessa Vargas

#### **D. Restricciones generales**

##### **CONSIDERACIONES ACERCA DE LA SEGURIDAD**

La seguridad informática no es un estado permanente que deba alcanzarse una única vez para su perpetuación; se genera en función a una reingeniería constante de los sistemas. Por tanto, es responsabilidad de la organización mantener al sistema en un permanente proceso de actualización a fin de obtener un sistema web seguro en el tiempo.

##### **CRITICALIDAD DE LA APLICACIÓN**

Aunque todos los procesos son importantes y requeridos en el ámbito de la organización, no existen tareas críticas que no puedan ser registradas tardíamente, en un caso la infraestructura del sistema se viera afectada de algún modo.

#### **E. Suposiciones y Dependencias**

El sistema funciona en una infraestructura basada en la web, mediante un servidor local. Debe suponerse que, al migrarse el sistema a un servidor institucional o externo, el módulo de diagnóstico vehicular será accesible por cualquier usuario de internet, y no únicamente dentro del taller.

#### **F. Evolución previsible del sistema**

Sistema accesible desde un servidor institucional o externo. Incorporación de interfaces de hardware para el control de bodega.

#### **3.1.3. Requisitos Específicos**

Los requisitos han sido desarrollados en función a la metodología a utilizarse (SCRUM). Por tal razón, se ha generado una pila de producto organizada por SPRINTS

y módulos. Cada requerimiento tiene un alto nivel de detalle, al cual se ha agregado información sobre: la estimación del tiempo de desarrollo, prioridad, dependencias y pruebas de aceptación.

## **A. Requisitos comunes de las interfaces**

### **INTERFACES DE USUARIO**

Con la finalidad de mejorar la apariencia visual de las aplicaciones, y así mejorar las interfaces del usuario, se utilizará la librería de componentes visuales de Java PrimeFaces. Se utilizarán elementos de navegación e interacción como: botones, enlaces, tablas, formularios, ventanas flotantes, etc.

**NOTA:** Para un mayor detalle, ver el Anexo III (Arquitectura del Sistema e Interfaz Gráfica).

### **INTERFACES DE HARDWARE**

**Monitor:** presentación de la información del sistema al usuario, así como la respuesta a cada una de las peticiones o acciones del usuario.

**Ratón:** interacción del usuario con la aplicación mediante botones y enlaces.

**Teclado:** ingreso de datos a la aplicación para su validación y procesamiento.

**Impresora:** salida física de información del sistema (reportes)

### **INTERFACES DE SOFTWARE (EQUIPO CLIENTE)**

El usuario del sistema tendrá instalado en su equipo de comunicación (computador) un navegador web y Java.

### **INTERFACES DE SOFTWARE (EQUIPO SERVIDOR)**

Se tendrá instalado en el servidor los siguientes programas: PostgreSQL 9.2 y Glassfish Server 4.1 (deberán configurarse para que arranquen cada vez que se encienda el servidor)

## **INTERFACES DE COMUNICACIÓN**

La comunicación entre el software del cliente (navegador web) y el sistema SISCOVE se realizará mediante el protocolo HTTP. Tanto la base de datos como la aplicación reposarán en el equipo servidor.

**NOTA:** Para un mayor detalle, ver el Anexo III (Arquitectura del Sistema e Interfaz Gráfica).

### **B. Requisitos funcionales**

Se han recabado los requisitos funcionales en base a SPRINTS y MÓDULOS, tal y como puede observarse en el Anexo I. Los requisitos funcionales identificados se listan a continuación:

#### **MODULO 1 – Información General**

**REQF(1.1)** El administrador del sistema debe *gestionar la información de mecánicos* (ingreso, modificación, activación/desactivación), basado en los siguientes datos: CI, Apellido, Nombre, Teléfono Convencional, Teléfono Celular, Fecha de Nacimiento, Tipo de Sangre, Enfermedades y Estado (activo o inactivo).

**REQF(1.2)** El administrador del sistema debe *gestionar la información de choferes* (ingreso, modificación, activación/desactivación), basado en los siguientes datos: CI, Apellido, Nombre, Teléfono Convencional, Teléfono Celular y Estado (activo o inactivo).

**REQF(1.3)** El administrador del sistema debe *gestionar la información de vehículos* (ingreso, modificación, cambio de estado, cambio de disponibilidad), basado en los siguientes datos: Número de vehículo, Dependencia, Tipo de Vehículo (bus o camioneta), Chofer, Año, Marca, Modelo, Número de Motor, Matrícula, Número de Chasis, Placa, Color, Km de Garantía, Estado (activo, inactivo o mantenimiento), Disponibilidad (si o no).

**REQF(1.4)** El administrador del sistema debe *gestionar la información de repuestos/lubricantes* (ingreso, modificación, activación/desactivación, kardex), basado

en los siguientes datos: Código (automático), Nombre, Descripción, Tipo (repuesto o lubricante), Unidad de Medida (litros, unidades, libras), Mínima Cantidad de descarga (1 unidad de medida, 1/4 de unidad de medida, etc), Stock bajo, Stock medio y Estado (activo o inactivo).

**REQF(1.5)** El administrador del sistema debe *gestionar la información de otros usuarios del sistema* (ingreso, modificación, activación/desactivación, seteo de clave), basado en los siguientes datos: CI, Apellidos, Nombres, Usuario, Clave, Tipo (administrador o formatos) y Estado (activo o inactivo).

**REQF(1.6)** El administrador debe *gestionar la información de dependencias* (ingreso, modificación, activación/desactivación), basado en los siguientes datos: Código (automático, Nombre).

## **MODULO 2 – Procesos del taller**

**REQF(2.1)** El administrador del sistema debe *gestionar la bodega* (ingreso, modificación), basado en los siguientes datos: Código (automático), Tipo (Repuesto/Lubricante/Aditivo), Fecha de Ingreso, Cantidad, Descripción, Número de Orden de Entrega.

**REQF(2.2)** El administrador del sistema debe *gestionar la reparación de los vehículos u ordenes de trabajo* (ingreso, modificación, visualización de datos relacionados y dar de alta al vehículo), basado en los siguientes datos: Número de Orden de Trabajo (automático), Vehículo y tipo de reparación (correctiva o preventiva).

**REQF(2.3)** El administrador del sistema debe *gestionar los egresos de bodega por reparaciones* (ingreso, modificación y eliminación), basado en los siguientes datos: Número de Orden de Trabajo, Repuesto/Lubricante/Aditivo, Cantidad y Fecha.

**REQF(2.4)** El administrador del sistema debe *gestionar los mecánicos asignados a una reparación* (ingreso, modificación y eliminación), basado en los siguientes datos: Orden de Trabajo, Mecánico y Fecha de asignación.

**REQF(2.5)** El administrador del sistema debe *gestionar los trabajos mecánicos de una reparación vehicular* (ingreso, modificación y eliminación), basado en los siguientes datos: Orden de Trabajo, Trabajo mecánico a realizar, Efectuado (estado: si o no).

### **MODULO 3 – Reportes del sistema**

**REQF(3.1)** El administrador del sistema debe *generar reportes de vehículos*, con los siguientes filtros de datos: estado, disponibilidad, dependencia, tipo de vehículo. El reporte contendrá toda la información visualizada del **vehículo** en la pantalla.

**REQF(3.2)** El administrador del sistema debe *generar reportes de inventario de lubricantes, repuestos y aditivos*, con los siguientes filtros de datos: código, tipo de elemento (repuesto, lubricante o aditivo), cantidad total en bodega . Formato de salida: pdf. El reporte contendrá la información visualizada de los *repuestos/lubricantes/aditivos* en la pantalla. Mediante una opción de pantalla, permitirá obtener el kardex del producto con datos específicos de ingresos y egresos de bodega, también en versión imprimible.

**REQF(3.3)** El administrador del sistema debe *generar reportes de reparación vehicular - mecánica*, con los siguientes filtros de datos: fecha de ingreso a la mecánica (entre fechas), fecha de salida (entre fechas), mecánico, vehículo, tipo de reparación, detección de accidente (si/no), trabajos efectuados, trabajos solicitados. Formato de salida: pdf. El reporte contendrá, por cada reparación vehicular, la información de: *número de orden de trabajo, vehículo (placa y marca), tipo de reparación, fechas de ingreso y salida, detección de accidente.*

**REQF(3.4)** El administrador del sistema debe *generar e imprimir un reporte de reparación vehicular individual* (de cada vehículo/orden de trabajo), para efectos administrativos. Formato de salida: pdf. El reporte contendrá la siguiente información: *vehículo, mecánico(s) asignados, reparación (datos generales y trabajos mecánicos aplicados), chofer responsable.*

**REQF(3.5)** El administrador del sistema debe *generar reportes de reparación vehicular - inventario*, con los siguientes filtros de datos: tipo (repuesto, lubricante o aditivo), repuesto/lubricante/aditivo, fecha de descarga de inventario (entre fechas).

Formato de salida: pdf. El reporte contendrá la siguiente información: número de orden de trabajo, nombre del repuesto/lubricante/aditivo, tipo de repuesto, cantidad, unidad, fecha de salida.

#### **MODULO 4 – Formatos**

**REQF(4.1)** Los usuarios formatos del sistema y administrador del sistema deben *generar/descargar formatos de documentos administrativos*, para su impresión: "Orden de Entrega de Lubricantes y Repuestos", "Adquisición o Partes y Repuestos de Vehículos o Bienes Materiales", en formatos .doc y .pdf.

**REQF(4.2)** Los usuarios "formatos" y "administrador" del sistema deben *generar un "Informe Técnico de Vehículos" , "Informe de Reparación" y "Orden de Entrega de Repuestos/Lubricantes/Aditivos"* con información preimpresa, en base a un listado de vehículos y ordenes de trabajo registrados en el sistema y varios filtros de información.

#### **MODULO 5 – Diagnóstico Vehicular**

**REQF(5.1)** El mecánico debe *realizar el diagnóstico básico de las fallas mecánicas de un vehículo relacionadas a los sistemas de freno, embrague o motor*, mediante la contestación a un conjunto finito de preguntas pre-establecidas. El sistema presentará por defecto el problema y la solución PRIMARIA o MÁS PROBABLE. En caso de que la solución planteada no corrija la falla del vehículo, se presentará un listado de problemas y soluciones SECUNDARIAS O MENOS PROBABLES. No se contemplan opciones de retroalimentación para el sistema de reglas.

### **C. Requisitos no funcionales**

#### **Disponibilidad**

- **RQNF-D(001):** Proveer a los usuarios altos índices de funcionamiento del sistema sin interrupciones.

## **Mantenibilidad y Escalabilidad**

- **RQNF-M(001):** Estructurar programáticamente al sistema a fin de se puedan corregir errores de manera eficiente, mediante la revisión preliminar de la documentación técnica generada.
- **RQNF-M(002):** Estructurar programáticamente al sistema a fin de que puedan ser incluidos nuevos módulos en el sistema, según lo requiera la organización.

## **Accesibilidad**

- **RQNF-A(001):** Permitir la visualización de la información en tamaños de fuente más agrandados que los predeterminados.

### **3.2. Implementación del Sistema**

La metodología de desarrollo ágil de software SCRUM fue utilizada para la implementación del sistema. Se consideraron las siguientes etapas de desarrollo:

1. **FASE I - PLANIFICACIÓN:** dentro de esta fase se desarrollaron las actividades de Análisis y Diseño del software, en base a una “pila de requerimientos” o “pila del producto” elaborada conjuntamente con el cliente. Dicho documento contiene, entre otras cosas: la estimación del tiempo de desarrollo, prioridad, dependencias y pruebas de aceptación de cada requerimiento.
2. **FASE II - DESARROLLO DE SPRINTS:** en esta fase se desarrolló un proceso iterativo basado en sprints. En cada sprint se generó un producto potencialmente entregable, el cual iba “acumulando” todas las funcionalidades de anteriores sprints mientras era validado por el cliente final del sistema. Este proceso se realizó en un servidor de producción.
3. **FASE III - CIERRE:** se efectuaron pruebas generales del funcionamiento del sistema en el entorno de producción del sistema. Dentro de esta fase se elaboraron la documentación técnica y el manual de usuario.

### 3.2.1. Fase I: Planificación

#### Análisis y Diseño de Software - Documentos Técnicos

Los documentos técnicos elaborados en esta fase se encuentran detallados en la tabla 2-3. Su ubicación dentro del presente documento se encuentra detallada en la columna “Anexo”.

**Tabla 2-3.** Documentación técnica desarrollada – SISCOVE

ETAPAS DE INGENIERÍA DE SOFTWARE	DOCUMENTACIÓN DESARROLLADA SISTEMA TRANS-FB	ANEXO
ANÁLISIS	Pila del producto e historias de usuarios	Anexo I
	Diagramas de Modelado UML	Anexo II
DISEÑO	Arquitectura del Sistema e Interfaz Gráfica	Anexo III
	Estructuras de Datos – Base de Datos SISCOVE	Anexo IV

**Fuente:** Patricia Herrera y Vanessa Vargas

#### Gestión del equipo SCRUM

La conformación del equipo SCRUM, con sus diversos roles, se encuentra definida en la Tabla 3-3:

**Tabla 3-3.** Equipo SCRUM – roles y asignaciones

ROL	ASIGNACIÓN	OBSERVACIONES
SCRUM MASTER	Ing. Raúl Rosero	DIRECTOR DE LA TESIS
PRODUCT OWNER	Tlgo. Gustavo Tapia	JEFE DE TALLER DE MECÁNICA DE LA ESPOCH
TEAM	Patricia Herrera (TS1) Vanessa Vargas (TS2)	TESISTAS

**Fuente:** Patricia Herrera y Vanessa Vargas

### 3.2.2. Fase II: Desarrollo de sprints

**Duración por SPRINTS:** 2 semanas (10 días laborables)

**Trabajo diario:** 16 horas / día (TEAM: 2 miembros trabajando 8 horas al día)

Para el desarrollo de la pila del producto se efectuaron 2 reuniones de trabajo entre el equipo SCRUM y el cliente del sistema (ver Anexo I). Con relación a la versión final de ese documento (versión 2), se planificó el desarrollo de los sprints en función a lo detallado en la tabla 4-3. Se detallan los responsables de la implementación y de la ejecución de las pruebas de aceptación de cada sub-módulo, así como el tiempo estimado de desarrollo (en semanas).

**Tabla 4-3.** Planificación de sprints

ID SPRINT	MÓDULOS	SUBMÓDULOS	RESPONSABLE	ESTIMACIÓN (SEMANAS)
1	INFORMACIÓN GENERAL	1. MECÁNICOS 2. CHOFERES 3. VEHÍCULOS	TS1 (1 SEMANA)	1 *
		4. REPUESTO/LUBRICANTE/ADITIVO 5. USUARIOS DEL SISTEMA 6. DEPENDENCIAS	TS2 (1 SEMANA)	
2	PROCESOS DEL TALLER	7. BODEGA (INVENTARIO) 8. REPARACIÓN VEHICULAR (REQ. 2.2, 2.3, 2.4)	TS1 (2 SEMANAS)	2 *
		9. REPARACIÓN VEHICULAR (REQ. 2.5, 2.6)	TS2 (2 SEMANAS)	
3	REPORTES DEL SISTEMA	10. VEHÍCULOS 11. BODEGA (INVENTARIO)	TS1 (0.5 SEMANAS)	1 *
	FORMATOS	12. REPARACIÓN VEHICULAR	TS2 (1 SEMANA)	
		13. FORMATOS GENERALES 14. FORMATOS CON DATOS PREIMPRESOS	TS1 (0.5 SEMANAS)	
4	DIAGNÓSTICO VEHICULAR	15. DIAGNÓSTICO BÁSICO DE FALLAS MECÁNICAS	TS1 (0.5 SEMANAS) TS2 (0.5 SEMANAS)	0.5 *
* TRABAJO DESARROLLADO EN PARALELO			<b>TOTAL</b>	<b>4.5 SEMANAS</b>

Fuente: Patricia Herrera y Vanessa Vargas

Para generar el Spring Backlog de este proyecto, los miembros del TEAM SCRUM aplicaron la siguiente matriz de seguimiento por cada sub-módulo a desarrollar:

**Tabla 5-3.** Matriz de Seguimiento - Sprint Backlog

ID SPRINT	1	RESPONSABLE	TS1	VERSION #	1
MÓDULO	INFORMACIÓN GENERAL		SUBMÓDULO	Mecánicos	
FECHA INICIO	04-04-2016		FECHA FIN	04-04-2016	
REQ.	DURACIÓN (DÍAS)				
	BASE DE DATOS	LÓGICA DE NEGOCIOS	INTERFAZ GRÁFICA	PRUEBAS DE ACEPTACIÓN	TOTAL
SUB- MÓDULO	MECÁNICOS				
1.1	0.25	0.25	0.25	0.25	1
SUB- MÓDULO	CHOFERES				
1.2	0.25	0.25	0.25	0.25	1
....	.....				
	<b>TOTAL DIAS</b>				
OBSERVA- CIONES					

Fuente: Patricia Herrera y Vanessa Vargas

En la tabla 7-3 se resume el desarrollo de cada uno de los sprints del proyecto y el IDP (incremento del producto), así como su relación con las otras fases de la metodología aplicada.

### 3.2.3. Fase III: Cierre

En esta etapa se desarrollaron varias pruebas de integración del sistema en su ambiente de producción. Cada módulo fue testeado individualmente y en su interacción con los demás módulos del sistema, según el cronograma de la tabla 6-3.

**Tabla 6-3.** Pruebas de integración del sistema

<b>N°</b>	<b>PRUEBAS DE INTEGRACIÓN</b>	<b>FECHA INICIO</b>	<b>FECHA FIN</b>
<b>1</b>	MÓDULO 1	02/06/2016	02/06/2016
<b>2</b>	MÓDULO 2	03/06/2016	03/06/2016
<b>3</b>	MÓDULO 3	04/06/2016	04/06/2016
<b>4</b>	MÓDULO 4	05/06/2016	05/06/2016
<b>5</b>	MÓDULO 5	06/06/2016	06/06/2016
<b>6</b>	ELABORACIÓN DE MANUALES	09/06/2016	13/06/2016

**Fuente:** Patricia Herrera y Vanessa Vargas

En la Tabla 7-3 se resume la aplicación de la metodología en el desarrollo del sistema SISCOVE.

**Tabla 7-3.** Desarrollo de la metodología SCRUM – tiempos y responsables

ETAPAS SCRUM	PROCESO	MÓDULOS	RESPONSABLES	FECHA INICIO	FECHA ENTREGA	DURACIÓN	OBSERVACIONES
PLANIFICACIÓN	ANÁLISIS	TODOS	TS1, TS2, SCRUM MASTER	04/04/2016	15/04/2016	10 DIAS	CORRECTO
	DISEÑO	TODOS	TS1, TS2, SCRUM MASTER	18/04/2016	29/04/2016	10 DIAS	CORRECTO
DESARROLLO DE SPRINTS	SPRINT 1	TODOS	TS1 , TS2	02/05/2016	06/05/2016	5 DIAS	CORRECTO
	SPRINT 2	TODOS	TS1 , TS2	09/05/2016	18/05/2016	8 DIAS	CORRECTO
	IDP 1	SPRINTS 1+2	TS1 , TS2	19/05/2016	20/05/2016	2 DIAS	CORRECTO
	SPRINT 3	1	TS1, TS2	23/05/2016	26/05/2016	4 DIAS	CAMBIO REP. BODEGA
		2	TS1	23/05/2016	26/05/2016		CORRECTO
	IDP 2	SPRINTS 1+2+3	TS1 , TS2	27/05/2016	27/05/2016	1 DIAS	CORRECTO
	SPRINT 4	5	TS1 , TS2	30/05/2016	31/05/2016	2 DIAS	CAMBIO INTERFAZ SISTEMA .EXP.

	<b>IDP 3</b>	SPRINTS 1+2+3+4	TS1 , TS2	01/06/2016	01/06/2016	1 DIA	CORRECTO
<b>CIERRE</b>	<b>PRUEBAS</b>	TODOS	TS1 , TS2, SCRUM MASTER, PRODUCT OWNER	02/06/2016	06/06/2016	5 DIAS	CORRECTO
	<b>DOCUMENTACIÓN</b>	TODOS	TS1 , TS2	09/06/2016	13/06/2016	5 DIAS	MANUALES COMPLETOS
<b>*IDC: INCREMENTO DEL PRODUCTO</b>						<b>TOTAL</b>	<b>53 DIAS</b>

**Fuente:** Patricia Herrera y Vanessa Vargas

### 3.2.4. Software utilizado en la implementación

En la tabla 8-3 se detalla las características generales del software utilizado en la etapa de Desarrollo de Sprints.

**Tabla 8-3.** Software utilizado en la implementación del Sistema SISCOVE

PROPÓSITO	SOFTWARE	VERSIÓN	LIBRERIAS ADICIONALES
BASE DE DATOS	Toad Data Modeler (Modelado de Base de Datos)	5.3	Ninguna
	pgAdmin III (Administrador de Base de Datos)	1.22.1	Ninguna
	postgreSQL (Sistema de Gestión de Base de Datos)	9.2	Ninguna
PROGRAMACIÓN WEB EN JAVA	Netbeans IDE (Entorno de Desarrollo Integrado Libre para JAVA)	8.0.2	Primefaces
	Glassfish Server (Versión 4.1) (Servidor de Aplicaciones)	4.1	Ninguna

**Fuente:** Patricia Herrera y Vanessa Vargas

### 3.2.5. Implementación del Módulo de Diagnóstico Vehicular

Para la implementación del módulo de diagnóstico vehicular, se efectuaron las siguientes actividades:

- a. **Aplicación de encuestas y su tabulación:** a fin de determinar las características principales del módulo
- b. **Definición de la lógica de negocios del módulo:** para establecer la secuencia de procesos que debe seguir el usuario hasta llegar al diagnóstico de su vehículo.

- c. **Definición de reglas/preguntas de diagnóstico:** a fin de ingresarlas a la base de datos y probar el funcionamiento del sistema

### Aplicación de encuestas y su tabulación

Las encuestas fueron aplicadas a los estudiantes de 7mo, 8vo y 9no semestre de la Escuela de Ingeniería Automotriz, considerando que las prácticas pre-profesionales se realizan a partir de 7mo semestre. La cantidad de estudiantes matriculados en cada uno de estos semestres durante el periodo académico en curso puede observarse en la siguiente tabla:

**Tabla 9-3.** Estudiantes matriculados en la Escuela de Ingeniería Automotriz

SEMESTRE	CANTIDAD ESTUDIANTES
7mo	18
8vo	50
9no	20
<b>TOTAL</b>	<b>88</b>

**Fuente:** Patricia Herrera y Vanessa Vargas

Se procedió al cálculo de la muestra (n), mediante la siguiente fórmula:

$$n = \frac{k^2 N p q}{e^2 (N - 1) + k^2 p q}$$

Donde,

**N** = tamaño de la población = 88 tablas

**n** = tamaño de la muestra = ?

**e** = Límite aceptable de error muestral = 5% (nivel de confianza del 95%)

**k** = nivel de confianza = 1,96

**p** = proporción de individuos que poseen en la población la característica de estudio = 0,5

$q$  = proporción de individuos que no poseen esa característica, es decir, es  $1-p = 0,5$

Aplicando la fórmula se obtuvo lo siguiente:

$$n = \frac{1,96^2 * 88 * 0,5 * 0,5}{(0,05)^2 * (88 - 1) + 1,96^2 * 0,5 * 0,5}$$

$$n = \frac{84,52}{0,22 + 0,96}$$

$$n = 71,62 = 72$$

La muestra de investigación, para un nivel de confianza del 95%, es de 72 estudiantes. Se procedió a efectuar la encuesta con los 88 elementos de la población, ya que la diferencia entre la población y la muestra calculada no es significativa (el método de discriminación resulta difícil de determinar).

La encuesta constó de 3 preguntas (2 abiertas y 1 cerrada), la cual puede observarse en el Anexo V de este documento. Los resultados obtenidos pueden resumirse en las siguientes tablas:

**Tabla 10-3.** Pregunta 1 - Tabulación

ÁREAS DE DIAGNÓSTICO VEHICULAR	CANTIDAD ESTUDIANTES
MOTOR	75
EMBRAGUE	80
FRENOS	63
OTROS	5

Fuente: Patricia Herrera y Vanessa Vargas

En base al análisis del contenido de la tabla anterior, se consideraron las siguientes categorías de diagnóstico: MOTOR, EMBRAGE Y FRENOS

**Tabla 11-3.** Pregunta 2 - Tabulación

ÁREAS DE DIAGNÓSTICO VEHICULAR	CANTIDAD ESTUDIANTES
VIDEOS	88
OTRAS ALTERNATIVAS DE DIAGNÓSTICO MENOS PROBABLES	70
MANUALES DESCARGABLES	10
OTROS	5

Fuente: Patricia Herrera y Vanessa Vargas

En base al análisis del contenido de la tabla anterior, se consideró la presentación de varias alternativas de diagnóstico y la inclusión de videos multimedia.

**Tabla 12-3.** Pregunta 3 - Tabulación

ES ÚTIL DEL MÓDULO A IMPLEMENTARSE	CANTIDAD ESTUDIANTES
SI	63
NO	25
<b>TOTAL</b>	88

Fuente: Patricia Herrera y Vanessa Vargas

La mayoría de estudiantes consideraron útil la implementación del módulo de diagnóstico vehicular (71,59%). Entre las principales razones se encuentran: “aprendizaje”, “retroalimentación” y “validación de conocimientos”.

El 28,41% de los estudiantes consideraron que no sería útil. Entre las principales razones se encuentran: “existencia de software especializado por marcas de vehículos” y la “no adaptabilidad en un 100% a circunstancias reales”.

En función a los resultados obtenidos (mayoritaria aceptación de los clientes potenciales), se continuó con el proceso.

### **Definición de la lógica de negocios del módulo**

El funcionamiento del módulo de diagnóstico responderá al siguiente algoritmo:

- a. INICIO
- b. LISTAR CATEGORIAS DE DIAGNÓSTICO
- c. <<USUARIO>> SELECCIONAR UNA CATEGORÍA E INICIA EL PROCESO DE DIAGNÓSTICO
- d. OBTENER PREGUNTAS DE LA CATEGORÍA
- e. POR CADA PREGUNTA
  - a. Presentar Enunciado y Opciones de Respuesta al usuario
  - b. Almacenar respuestas
- f. OBTENER LOS POSIBLES PROBLEMAS Y PROBABILIDADES ACORDE A LO RESPONDIDO
- g. FIN

En el Anexo II, se puede observar el diagrama de secuencias con un mayor nivel de detalle (CU016 – DIAGNOSTICAR FALLAS MECÁNICAS)

### **Definición de reglas/preguntas de diagnóstico**

Para la definición de preguntas, se recurrió a la ayuda de 3 expertos en la rama:

1. Tglo. Gustavo Tapia, Jefe de Taller de Mecánica de la ESPOCH.
2. Manuel Vilan, Auxiliar de Taller de Mecánica de la ESPOCH.
3. Juan Carlos Rivera, Mecánico Particular

Se realizó un proceso de revisión incremental con cada uno de ellos, habiéndose adquirido rectificaciones y mejoras en las preguntas en cada incremento.

Luego de establecidas las preguntas, se procedió a la implementación del módulo y luego a su validación. Para el efecto, se permitió la utilización del módulo a 20

estudiantes de 9no semestre en un laboratorio de informática de la facultad, obteniéndose los siguientes resultados:

**Tabla 13-3.** Validación de preguntas y lógica de negocios – módulo de diagnóstico vehicular

<b>VALORACIÓN DEL PROCESO</b>	<b>CANTIDAD ESTUDIANTES</b>
ALTA	62
MEDIA	15
BAJA	11
<b>TOTAL</b>	88

**Fuente:** Patricia Herrera y Vanessa Vargas

Luego de la valoración, se solicitó a los participantes entregaran sus apreciaciones generales. Los principales hallazgos fueron:

- 9,09% de los estudiantes (8 estudiantes en total) dijeron no estar de acuerdo con algunos de los diagnósticos finales obtenidos en el sistema.
- Apenas el 2,27% (2 estudiantes) tuvieron dificultades en el manejo del sistema. No comprendieron la lógica de negocios.

## CAPÍTULO IV

### 4. EVALUACIÓN DE LA EFICIENCIA Y EFICACIA DEL SOFTWARE DESARROLLADO

A fin de determinar el efecto causado por el sistema implantado, se procedió a comparar la eficiencia (en términos de tiempo) de los procesos realizados manualmente y con el sistema. Para ello se realizó la medición del tiempo requerido para la realización de una determinada tarea, por cada módulo y sub-módulo del sistema SISCOVE. Los resultados, en detalle, se encuentran en las tablas 1-4, 2-4, 3-4, 4-4. Un resumen de estos resultados se presenta en la tabla 5-4. Cabe recalcar que no se ha considerado en el presente análisis al módulo de “información general”, pues se relaciona con procesos que, en su mayoría, no se realizan manualmente.

**Tabla 1-4.** Evaluación de Eficiencia – Procesos del Taller

MÓDULO	SUBMODULO	ID REQ.	TIEMPO PROCESOS MANUAL	TIEMPO SISTEMA
<b>2. PROCESOS DEL TALLER</b>	BODEGA (INVENTARIO)	2.2	1080 seg	30 seg
	REPARACIÓN VEHICULAR	2.3	1200 seg	30 seg
		2.4	1080 seg	30 seg
		2.4	600 seg	45 seg
		2.5	900 seg	45 seg
		<b>PROMEDIO</b>		<b>972 seg</b>

**Fuente:** Patricia Herrera y Vanessa Vargas

Como puede observarse en cada una de las tablas, el sistema es mucho más eficiente que el proceso manual, pues requiere de un menor tiempo para la realización de las

actividades. Se debe considerar además el aspecto de la eficacia que, aunque no puede ser cuantificado para los procesos no sistematizados pues intervienen factores externos, es mucho mayor en procesos automatizados, pues evita ciertos errores comunes de digitación, asignación de recursos, contabilización, entre otros. La organización y centralización de la información que se obtienen con el manejo del sistema, son también factores que deben considerarse dentro de la evaluación.

**Tabla 2-4.** Evaluación de Eficiencia – Reportes del Sistema

MÓDULO	SUBMÓDULO	ID REQ.	TIEMPO PROCESOS MANUAL	TIEMPO SISTEMA
<b>3. REPORTES DEL SISTEMA</b>	VEHICULOS	3.1	1200 seg	10 seg
	BODEGA (INVENTARIO)	3.4	1200 seg	10 seg
	REPARACIÓN VEHICULAR	3.5	900 seg	10 seg
		3.6	900 seg	5 seg
		3.7	1080 seg	10 seg
	<b>PROMEDIO</b>			<b>1056 seg</b>

Fuente: Patricia Herrera y Vanessa Vargas

**Tabla 3-4.** Evaluación de Eficiencia – Formatos

MÓDULO	SUBMÓDULO	ID REQ.	TIEMPO PROCESOS MANUAL	TIEMPO SISTEMA
<b>4. FORMATOS</b>	FORMATOS GENERALES	4.1	900 seg	5 seg
	FORMATOS CON DATOS PREIMPRESOS	4.2	480 seg	10 seg
<b>PROMEDIO</b>			<b>690 seg</b>	<b>7,5 seg</b>

Fuente: Patricia Herrera y Vanessa Vargas

**Tabla 4-4.** Evaluación de Eficiencia – Diagnóstico vehicular

MÓDULO	SUBMÓDULO	ID REQ.	TIEMPO PROCESOS MANUAL	TIEMPO SISTEMA
5. DIAGNÓSTICO VEHICULAR	DIAGNÓSTICO BÁSICO DE FALLAS MECÁNICAS	5.1	1800 seg	45 seg
PROMEDIO			1800 seg	45 seg

Fuente: Patricia Herrera y Vanessa Vargas

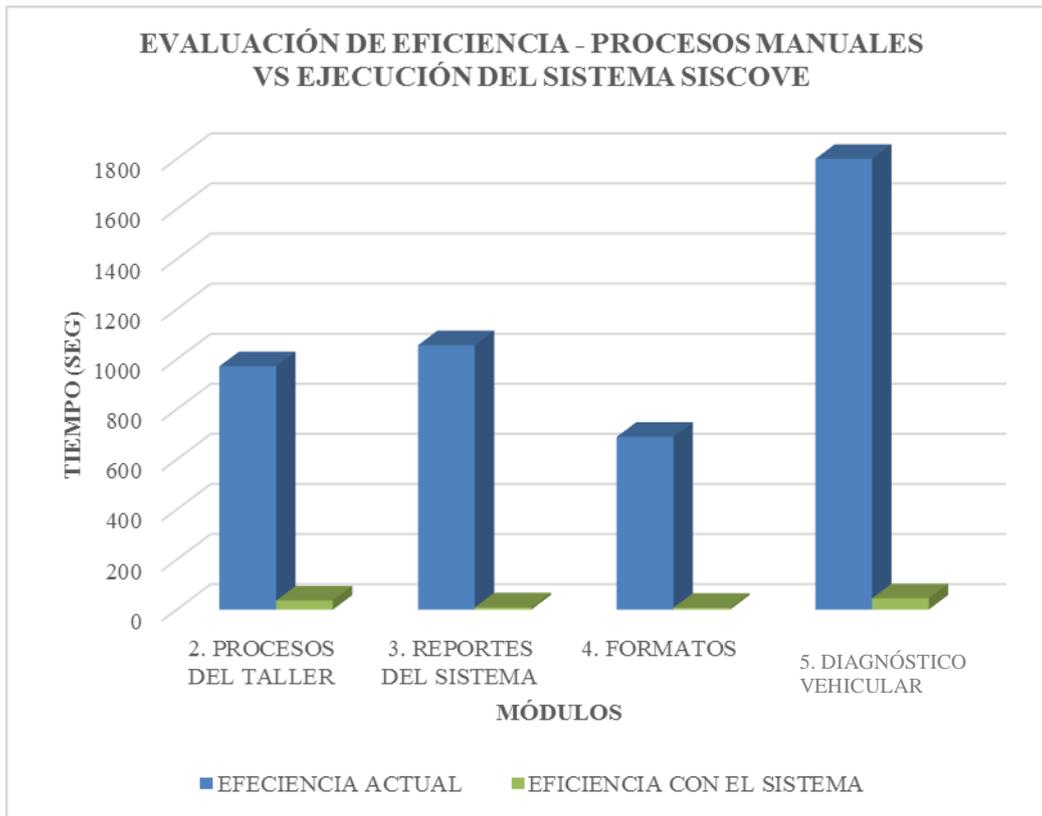
Si se analizan los tiempos globales, un proceso manejado por el sistema puede realizarse en un tiempo promedio de 24,38 segundos (ver tabla 5-4). Si se compara con los 1129,50 segundos (casi 19 minutos) en promedio que toma la realización de un proceso manual, se puede concluir que el sistema mejora sustancialmente la eficiencia de los procesos, reduciendo los tiempos de ejecución en un promedio del 98%.

**Tabla 5-4.** Evaluación de Eficiencia – Resumen

MÓDULO	TIEMPO PROCESOS MANUAL	SISTEMA	
		TIEMPO SISTEMA	% RESPECTO AL PROCESO MANUAL
2. PROCESOS DEL TALLER	972 seg	36 seg	3,70%
3. REPORTES DEL SISTEMA	1056 seg	9 seg	0,85%
4. FORMATOS	690 seg	7,5 seg	1,09%
5. DIAGNÓSTICO VEHICULAR	1800 seg	45 seg	2,50%
<b>PROMEDIO GENERAL DEL SISTEMA</b>	<b>1129,50 seg</b>	<b>24,38 seg</b>	<b>2,16%</b>

Fuente: Patricia Herrera y Vanessa Vargas

Como puede observar en la figura 1-4, todos los procesos son mucho más eficientes con el sistema. Los módulos que requieren un mayor tiempo de ejecución con el sistema son “procesos del taller” y el “diagnóstico vehicular”. Los procesos con mayor disminución porcentual de tiempo promedio son “reportes del sistema” y “formatos”.



**Figura 1-4:** Evaluación de Eficiencia – procesos manuales vs ejecución del sistema SISCOVE

**Fuente:** Patricia Herrera y Vanessa Vargas

## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

- En base a varias entrevistas con el cliente del sistema, se logró determinar la necesidad de automatizar los procesos de gestión de información del taller mecánico de la ESPOCH. El sistema debería incluir los siguientes módulos básicos: “Información General”, “Procesos del Taller”, “Reportes del Sistema” y “Generación de Formatos”. El sistema debería además, incluir un sistema de diagnóstico de fallas mecánicas. Se implementaría en un computador local, sin acceso a internet, como una aplicación web con JSF.
- La metodología de desarrollo ágil de software SCRUM fue aplicada durante la ingeniería del software. La pila del producto fue verificada en un total de dos ocasiones, luego de lo cual fue aceptada por el cliente y el equipo de desarrollo. El uso de sprints permitió evaluar las funcionalidades del sistema de manera incremental.
- El Sistema de Control Vehicular (SISCOVE) fue implantado satisfactoriamente en el departamento de mecánica de la ESPOCH, instalándose en el único equipo disponible como cliente-servidor del sistema. Luego de dicho proceso se efectuaron las respectivas pruebas en ambiente de producción.
- La implementación del sistema ha mejorado sustancialmente la eficiencia de los procesos, reduciendo los tiempos de ejecución en un promedio del 98%. Los procesos automatizados que presentaron una mayor disminución porcentual de su tiempo promedio fueron “reportes del sistema” (99,15%) y “formatos” (98,91%). Además pudo determinarse que los módulos automatizados que requieren un mayor tiempo de ejecución son “procesos del taller” (36 seg) y “diagnóstico vehicular” (45 seg).

## RECOMENDACIONES

- La implantación de SISCOVE en uno de los servidores institucionales permitiría el acceso general a su Módulo de Diagnóstico Vehicular, especialmente para quienes estudian la carrera de Mecánica.
- La especialización del Módulo de Diagnóstico Vehicular (por marca de vehículo, por ejemplo), así como la implementación de interfaces hardware para el control de bodega (con lectores de código de barra), pueden considerarse futuras mejoras para la version 2.0 de SISCOVE.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA GENERAL

- Escuela de Ingeniería en Mantenimiento. (2012). Actualización Curricular 2012-2013. Riobamba: ESPOCH, Facultad de Mecánica, Escuela de Ingeniería en Mantenimiento. ESPOCH. (1969). visión. Obtenido de [www.espoch.edu.ec](http://www.espoch.edu.ec)

### BIBLIOGRAFÍA RELACIONADA AL TEMA

- Álvarez Teruelo, S., Barcenilla Mancha, M., Rodríguez Gallego, C., & Salas Fernández, A. (2008). *Java Server Faces (JSF)*. Obtenido de <http://www.it.uc3m.es/spickin/docencia/comsoft/presentations/spanish/07-08/JSF.pdf>
- Belmonte Fernández, O. (2005). *Introducción al lenguaje de programación Java: Una guía básica*. Obtenido de <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>
- Creative Commons Attribution-Share Alike. (2014). *Modelo vista controlador*. Obtenido de [http://material.concursos.econo.unlp.edu.ar/concursos/T%C3%A9cnico-Profesional%20\(Inform%C3%A1tica\)/patrones/Modelo%20%93vista%20%93controlador.pdf](http://material.concursos.econo.unlp.edu.ar/concursos/T%C3%A9cnico-Profesional%20(Inform%C3%A1tica)/patrones/Modelo%20%93vista%20%93controlador.pdf)
- Egas, P. (2015). *Primefaces CRUD Generador para Netbeans*. Obtenido de <http://www.dspace.uce.edu.ec/bitstream/25000/4310/1/T-UCE-0011-166.pdf>
- Ferro, S., Vargas, D., & Franco, F. (2006). *Java Server Faces*. Obtenido de <http://proyectoremar.tripod.com/Documentos/Herramientas/JavaServerFaces.pdf>
- ForoNSN. (2013). *POO y MVC en PHP*. Obtenido de <http://www.foronsn.com/Thread-POO-y-MVC-en-PHP-PDF-ya-no-ienes-excusa>
- García, J., Rodríguez, J., Mingo, I., Imaz, A., Brazález, A., Larzabal, A., . . . García, J. (1999). *Aprenda Java como si estuviera en primero*. Obtenido de <http://www.upv.es/sma/java/manuales/Aprenda%20Java.pdf>
- Gimeno, J., & González, J. (2011). *Introducción a Netbeans*. Obtenido de <http://ocw.udl.cat/enginyeria-i-arquitectura/programacio-2/continguts-1/1-introduccioi81n-a-netbeans.pdf>

- Palacio, J. (2014). *Gestión de proyectos Scrum Manager*. Obtenido de [http://www.scrummanager.net/files/sm\\_proyecto.pdf](http://www.scrummanager.net/files/sm_proyecto.pdf)
- Peñarrieta, R. (2011). *Java y NetBeans*. Obtenido de [https://javaagricola.wikispaces.com/file/view/0\\_Java+y+NetBeans.pdf](https://javaagricola.wikispaces.com/file/view/0_Java+y+NetBeans.pdf)
- Pool Bautista, L. (2010). *Ensayo de comparaciones de Postgresql vs Mysql vs Oracle*. Campeche. Obtenido de <https://alberto121888.files.wordpress.com/2010/08/ensayo-postgresql-vs-mysql-vs-oracle.pdf>
- Sutherland, J., & Schwaber, K. (2013). *La Guía Definitiva de Scrum: Las Reglas del Juego*. Obtenido de <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Trigas Gallego, M. (2012). *Metodología Scrum*. Obtenido de <http://www.quimbiotec.gob.ve/sistem/auditoria/pdf/ciudadano/mtrigasTFC0612memoria.pdf>

## ANEXOS

### Anexo I: Pila del Producto e Historias de Usuarios

**Tabla 1-I.** Pila del producto, SPRINT 1- Requerimientos funcionales

#### PILA DEL PRODUCTO

<b>SISTEMA:</b>			Sistema de Control Vehicular SISCOVE V1.0	<b>SCRUM TEAM RESPONSABLE:</b>			TS1, TS2, SCRUM MASTER
<b>VERSIÓN DEL DOCUMENTO:</b>			1	<b>FECHA DE REVISIÓN:</b>			07/04/2016
<b>DESCRIPCIÓN</b>			El presente documento contiene una lista ordenada y priorizada de las tareas que comprenden el proyecto de desarrollo del Sistema de Control Vehicular SISCOVE V1.0. Su objetivo es definir el alcance del proyecto tanto al equipo de trabajo SCRUM como al propietario del sistema. Deberá ser discutido y modificado hasta generar una versión final del mismo, el cual guíe la elaboración de documento técnico de Especificación de Requerimientos de Software (SRS).				
MODULO	SUBMODULO	ID	DESCRIPCIÓN	ESTIMA- CIÓN	PRIORI- DAD	DEPEN- DEN- CIAS	PRUEBAS DE ACEPTACIÓN
1. INFORMACIÓN GENERAL	MECÁNICOS	1.1	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de mecánicos</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: Código (automático), Apellidos, Nombres, Teléfono Convencional, Teléfono Celular y Estado (activo o inactivo)	16	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.  Activar/Desactivar un mecánico y verificar si el cambio de estado se refleja correctamente en el sistema (no se puede desactivar un mecánico que se encuentre reparando un vehículo).
		1.2	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de vehículos</i> (ingreso, modificación, cambio de estado, cambio de disponibilidad), basado en los siguientes datos: Número de vehículo, Dependencia, Número de Placa, Tipo de Vehículo (bus o camioneta), Año, Marca, Tipo Específico, Número de Motor, Matrícula, Número de Chasis, Color, Km de Garantía, Estado (activo, inactivo o mantenimiento), Disponibilidad (si o no)	16	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.  Modificar el estado de un vehículo y verificar si éste se refleja correctamente en el sistema.

	CHOFERES	1.3	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de choferes</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: CI, Apellidos, Nombres, Teléfono Convencional, Teléfono Celular y Estado (activo o inactivo)	16	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información. Activar/Desactivar un chofer y verificar si el cambio de estado se refleja correctamente en el sistema (no se puede desactivar un chofer que se encuentre asignado a un vehículo).
	RESPUESTO / LUBRICANTE	1.4	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de repuestos/lubricantes</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: Código (automático), Nombre, Descripción, Tipo (repuesto o lubricante), Unidad de Medida (litros, unidades, libras), Mínima Cantidad de descarga (1 unidad de medida, 1/4 de unidad de medida, etc), Stock bajo, Stock medio y Estado (activo o inactivo)	16	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información. Activar/Desactivar un repuesto/lubricante y verificar si el cambio de estado se refleja correctamente en el sistema (no se puede desactivar un repuesto/lubricante que tenga 'existencias' en bodega).
	USUARIOS_SIS TEMAS	1.5	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de otros usuarios del sistema</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: CI, Apellidos, Nombres, Usuario, Clave, Tipo (administrador o formatos) y Estado (activo o inactivo)	16	M	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información. Activar/Desactivar un chofer y verificar si el cambio de estado se refleja correctamente en el sistema.
	DEPARTAMENTOS	1.6	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de departamentos</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: Código (automático, Nombre)	8	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información. Activar/Desactivar un departamento y verificar si el cambio de estado se refleja correctamente en el sistema.
<b>2. PROCESOS DEL TALLER</b>	VEHÍCULOS_CHOFERES (MOVILIDAD)	2.1	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la asignación de choferes activos a vehículos activos</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: Código (automático), Vehículo, Chofer,	24	A	1.2, 1.3	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.

			Número de Autorización, Fecha y Hora de Salida, Fecha y Hora Aproximada de Retorno, Fecha y Hora de Retorno (real), Motivo (Gira, Gestión Administrativa, etc...), Observaciones (motivo de retraso, desactivación, etc...), Estado (activo, inactivo) y el Tipo de Asignación (permanente o no)				Activar/Desactivar un vehículo_chofer y verificar si el cambio de estado se refleja correctamente en el sistema.
	INVENTARIO	2.2	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>gestionar el inventario</b> (ingreso, modificación), basado en los siguientes datos: Código (automático), Repuesto/Lubricante, Fecha de Ingreso, Cantidad, Descripción, Número de Orden de Entrega.	24	A	1.4	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
	REPARACIÓN VEHICULAR	2.3	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>gestionar la reparación de los vehículos</b> (ingreso, modificación), basado en los siguientes datos: Número de Orden de Trabajo, Vehículo, Mecánico(s), Tipo de Reparación, Fecha y Hora de Ingreso al Taller, Fecha y Hora de Salida del Taller, Observaciones Generales, Repuestos y Lubrificantes Utilizados, Reparaciones Solicitadas y Realizadas con sus debidas Observaciones.	32	A	1.1, 1.2 o 2,1, 2.2	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
<b>3. REPORTES DEL SISTEMA</b>	VEHICULOS	3.1	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de vehículos</b> , con los siguientes filtros de datos: estado, disponibilidad, departamento, tipo de vehículo. <b>ACLARACIONES:</b> El reporte contendrá toda la información del <i>vehículo</i> y del <i>departamento</i> al cual pertenece.	8	M	1.2	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
	MOVILIDAD	3.2	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de movilidad - vehículos</b> , con los siguientes filtros de datos: fecha de salida (entre fechas), motivo, número de autorización, disponibilidad del vehículo. Formato de salida: pdf. <b>ACLARACIONES:</b> El reporte contendrá la información de todos los <i>vehículos activos</i> con sus <i>choferes asignados</i> , además de los datos específicos de dicha asignación.	8	A	2.1	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.

		3.3	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de movilidad - choferes</b>, con los siguientes filtros de datos: fecha de salida (entre fechas), motivo, número de autorización. Formato de salida: pdf.</p> <p><b>ACLARACIONES:</b> El reporte contendrá la información de todos los <b>choferes activos</b> con sus <b>vehículos asignados</b>, además de los datos específicos de dicha asignación.</p>	8	A	2.1	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
	INVENTARIO	3.4	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de inventario de lubricantes y repuestos</b>, con los siguientes filtros de datos: tipo de elemento (repuestos o lubricantes), número de orden de entrega, cantidad total en bodega (rangos de stock). Formato de salida: pdf.</p> <p><b>ACLARACIONES:</b> El reporte contendrá la información de todos los <b>repuestos/lubricantes activos</b> con sus <b>existencias en inventario</b>. Mediante una opción de pantalla, permitirá obtener el kardex del producto con datos específicos de ingresos y egresos de bodega, también en versión imprimible.</p>	16	M	2.2	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
	REPARACIÓN VEHICULAR	3.5	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de reparación vehicular - mecánica</b>, con los siguientes filtros de datos: mecánico, tipo de reparación, fecha de ingreso a la mecánica (entre fechas), fecha de salida (entre fechas), detección de accidente (si/no), trabajos efectuados, trabajos solicitados. Formato de salida: pdf.</p> <p><b>ACLARACIONES:</b> El reporte contendrá, por cada reparación vehicular, la información del <b>vehículo, mecánico(s) asignados, reparación (datos generales y trabajos mecánicos aplicados)</b>.</p>	16	A	2.3	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.

		3.6	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar e imprimir un reporte de reparación vehicular individual</b> (de cada vehículo/orden de trabajo), para efectos administrativos. Formato de salida: pdf.</p>	8	A	2.3	Comprobar que se genera e imprime correctamente.
			<p><b>ACLARACIONES:</b> El reporte contendrá la información del <i>vehículo, mecánico(s) asignados, reparación (datos generales y trabajos mecánicos aplicados)</i>.</p>				
		3.7	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de reparación vehicular - inventario</b>, con los siguientes filtros de datos: tipo (repuestos o lubricantes), fecha de descarga de inventario (entre fechas). Formato de salida: pdf.</p>	16	M	2.3	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
			<p><b>ACLARACIONES:</b> El reporte contendrá la información del <i>vehículo, repuesto o lubricante, y datos específicos de la descarga del producto del inventario (fecha, cantidad)</i>.</p>				
<b>4. FORMATOS</b>	FORMATOS	4.1	<p><b>REQUERIMIENTO:</b> Como cliente, necesito el <u>usuario "formatos" del sistema</u> pueda <b>generar formatos de documentos administrativos</b>, para su impresión: "Orden de Entrega de Lubricantes y Repuestos" y "Adquisición o Partes y Repuestos de Vehículos o Bienes Materiales".</p>	16	B	-	Seleccionar el formato y comprobar que se genera e imprime correctamente.
<b>5. DIAGNÓSTICO VEHICULAR</b>	DIAGNÓSTICO BÁSICO DE FALLAS MECÁNICAS	5.1	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>mecánico</u> pueda <b>realizar el diagnóstico básico computarizado de las fallas mecánicas de un vehículo</b>, mediante la contestación a un conjunto finito de preguntas.</p>	40	A	SISTEMA DE REGLAS PREVIAMENTE ESTABLECIDO	Introducir varias combinaciones de respuestas a las preguntas del módulo de diagnóstico, y comprobar que el sistema propone correctamente las soluciones en base al sistema de reglas planteado .

		<p><b>ACLARACIONES:</b> El sistema presentará por defecto el problema y la solución PRIMARIA o MÁS PROBABLE. En caso de que la solución planteada no corrija la falla del vehículo, se presentará un listado de problemas y soluciones SECUNDARIAS O MENOS PROBABLES.</p>					
		<b>TOTAL EN HORAS ESTIMADAS DE DESARROLLO</b>	304	<b>HORAS</b>	<b>NOTACIÓN:</b> A= ALTA, M = MEDIA, B= BAJA		

Fuente: Patricia Herrera y Vanessa Vargas

**Tabla 2-I.** Pila del producto, SPRINT 2- Requerimientos funcionales

**PILA DEL PRODUCTO**

<b>SISTEMA:</b>	Sistema de Control Vehicular SISCOVE V1.0	<b>SCRUM TEAM RESPONSABLE:</b>	TS1, TS2, SCRUM MASTER
<b>VERSIÓN DEL DOCUMENTO:</b>	2	<b>FECHA DE REVISIÓN:</b>	12/04/2016
<b>DESCRIPCIÓN</b>	El presente documento contiene una lista ordenada y priorizada de las tareas que comprenden el proyecto de desarrollo del Sistema de Control Vehicular SISCOVE V1.0. Su objetivo es definir el alcance del proyecto tanto al equipo de trabajo SCRUM como al propietario del sistema. Deberá ser discutido y modificado hasta generar una versión final del mismo, el cual guíe la elaboración de documento técnico de Especificación de Requerimientos de Software (SRS).		

MODULO	SUBMODULO	ID	DESCRIPCIÓN	ESTI-MACIÓN	PRIO-RIDAD	DEPEN-DENCIAS	PRUEBAS DE ACEPTACIÓN
--------	-----------	----	-------------	-------------	------------	---------------	-----------------------

1. INFORMACIÓN GENERAL	MECÁNICOS	1.1	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de mecánicos</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: CI, Apellido, Nombre, Teléfono Convencional, Teléfono Celular, Fecha de Nacimiento, Tipo de Sangre, Enfermedades y Estado (activo o inactivo)	16	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
							Activar/Desactivar un mecánico y verificar si el cambio de estado se refleja correctamente en el sistema (no se puede desactivar un mecánico que se encuentre reparando un vehículo).
	CHOFERES	1.2	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de choferes</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: CI, Apellido, Nombre, Teléfono Convencional, Teléfono Celular y Estado (activo o inactivo)	16	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
							Activar/Desactivar un chofer y verificar si el cambio de estado se refleja correctamente en el sistema (no se puede desactivar un chofer que se encuentre asignado a un vehículo).
	VEHÍCULOS	1.3	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de vehículos</i> (ingreso, modificación, cambio de estado, cambio de disponibilidad), basado en los siguientes datos: Número de vehículo, Dependencia, Tipo de Vehículo (bus o camioneta), Chofer, Año, Marca, Modelo, Número de Motor, Matrícula, Número de Chasis, Placa, Color, Km de Garantía, Estado (activo, inactivo o mantenimiento), Disponibilidad (sí o no)	16	A	1.2	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
							Modificar el estado de un vehículo y verificar si éste se refleja correctamente en el sistema.
	RESPUESTO / LUBRICANTE / ADITIVO	1.4	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de respuestas/lubricantes</i> (ingreso, modificación, activación/desactivación, kardex), basado en los siguientes datos: Código (automático),	16	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.

			Nombre, Descripción, Tipo (repuesto o lubricante), Unidad de Medida (litros, unidades, libras), Mínima Cantidad de descarga (1 unidad de medida, 1/4 de unidad de medida, etc), Stock bajo, Stock medio y Estado (activo o inactivo)				Activar/Desactivar un repuesto/lubricante y verificar si el cambio de estado se refleja correctamente en el sistema (no se puede desactivar un repuesto/lubricante que tenga 'existencias' en bodega).
	USUARIOS_SISTEMAS	1.5	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la información de otros usuarios del sistema</i> (ingreso, modificación, activación/desactivación, seteo de clave), basado en los siguientes datos: CI, Apellidos, Nombres, Usuario, Clave, Tipo (administrador o formatos) y Estado (activo o inactivo)	16	M	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.  Activar/Desactivar un chofer y verificar si el cambio de estado se refleja correctamente en el sistema.
	DEPENDENCIAS	1.6	<b>REQUERIMIENTO:</b> Como cliente, necesito que el administrador del sistema pueda <i>gestionar la información de dependencias</i> (ingreso, modificación, activación/desactivación), basado en los siguientes datos: Código (automático, Nombre)	8	A	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.  Activar/Desactivar un departamento y verificar si el cambio de estado se refleja correctamente en el sistema.
<b>2. PROCESOS DEL TALLER</b>	BODEGA (INVENTARIO)	2.1	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la bodega</i> (ingreso, modificación), basado en los siguientes datos: Código (automático), Tipo (Repuesto/Lubricante/Aditivo), Fecha de Ingreso, Cantidad, Descripción, Número de Orden de Entrega.	24	A	1.4	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
	REPARACIÓN VEHICULAR	2.2	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar la reparación de los vehículos u ordenes de trabajo</i> (ingreso, modificación, visualización de datos relacionados y dar de alta al vehículo), basado en los siguientes datos: Número de Orden de Trabajo (automático), Vehículo y tipo de reparación (correctiva o preventiva)	32	A	1.1, 1.3, 2.1	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.

		2.3	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar los egresos de bodega por reparaciones</i> (ingreso, modificación y eliminación), basado en los siguientes datos: Número de Orden de Trabajo, Repuesto/Lubricante/Aditivo, Cantidad y Fecha.	32	A	2.1, 2,2	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
							Eliminar un egreso de bodega y verificar si el cambio de estado se refleja correctamente en el sistema.
		2.4	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar los mecánicos asignados a una reparación</i> (ingreso, modificación y eliminación), basado en los siguientes datos: Orden de Trabajo, Mecánico y Fecha de asignación.	32	A	2.2	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
							Eliminar un mecánico asignado a una orden de trabajo y verificar si el cambio de estado se refleja correctamente en el sistema.
		2.5	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>gestionar los trabajos mecánicos de una reparación vehicular</i> (ingreso, modificación y eliminación), basado en los siguientes datos: Orden de Trabajo, Trabajo mecánico a realizar, Efectuado (estado: si o no).	32	A	2.2	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se ingresa/modifica la información.
							Eliminar un trabajo mecánico y verificar si el cambio de estado se refleja correctamente en el sistema.
<b>3. REPORTES DEL SISTEMA</b>	VEHICULOS	3.1	<b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <i>generar reportes de vehículos</i> , con los siguientes filtros de datos: estado, disponibilidad, dependencia, tipo de vehículo.	8	M	1.3	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
			<b>ACLARACIONES:</b> El reporte contendrá toda la información visualizada del <b>vehículo</b> en la pantalla.				

	BODEGA (INVENTARIO)	3.2	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de inventario de lubricantes, repuestos y aditivos</b>, con los siguientes filtros de datos: código, tipo de elemento (repuesto, lubricante o aditivo), cantidad total en bodega . Formato de salida: pdf.</p>	16	M	2.1, 2.3	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
			<p><b>ACLARACIONES:</b> El reporte contendrá la información visualizada de los <b>repuestos/lubricantes/aditivos</b> en la pantalla. Mediante una opción de pantalla, permitirá obtener el kardex del producto con datos específicos de ingresos y egresos de bodega, también en versión imprimible.</p>				
	REPARACIÓN VEHICULAR	3.3	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de reparación vehicular - mecánica</b>, con los siguientes filtros de datos: fecha de ingreso a la mecánica (entre fechas), fecha de salida (entre fechas), mecánico, vehículo, tipo de reparación, detección de accidente (si/no), trabajos efectuados, trabajos solicitados. Formato de salida: pdf.</p>	16	A	2.2	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
			<p><b>ACLARACIONES:</b> El reporte contendrá, por cada reparación vehicular, la información de: <b>número de orden de trabajo, vehículo (placa y marca), tipo de reparación, fechas de ingreso y salida, detección de accidente.</b></p>				
		3.4	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar e imprimir un reporte de reparación vehicular individual</b> (de cada vehículo/orden de trabajo), para efectos administrativos. Formato de salida: pdf.</p>	8	A	2.2	Comprobar que se genera e imprime correctamente.
			<p><b>ACLARACIONES:</b> El reporte contendrá la siguiente información: <b>vehículo, mecánico(s) asignados, reparación (datos generales y trabajos mecánicos aplicados), chofer responsable.</b></p>				

		3.5	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>administrador del sistema</u> pueda <b>generar reportes de reparación vehicular - inventario</b>, con los siguientes filtros de datos: tipo (repuesto, lubricante o aditivo), repuesto/lubricante/aditivo, fecha de descarga de inventario (entre fechas). Formato de salida: pdf.</p> <p><b>ACLARACIONES:</b> El reporte contendrá la siguiente información: número de orden de trabajo, nombre del repuesto/lubricante/aditivo, tipo de repuesto, cantidad, unidad, fecha de salida.</p>	16	M	2.2	Introducir datos incorrectos en los filtros de información y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
4. FORMATOS	FORMATOS GENERALES	4.1	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que los usuarios <u>formatos del sistema y administrador del sistema</u> puedan <b>generar/descargar formatos de documentos administrativos</b>, para su impresión: "Orden de Entrega de Lubricantes y Repuestos", "Adquisición o Partes y Repuestos de Vehículos o Bienes Materiales", en formatos .doc y .pdf.</p>	16	B	-	Seleccionar el formato y comprobar que se genera e imprime correctamente.
	FORMATOS CON DATOS PREIMPRESOS	4.2	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que los <u>usuarios "formatos" y "administrador"</u> del <u>sistema</u> puedan <b>generar un "Informe Técnico de Vehículos"</b>, <b>"Informe de Reparación"</b> y <b>"Orden de Entrega de Repuestos/Lubricantes/Aditivos"</b> con información preimpresa, en base a un listado de vehículos y ordenes de trabajo registrados en el sistema y varios filtros de información.</p>	8	B	-	Introducir datos incorrectos y comprobar que el sistema notifica el error. Introducir datos correctos y comprobar que se genera correctamente el reporte.
5. DIAGNÓSTICO VEHICULAR	DIAGNÓSTICO BÁSICO DE FALLAS MECÁNICAS	5.1	<p><b>REQUERIMIENTO:</b> Como cliente, necesito que el <u>mecánico</u> pueda <b>realizar el diagnóstico básico de las fallas mecánicas de un vehículo relacionadas a los sistemas de freno, embrague o motor</b>, mediante la contestación a un conjunto finito de preguntas pre-establecidas.</p>	40	A	SISTEMA DE REGLAS ESTABLECIDO CON UN PROFESIO	Introducir varias combinaciones de respuestas a las preguntas del módulo de diagnóstico, y comprobar que el sistema propone correctamente las soluciones en base al sistema de reglas planteado.

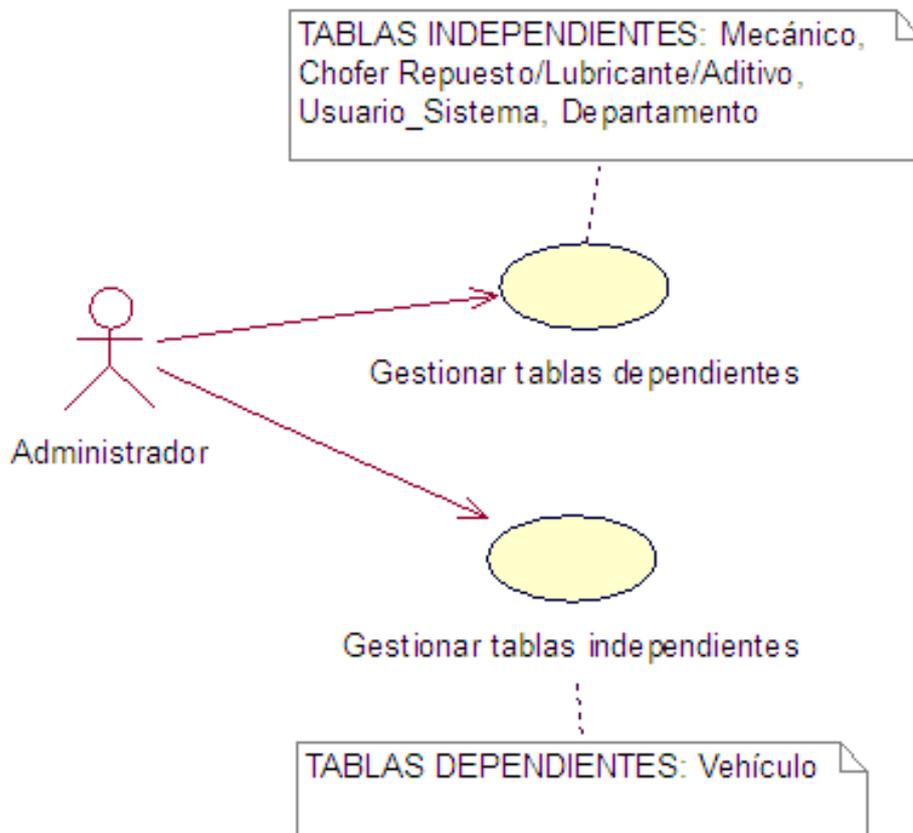
		<b>ACLARACIONES:</b> El sistema presentará por defecto el problema y la solución PRIMARIA o MÁS PROBABLE. En caso de que la solución planteada no corrija la falla del vehículo, se presentará un listado de problemas y soluciones SECUNDARIAS O MENOS PROBABLES. No se contemplan opciones de retroalimentación para el sistema de reglas.			NAL MECÁNICO
		<b>TOTAL EN HORAS ESTIMADAS DE DESARROLLO</b>	368	<b>HORAS</b>	<b>NOTACIÓN:</b> A= ALTA, M = MEDIA, B= BAJA

**Fuente:** Patricia Herrera y Vanessa Vargas

## Anexo II: Diagramas de Modelado UML

### DIAGRAMA DE CASOS DE USO

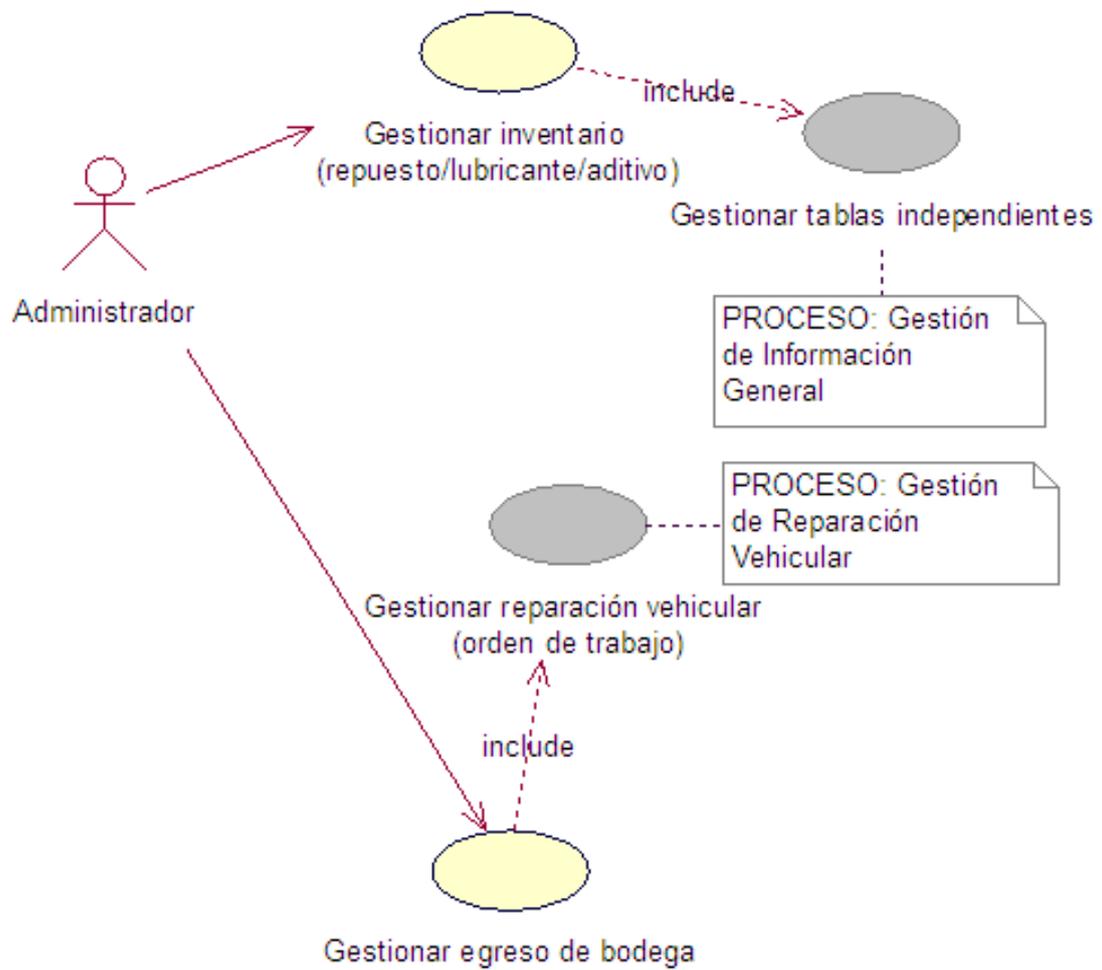
#### PROCESO P001: *GESTIÓN DE INFORMACIÓN GENERAL*



**Figura 1-II:** Diagrama de Casos de Uso – Proceso: Gestión de información general

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

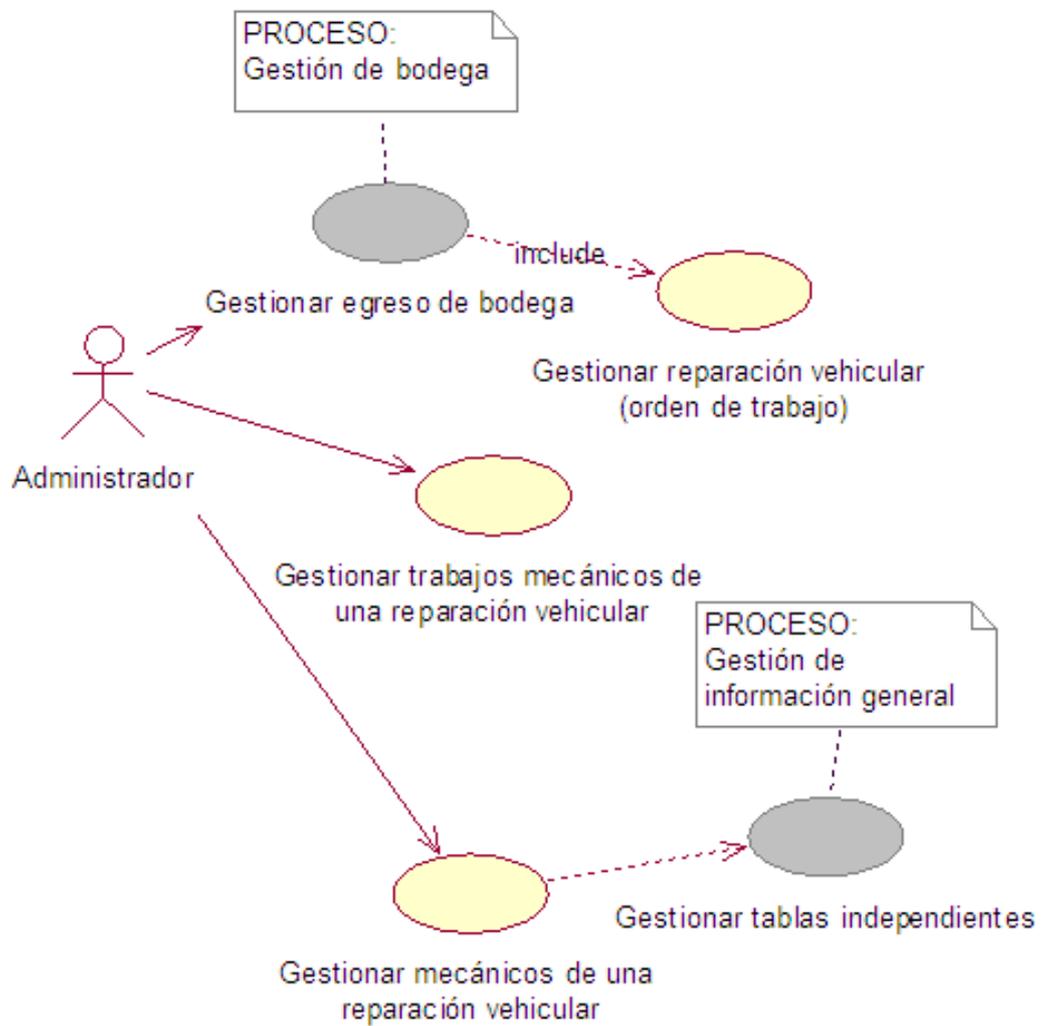
**PROCESO P002: GESTIÓN DE BODEGA**



**Figura 2-II:** Diagrama de Casos de Uso – Proceso: Gestión de bodega

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

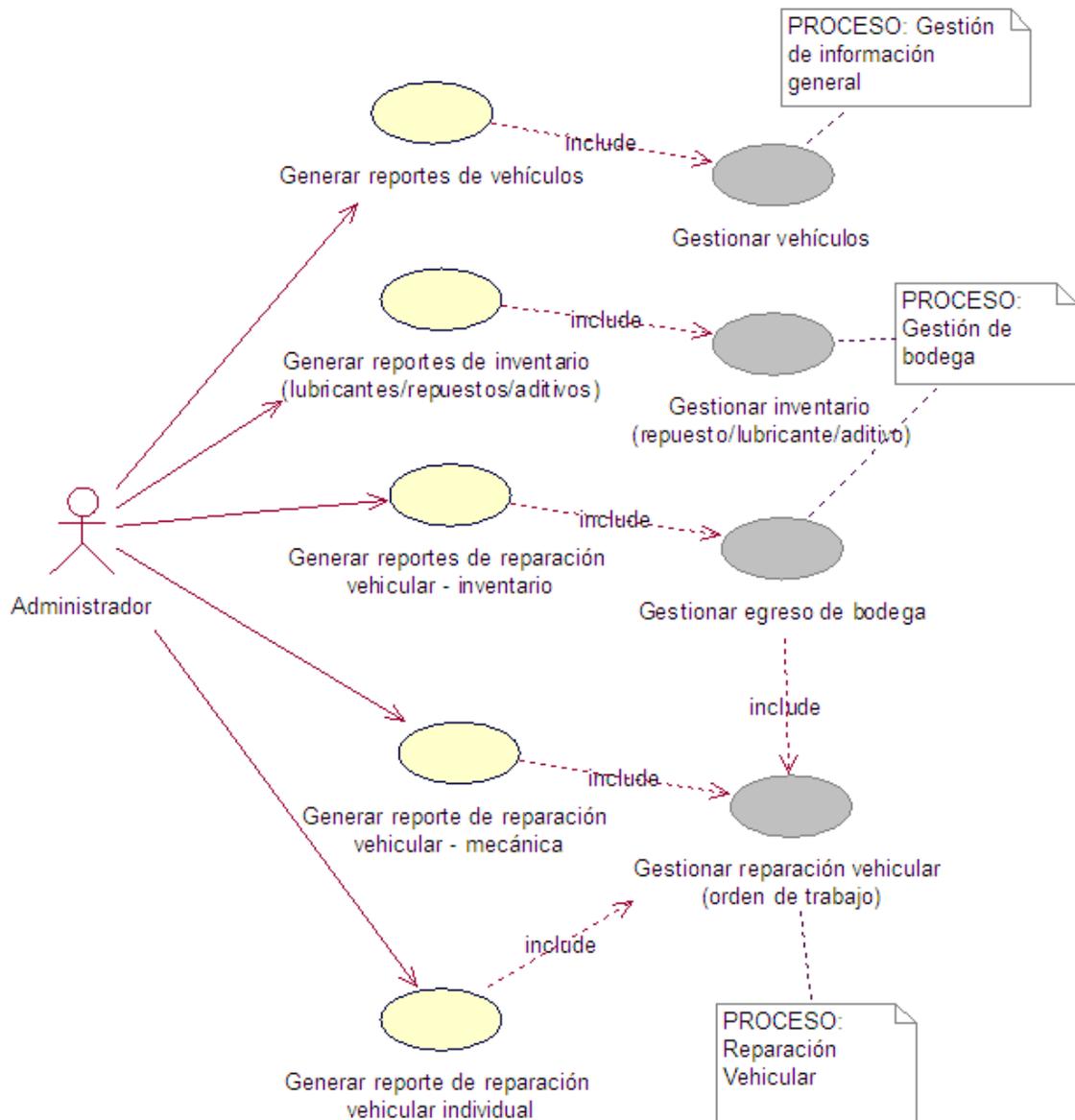
**PROCESO P003: REPARACIÓN VEHICULAR**



**Figura 3-II:** Diagrama de Casos de Uso – Proceso: Reparación Vehicular

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

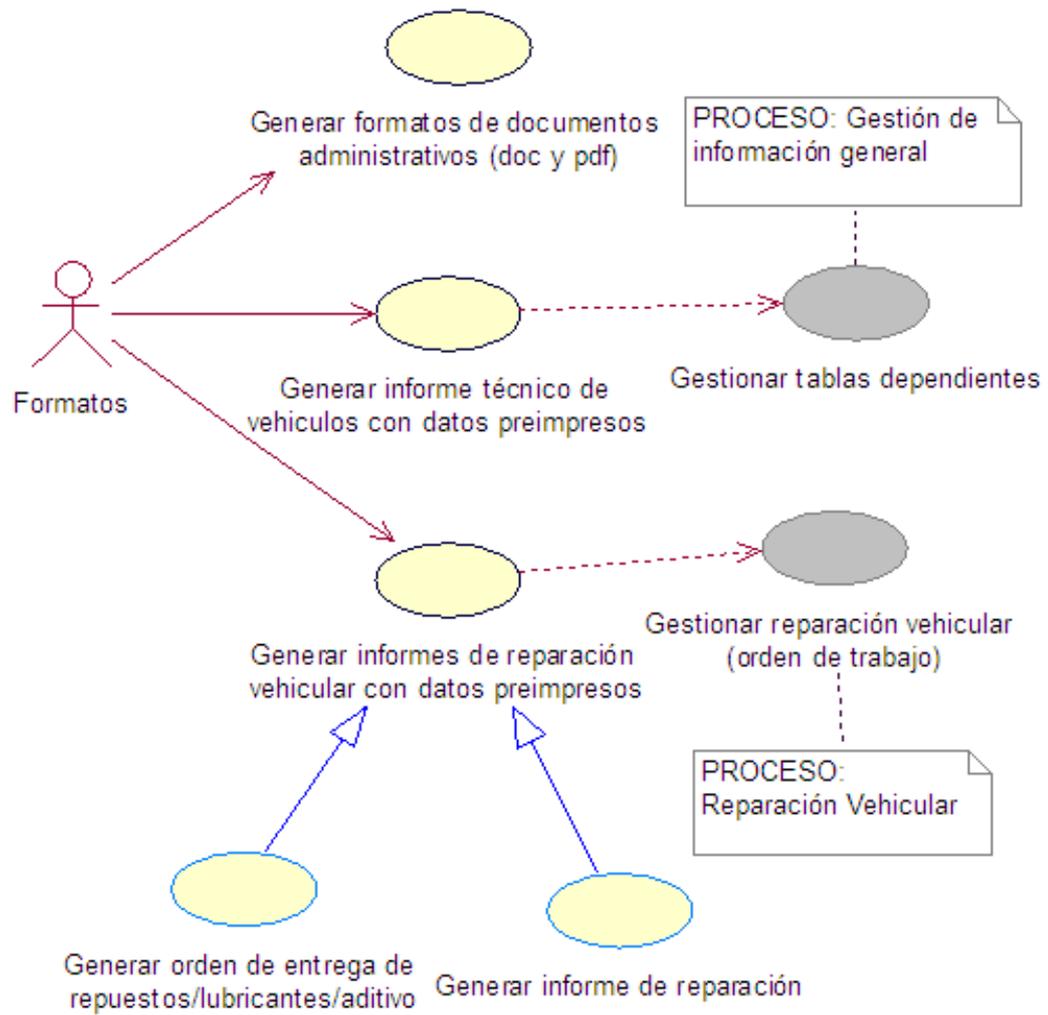
**PROCESO P004: GESTIÓN DE REPORTES DEL SISTEMA**



**Figura 4-II:** Diagrama de Casos de Uso – Proceso: Gestión de reportes del sistema

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

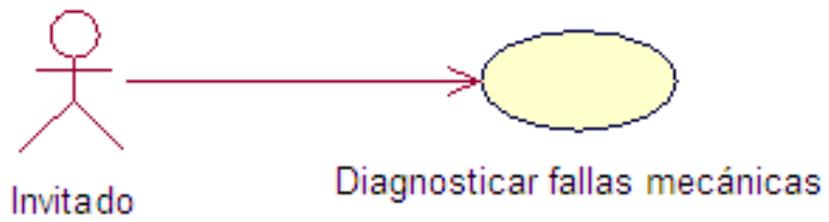
**PROCESO P005: GESTIÓN DE FORMATOS**



**Figura 5-II:** Diagrama de Casos de Uso – Proceso: Gestión de formatos

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

**PROCESO P006: DIAGNÓSTICO VEHICULAR**



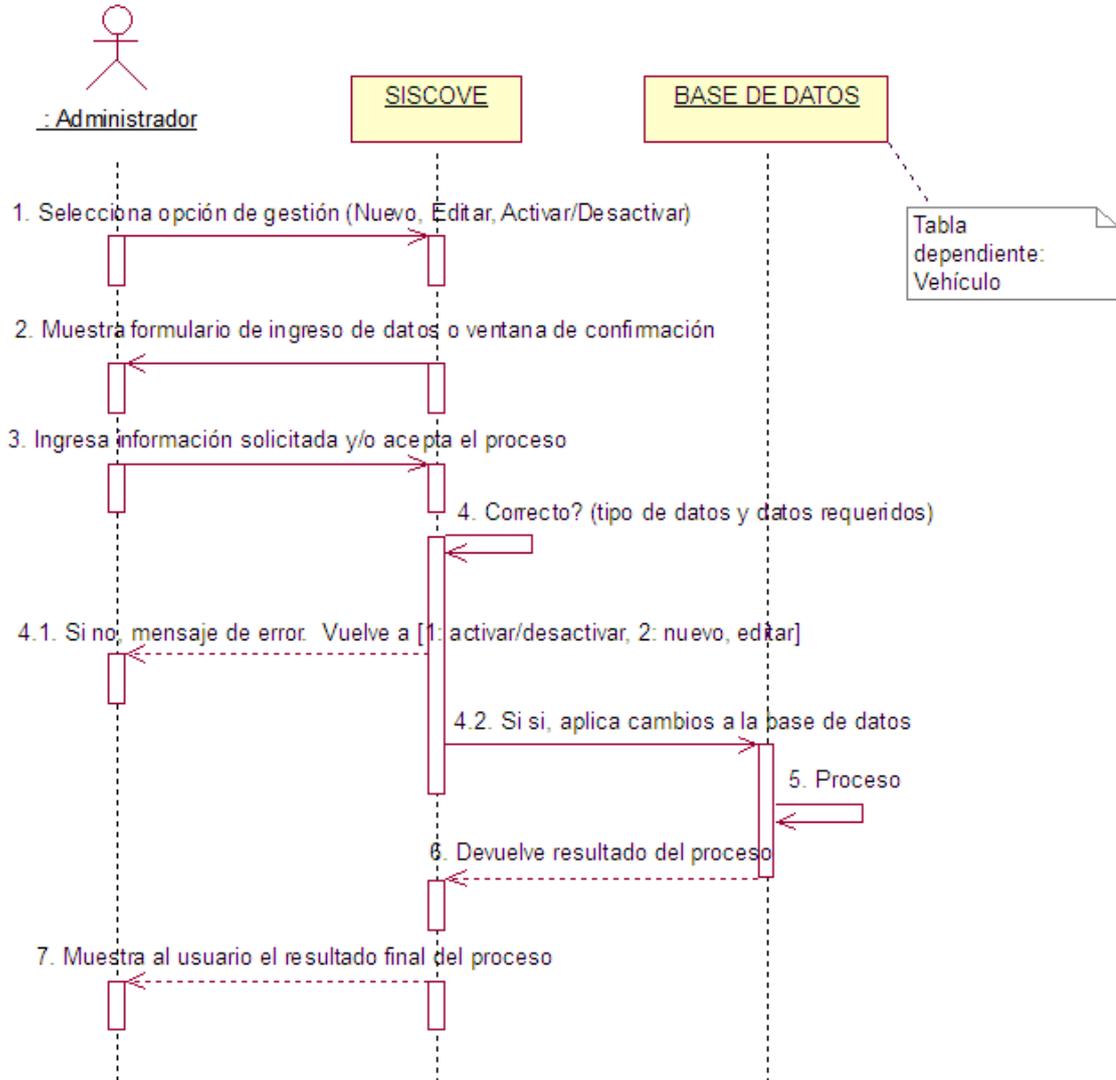
**Figura 6-II:** Diagrama de Casos de Uso – Proceso: Diagnóstico Vehicular

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

## DIAGRAMAS DE SECUENCIA

### **PROCESO P001: GESTIÓN DE INFORMACIÓN GENERAL**

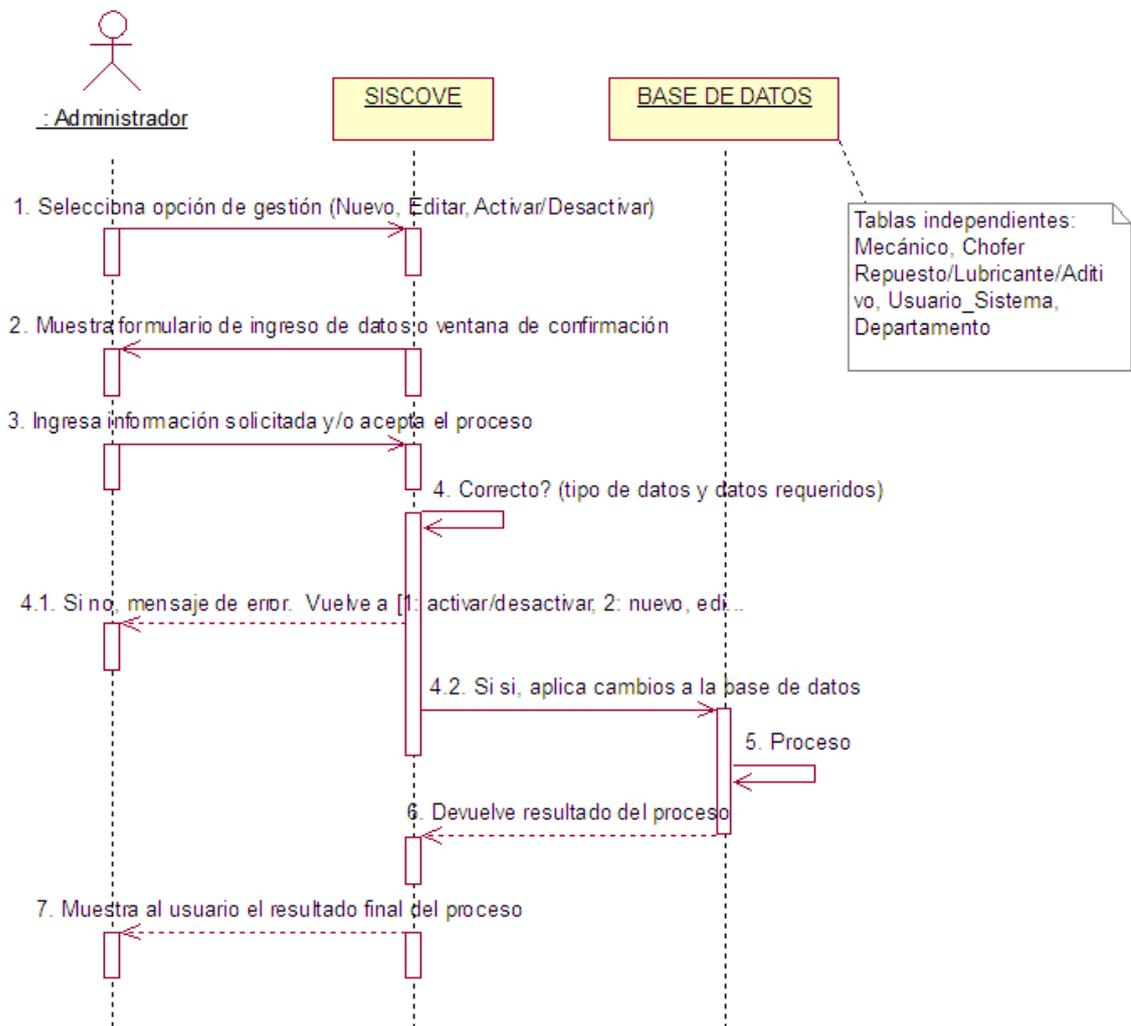
#### **CU001 – GESTIONAR TABLAS DEPENDIENTES**



**Figura 7-II:** Diagrama de Secuencias – Caso de Uso: Gestionar tablas dependientes

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

## CU002 – GESTIONAR TABLAS INDEPENDIENTES

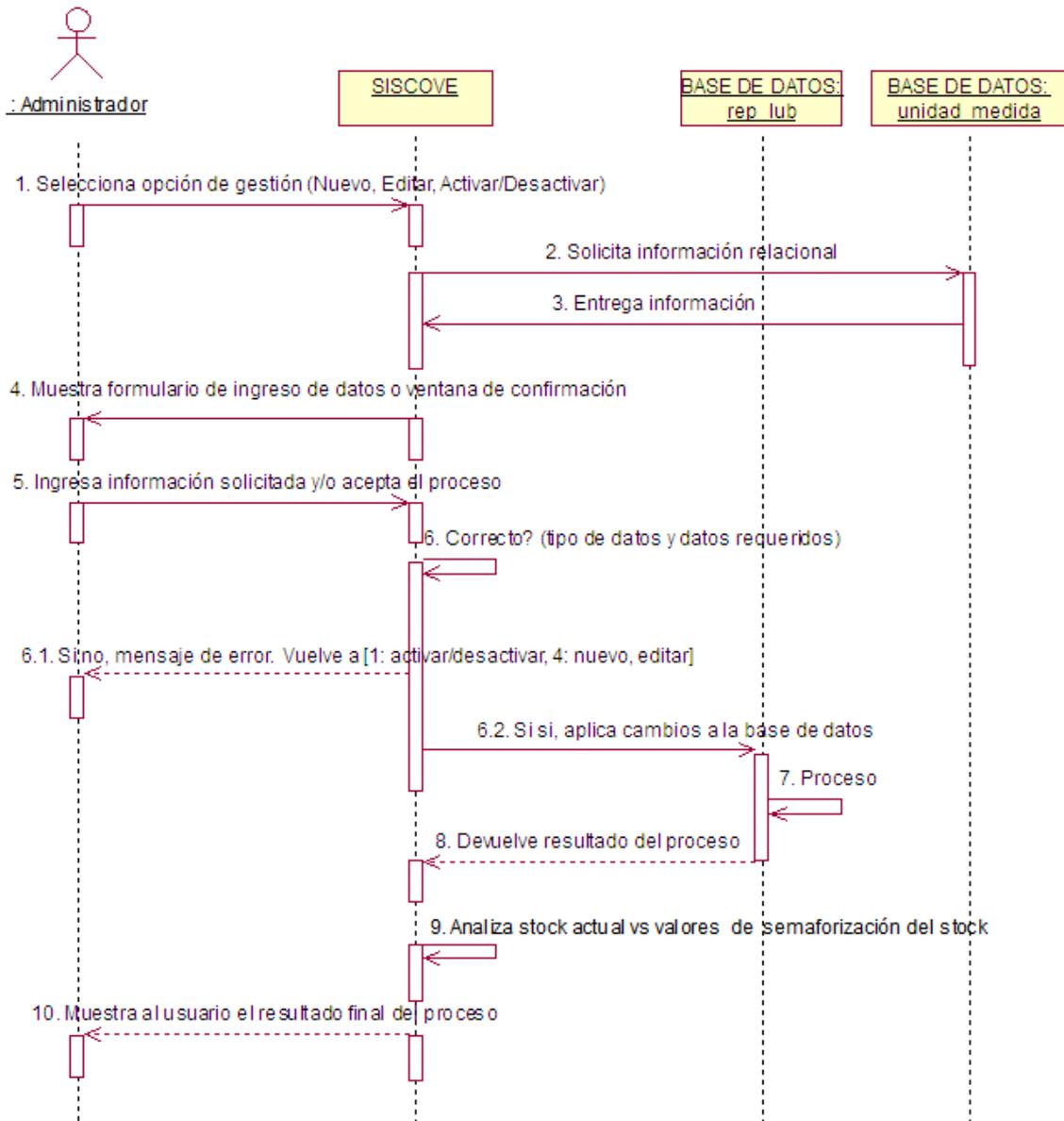


**Figura 8-II:** Diagrama de Secuencias – Caso de Uso: Gestionar tablas independientes

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

## PROCESO P002: GESTIÓN DE BODEGA

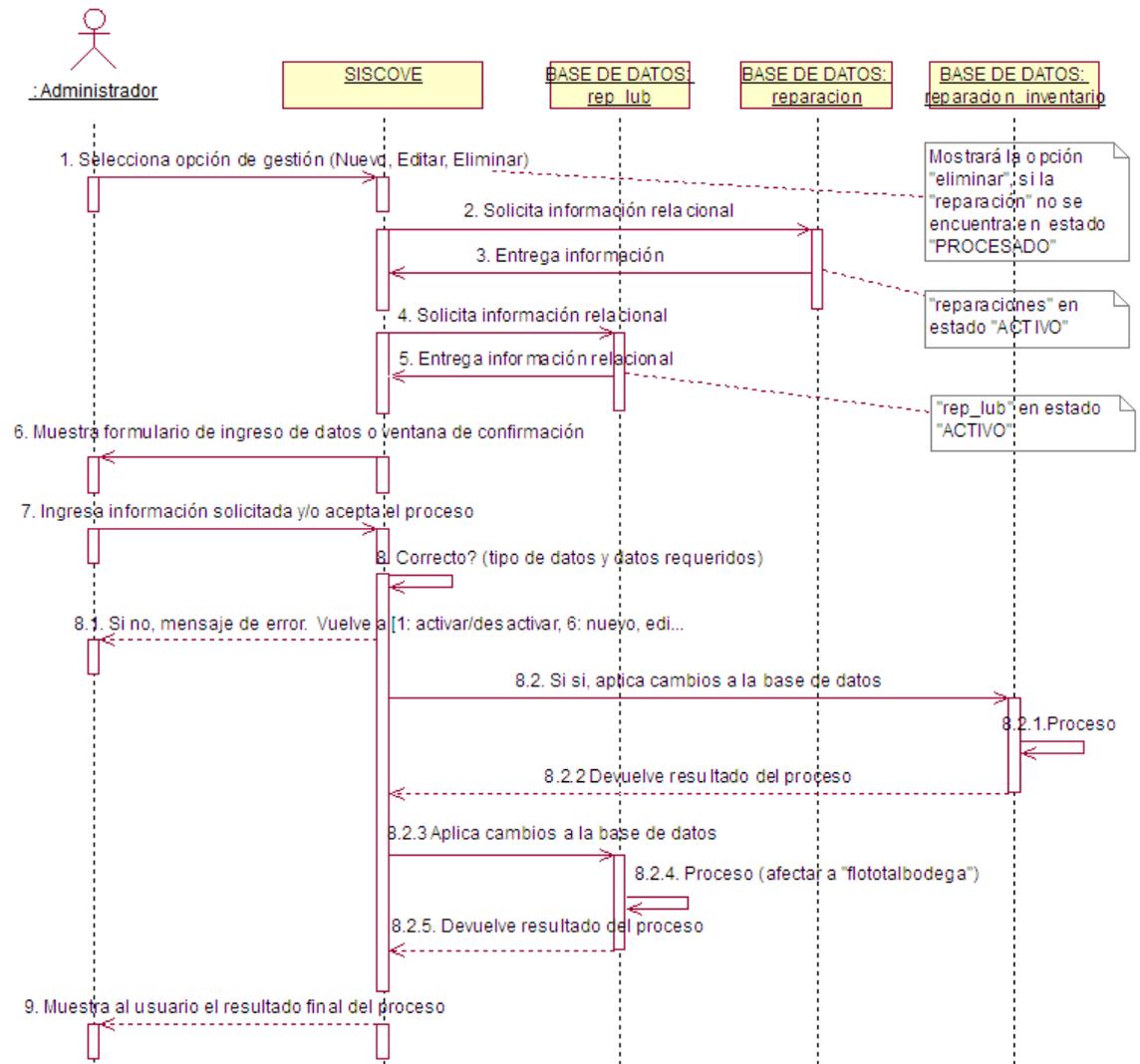
### CU003 – GESTIONAR INVENTARIO



**Figura 9-II:** Diagrama de Secuencias – Caso de Uso: Gestionar inventario

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

CU004 – GESTIONAR EGRESO DE BODEGA

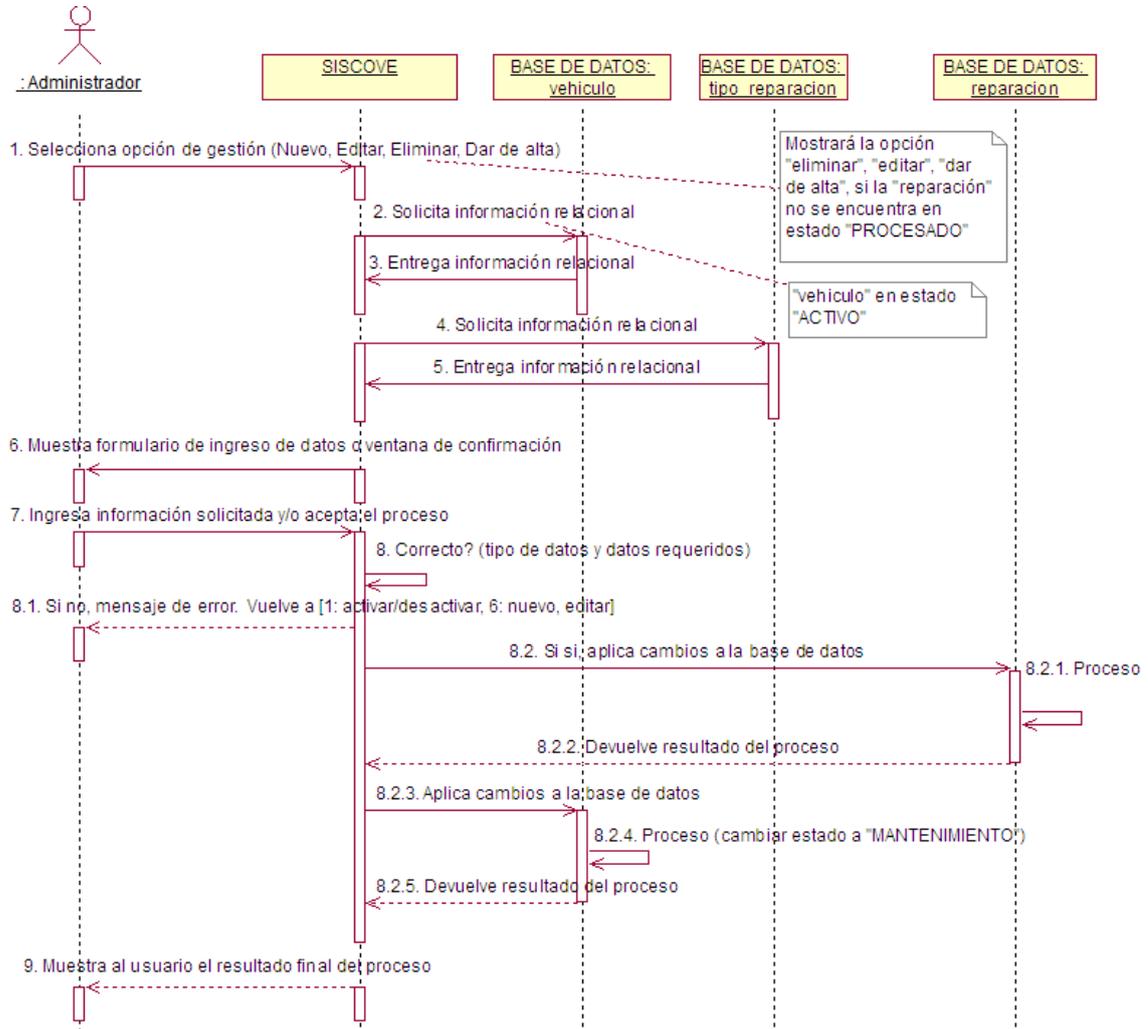


**Figura 10-II:** Diagrama de Secuencias – Caso de Uso: Gestionar egreso de bodega

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

## PROCESO P003: REPARACIÓN VEHICULAR

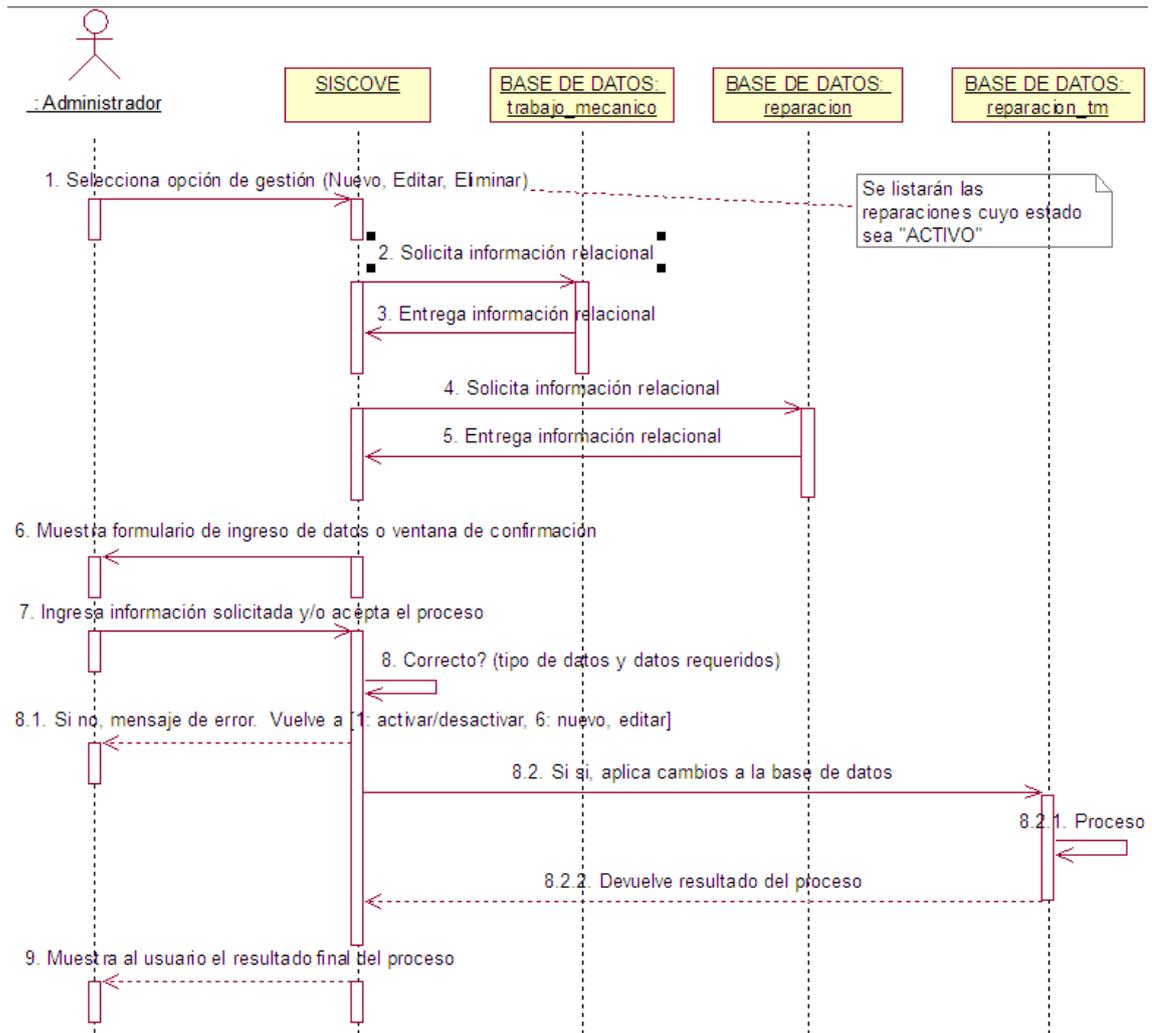
### CU005 – GESTIONAR REPARACIÓN VEHICULAR



**Figura 11-II:** Diagrama de Secuencias – Caso de Uso: Gestionar reparación vehicular

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

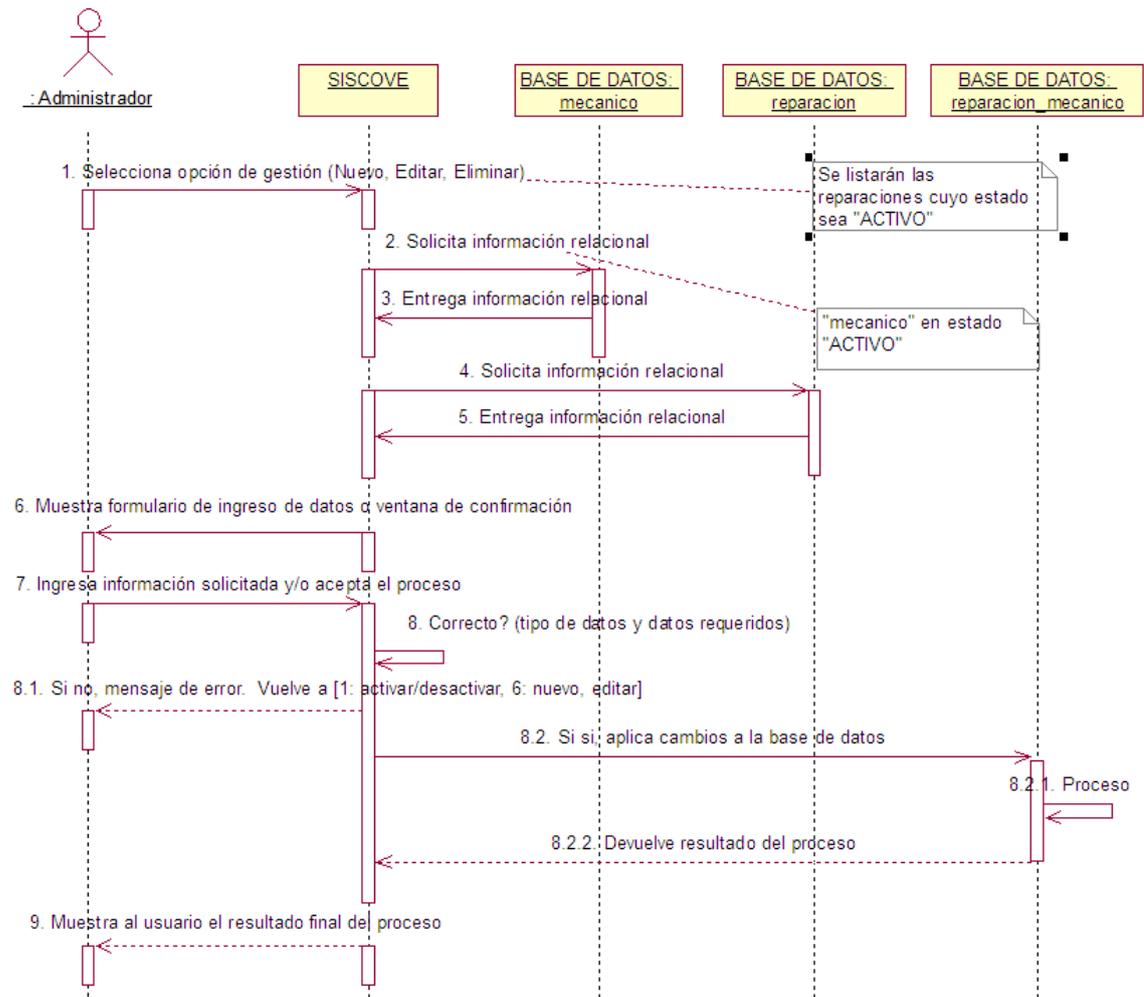
CU006 – GESTIONAR TRABAJOS MECÁNICOS DE UNA REPARACIÓN VEHICULAR



**Figura 12-II:** Diagrama de Secuencias – Caso de Uso: Gestionar trabajos mecánicos de una reparación vehicular

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

## CU007 – GESTIONAR MECÁNICOS DE UNA REPARACIÓN VEHICULAR

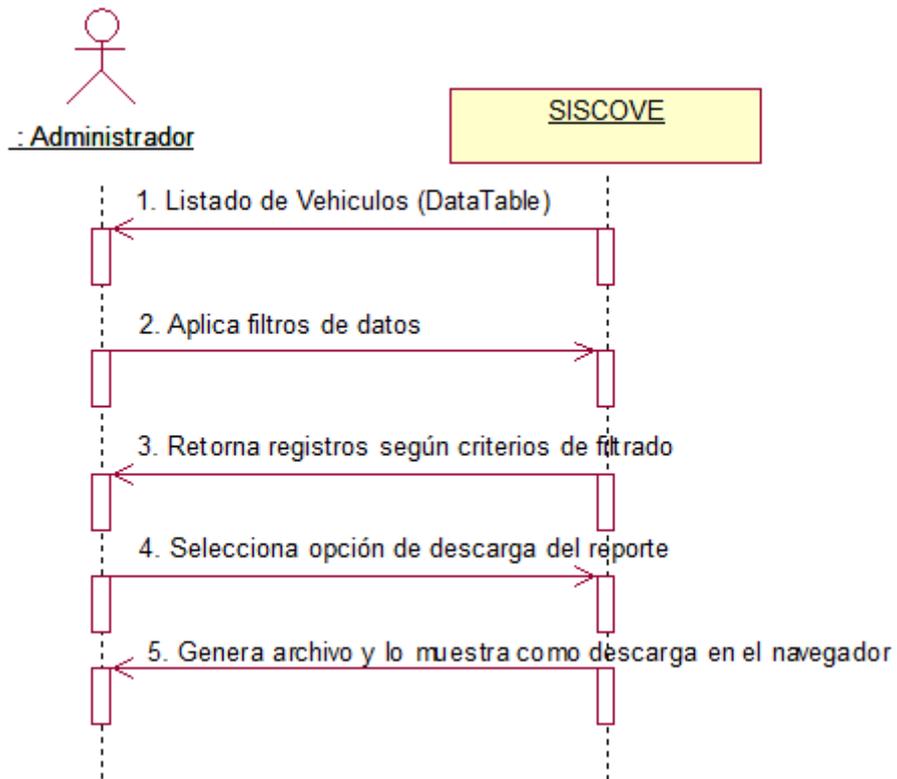


**Figura 13-II:** Diagrama de Secuencias – Caso de Uso: Gestionar mecánicos de una reparación vehicular

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

**PROCESO P004: GESTIÓN DE REPORTE DEL SISTEMA**

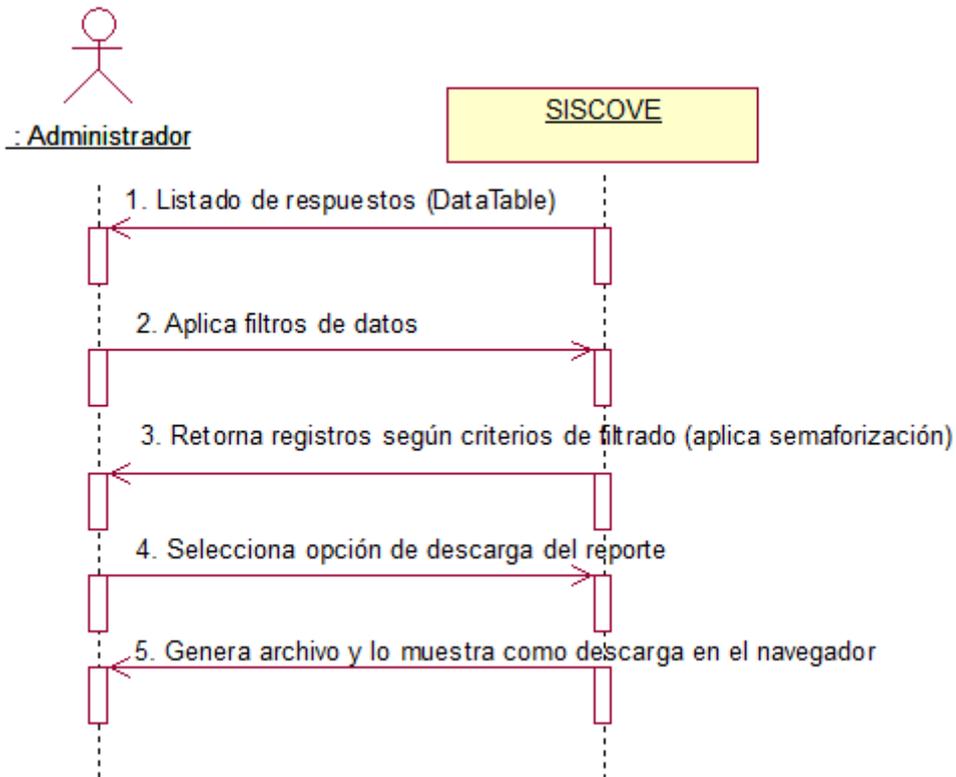
**CU008 – GENERAR REPORTE DE VEHÍCULOS**



**Figura 14-II:** Diagrama de Secuencias – Caso de Uso: Generar reportes de vehículos

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

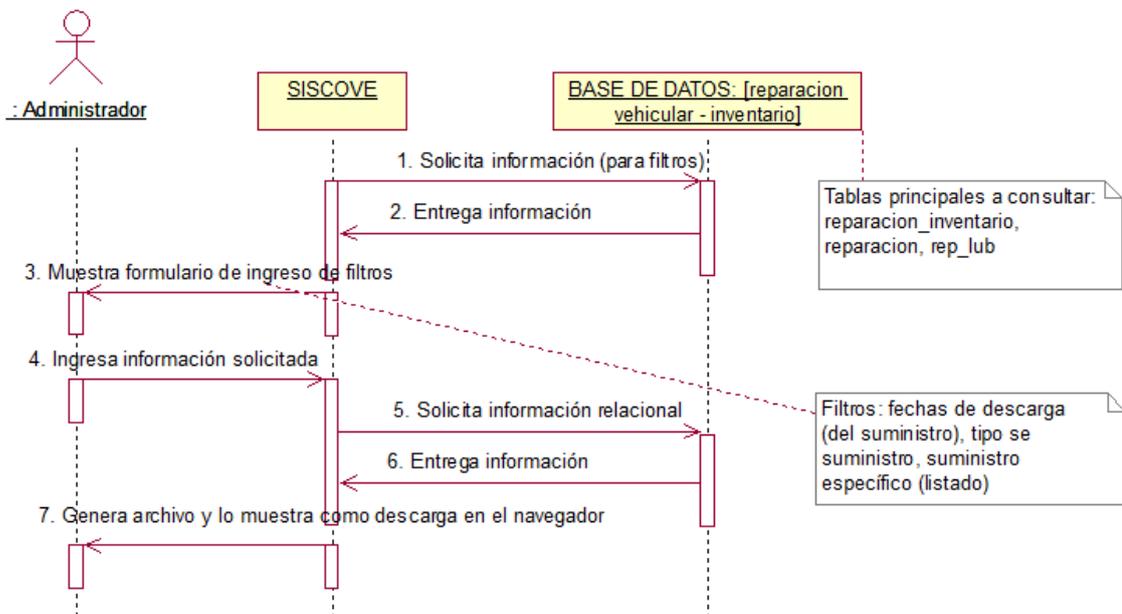
CU009 – GENERAR REPORTES DE INVENTARIOS



**Figura 15-II:** Diagrama de Secuencias – Caso de Uso: Generar reportes de inventarios

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

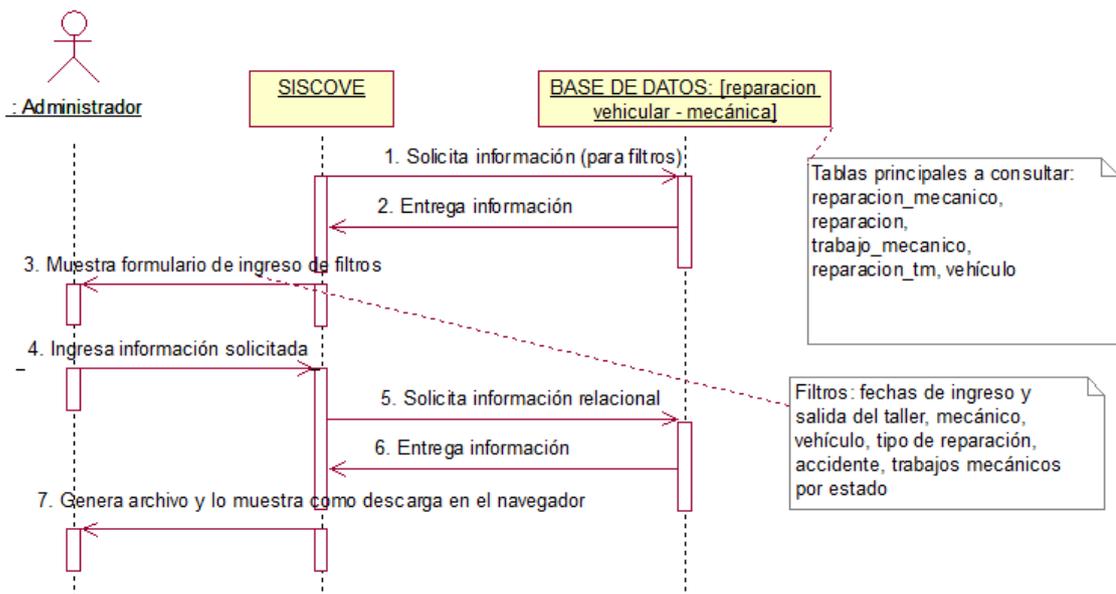
CU010 – GENERAR REPORTES DE REPARACIÓN VEHICULAR - INVENTARIO



**Figura 16-II:** Diagrama de Secuencias – Caso de Uso: Generar reportes de reparación vehicular - inventario

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

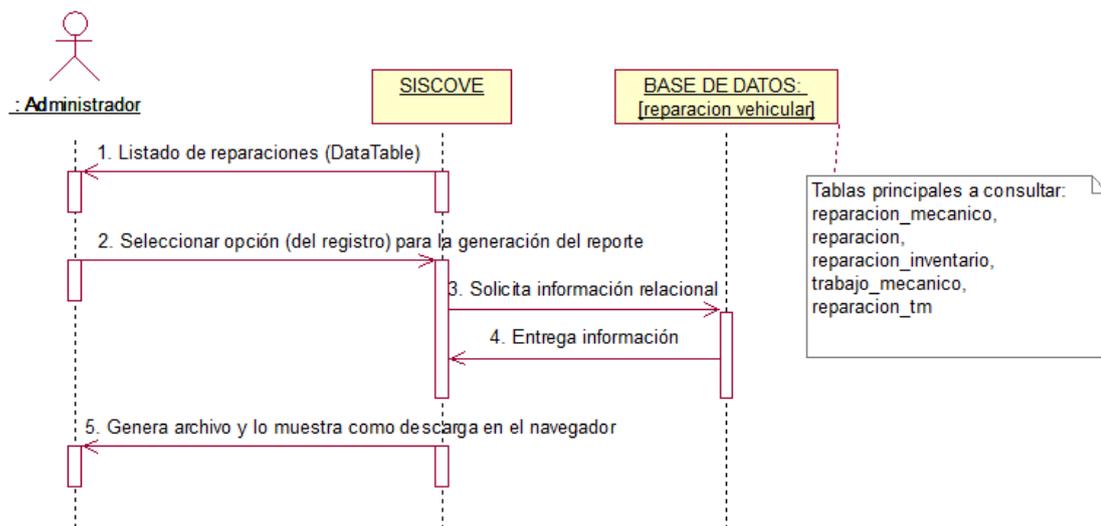
## CU011 – GENERAR REPORTES DE REPARACIÓN VEHICULAR - MECÁNICA



**Figura 17-II:** Diagrama de Secuencias – Caso de Uso: Generar reportes de reparación vehicular - mecánica

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

## CU012 – GENERAR REPORTE DE REPARACIÓN VEHICULAR INDIVIDUAL

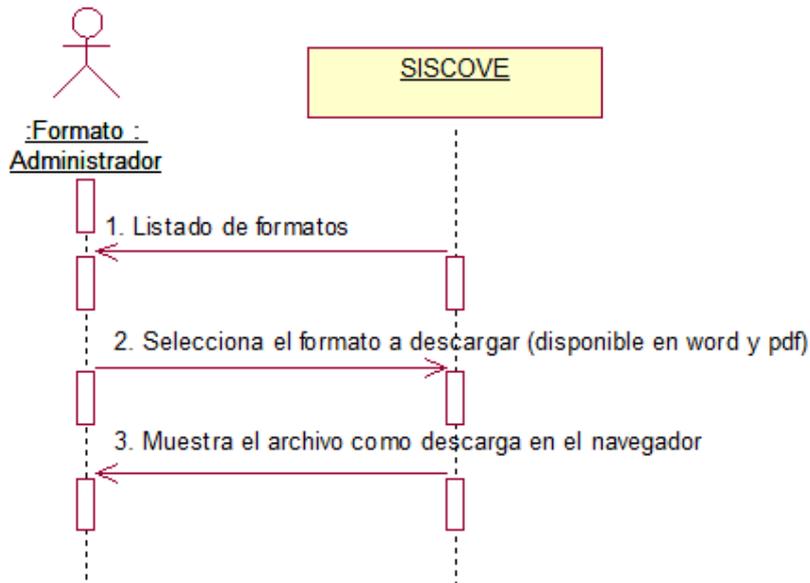


**Figura 18-II:** Diagrama de Secuencias – Caso de Uso: Generar reporte de reparación vehicular individual

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

**PROCESO P005: GESTIÓN DE FORMATOS**

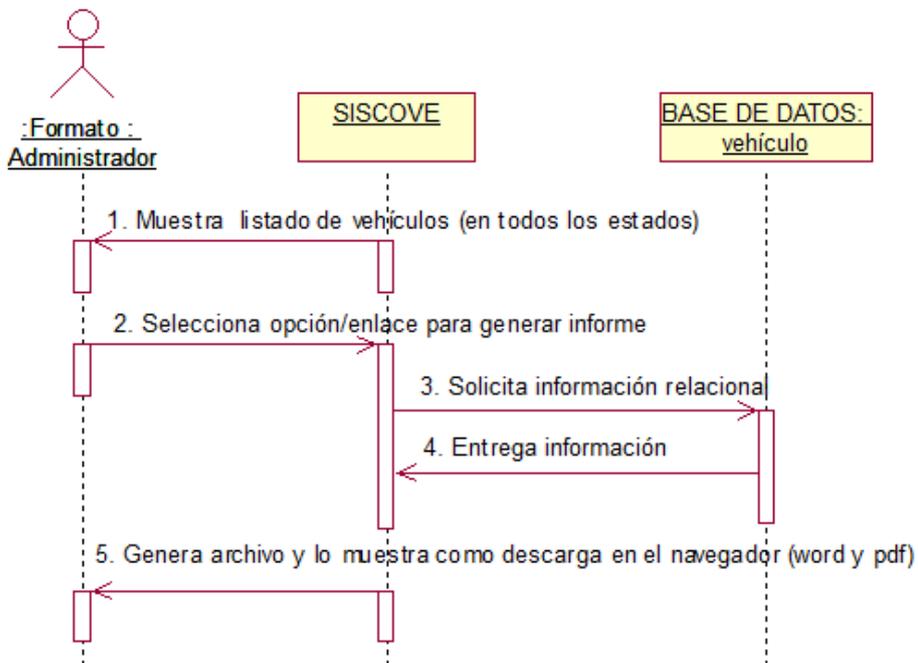
**CU013 – GENERAR FORMATOS DE DOCUMENTOS ADMINISTRATIVOS**



**Figura 19-II:** Diagrama de Secuencias – Caso de Uso: Generar formatos de documentos administrativos

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

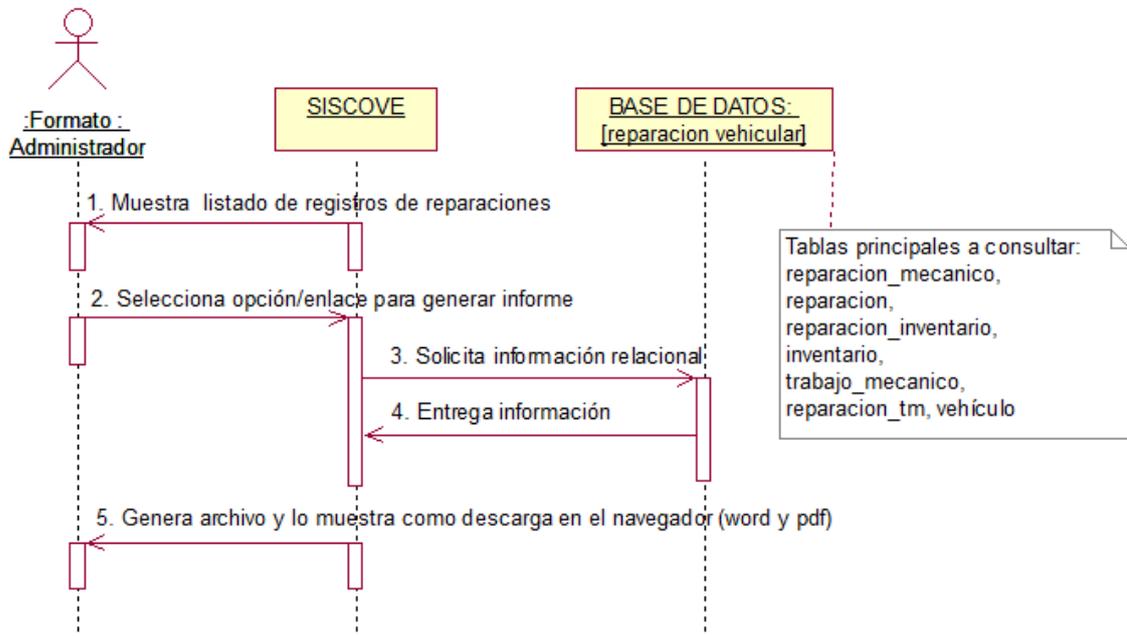
**CU014 – GENERAR INFORME TÉCNICO DE VEHÍCULOS CON DATOS PREIMPRESOS**



**Figura 20-II:** Diagrama de Secuencias – Caso de Uso: Generar informe técnico de vehículos con datos preimpresos

Fuente: Patricia Herrera y Vanessa Vargas (Rational Rose)

CU015 – GENERAR INFORMES DE REPARACIÓN VEHICULAR CON DATOS PREIMPRESOS

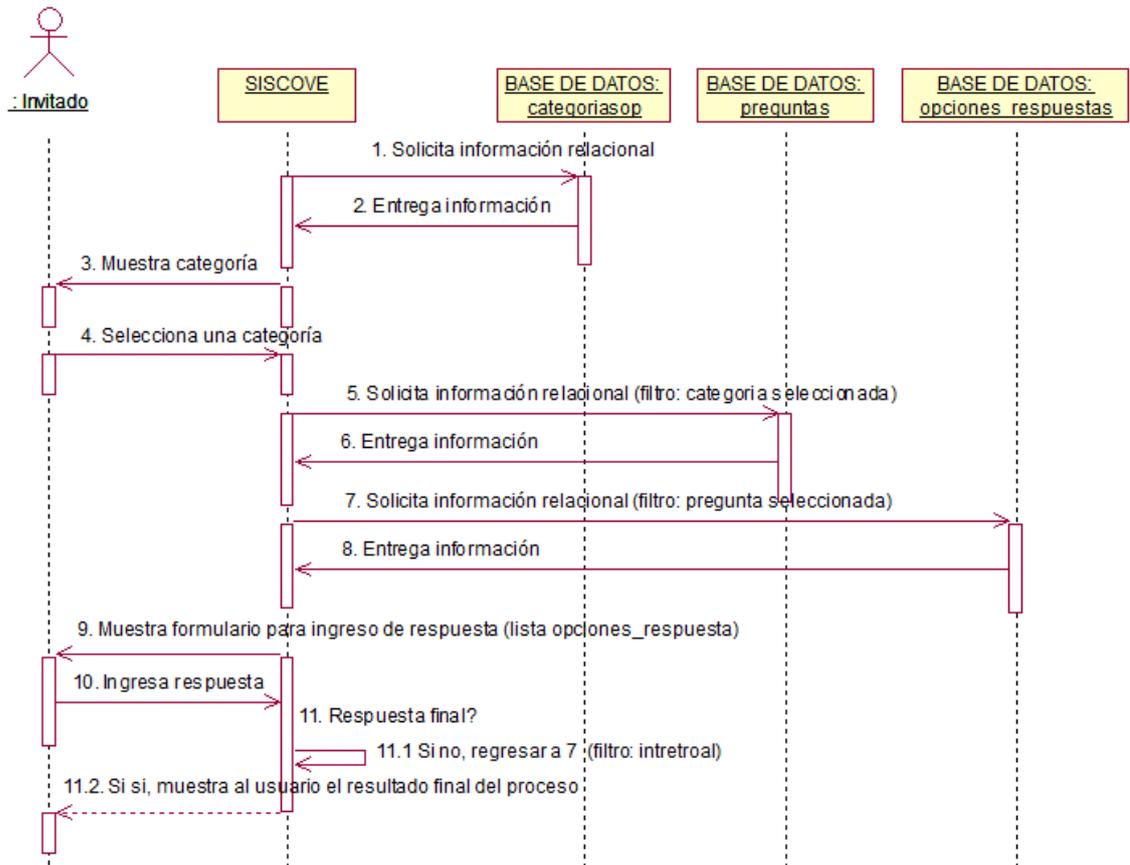


**Figura 21-II:** Diagrama de Secuencias – Caso de Uso: Generar informes de reparación vehicular con datos preimpresos

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

## PROCESO P006: DIAGNÓSTICO VEHICULAR

### CU016 – DIAGNOSTICAR FALLAS MECÁNICAS



**Figura 22-II:** Diagrama de Secuencias – Caso de Uso: Diagnosticar fallas mecánicas.

**Fuente:** Patricia Herrera y Vanessa Vargas (Rational Rose)

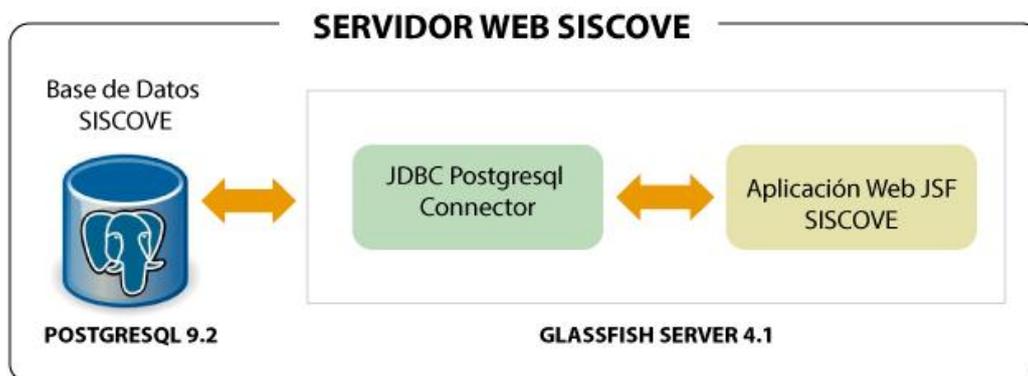
### **Anexo III: Arquitectura del Sistema e Interfaz Gráfica**

La definición de la arquitectura del sistema permite comprender, tanto al personal técnico como a otros colaboradores, la lógica del funcionamiento del sistema.

La representación de la interfaz gráfica, en etapas tempranas del desarrollo, le permite al propietario del sistema establecer comentarios o sugerencias entorno a su propuesta.

#### **ARQUITECTURA DEL SISTEMA**

El servidor de la aplicación almacenará tanto la base de datos como la aplicación web java. La comunicación entre ambas se efectuará mediante un conector JDBC (ver Figura 1-III).



**Figura 1-III:** Arquitectura del Sistema – Sistema SISCOVE en el Servidor Web

**Fuente:** Patricia Herrera y Vanessa Vargas

#### **INTERFAZ GRÁFICA**

Se describirá brevemente la estructura visual planteada para el sistema web, así como el logotipo y colores corporativos.

#### **IDENTIDAD CORPORATIVA**

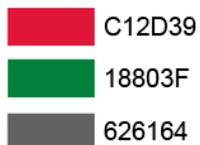
Con la finalidad de que el sistema cuente con una representación visual, se pidió la colaboración de un diseñador gráfico para la creación de un logotipo (Ver Figura 2-III).



**Figura 2-III:** Logotipo del sistema SISCOVE

**Fuente:** Patricia Herrera y Vanessa Vargas.

Se han considerado los colores corporativos de la ESPOCH:



### MAQUETACIÓN

La estructura visual de los contenidos del sistema, seguirán la disposición presentada en la Figura 3-III, considerando que se utilizarán elementos tipo layout (PrimeFaces) y CSS para la generación de dicha estructura.



**Figura 3-III:** Maquetación del sistema SISCOVE

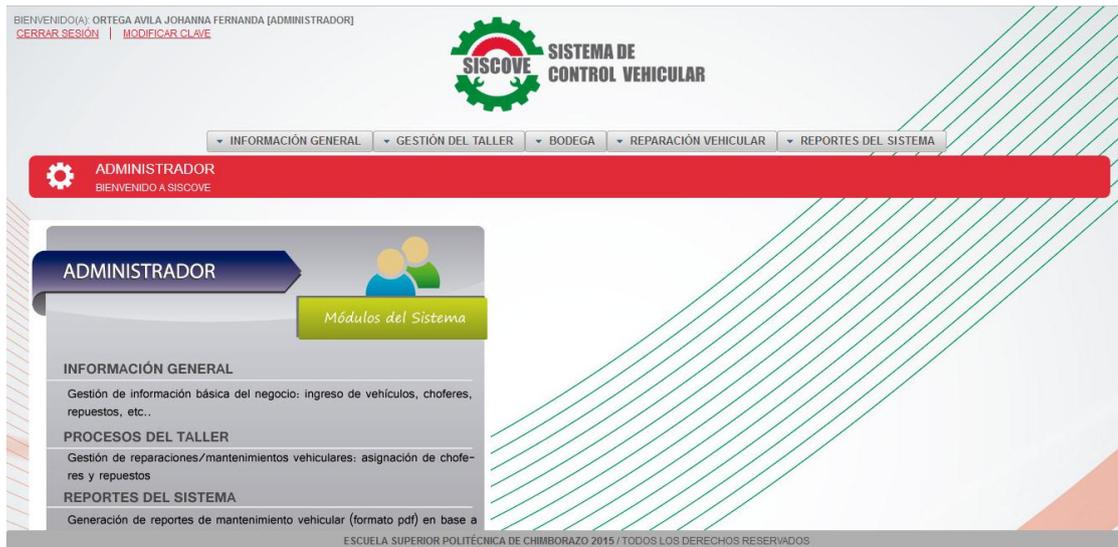
Fuente: Patricia Herrera y Vanessa Vargas

En la **sección superior** se colocará el MENÚ de navegación principal. Sus opciones se presentarán de acuerdo al rol del usuario que se encuentre utilizando el sistema. En la parte superior se mostrará el logotipo de SISCOVE (ver Figura 3-III).

La **sección central** contendrá los formularios, ventanas flotantes, tablas de datos y reportes con los que el usuario interactuará al obtener información del sistema.

La **sección inferior** presentará el nombre de la institución y el año de creación del sistema.

Un diseño preliminar de la interfaz puede visualizarse en la Figura 4-III.

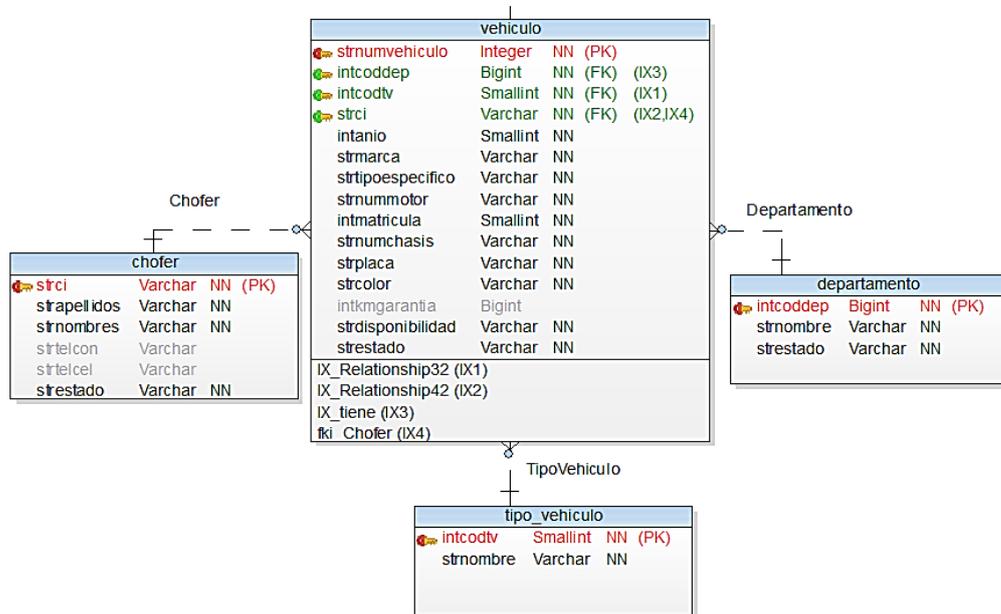


**Figura 4-III:** Diseño preliminar de la interfaz de usuario del Sistema SISCOVE

**Fuente:** Patricia Herrera y Vanessa Vargas.

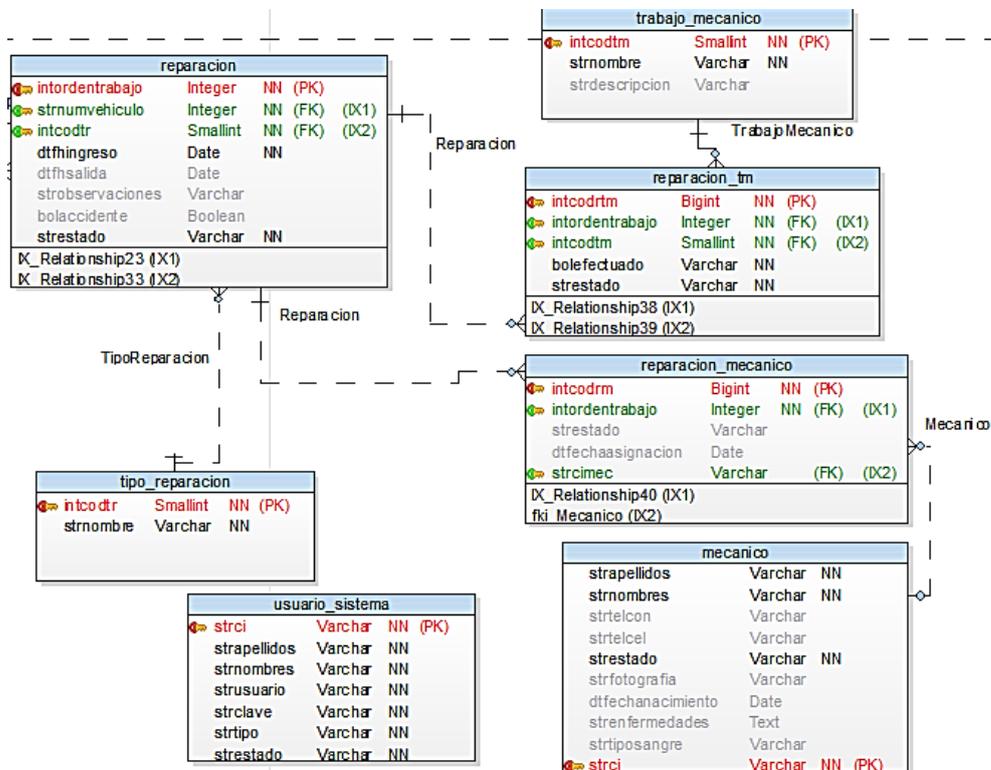
## Anexo IV: Estructuras de Datos – Base de Datos SISCOVE

### DIAGRAMAS DE CLASES



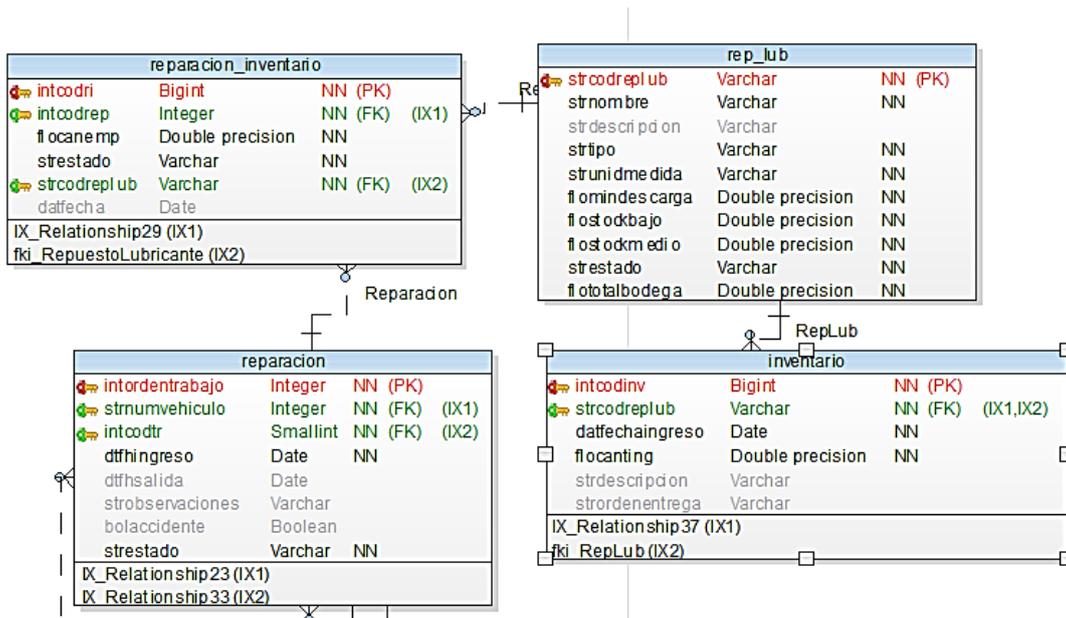
**Figura 1-IV:** Diagrama de Clases – Vehículo, tipo\_vehículo, departamento y chofer

Fuente: Patricia Herrera y Vanessa Vargas (Toad Modeler)



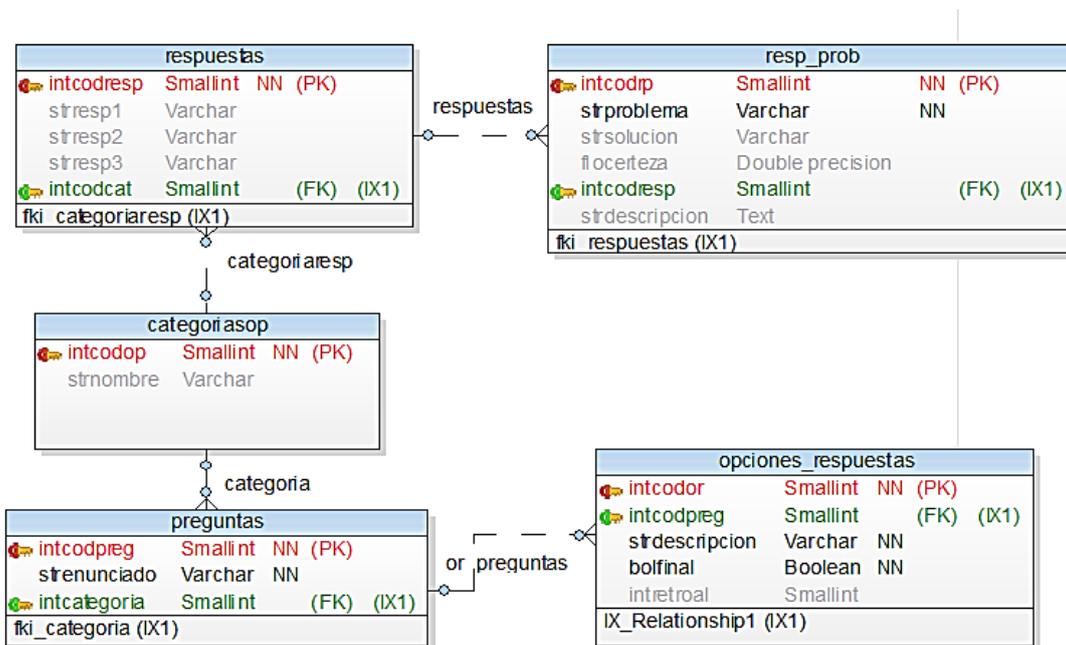
**Figura 2-IV:** Diagrama de Clases – Mecánico, Reparación Vehicular, Usuario\_Sistema y Trabajos\_Mecánicos

Fuente: Patricia Herrera y Vanessa Vargas (Toad Modeler)



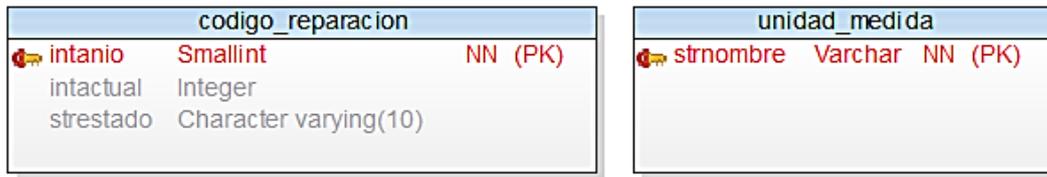
**Figura 3-IV:** Diagrama de Clases – Reparación e Inventario

Fuente: Patricia Herrera y Vanessa Vargas (Toad Modeler)



**Figura 4-IV:** Diagrama de Clases – Diagnóstico de fallas mecánicas

Fuente: Patricia Herrera y Vanessa Vargas (Toad Modeler)



**Figura 5-IV:** Diagrama de Clases – Tablas adicionales no relacionadas

Fuente: Patricia Herrera y Vanessa Vargas (Toad Modeler)

## DICCIONARIO DE DATOS

**Tabla 1-IV.** Diccionario de datos SISCOVE

TABLAS	CAMPOS			
	Nombre	Tipo de Dato	Llave primaria	Not Null
DEPARTAMENTO	intCodDep	Bigint	VERDADERO	VERDADERO
	strNombre	Varchar	FALSO	VERDADERO
	strEstado	Bigint	FALSO	FALSO
VEHICULO	strNumVehiculo	Integer	VERDADERO	VERDADERO
	intCodDep	Bigint	FALSO	VERDADERO
	intCodTV	Smallint	FALSO	FALSO
	strCI	Varchar	FALSO	FALSO
	intAnio	Smallint	FALSO	VERDADERO
	strMarca	Varchar	FALSO	VERDADERO
	strTipoEspecifico	Varchar	FALSO	FALSO
	strNumMotor	Varchar	FALSO	VERDADERO
	intMatricula	Smallint	FALSO	VERDADERO
	strNumChasis	Varchar	FALSO	VERDADERO
	strPlaca	Varchar	FALSO	VERDADERO
	strColor	Varchar	FALSO	FALSO
	intKmGarantia	Smallint	FALSO	FALSO
	strDisponibilidad	Varchar	FALSO	VERDADERO
strEstado	Varchar	FALSO	VERDADERO	
CHOFER	strCI	Varchar	VERDADERO	VERDADERO
	strApellidos	Varchar	FALSO	VERDADERO
	strNombres	Varchar	FALSO	VERDADERO
	strTelCon	Varchar	FALSO	FALSO
	strTelCel	Varchar	FALSO	FALSO
	strEstado	Varchar	FALSO	VERDADERO
MECANICO	strCI	Bigint	VERDADERO	VERDADERO
	strApellidos	Varchar	FALSO	VERDADERO
	strNombres	Varchar	FALSO	VERDADERO
	strTelCon	Varchar	FALSO	FALSO

	strTelCel	Varchar	FALSO	FALSO
	strEstado	Varchar	FALSO	VERDADERO
	dtFechaNacimiento	Bigint	FALSO	FALSO
	strEnfermedades	Bigint	FALSO	FALSO
	strTipoSangre	Bigint	FALSO	FALSO
<b>REPARACION</b>	intOrdenTrabajo	Integer	VERDADERO	VERDADERO
	strNumVehiculo	Integer	FALSO	VERDADERO
	intCodTR	Smallint	FALSO	VERDADERO
	dtFHIngreso	Timestamp	FALSO	VERDADERO
	dtFHSalida	Timestamp	FALSO	VERDADERO
	strObservaciones	Varchar	FALSO	FALSO
	bolAccidente	Boolean	FALSO	VERDADERO
	strEstado	Bigint	FALSO	VERDADERO
<b>REP_LUB</b>	strCodRepLub	Varchar	VERDADERO	VERDADERO
	strNombre	Varchar	FALSO	VERDADERO
	strDescripcion	Varchar	FALSO	FALSO
	strTipo	Varchar	FALSO	VERDADERO
	strUnidMedida	Varchar	FALSO	VERDADERO
	floMinDescarga	Double precision	FALSO	VERDADERO
	floStockBajo	Double precision	FALSO	VERDADERO
	floStockMedio	Double precision	FALSO	VERDADERO
	strEstado	Varchar	FALSO	VERDADERO
	floTotalBodega	Bigint	FALSO	VERDADERO
<b>INVENTARIO</b>	intCodInv	Bigint	VERDADERO	VERDADERO
	strCodRepLub	Varchar	FALSO	VERDADERO
	datFechaIngreso	Date	FALSO	VERDADERO
	floCantIng	Double precision	FALSO	VERDADERO
	strDescripcion	Varchar	FALSO	FALSO
	strOrdenEntrega	Bigint	FALSO	FALSO
<b>REPARACION_INVENTARIO</b>	intCodRI	Bigint	VERDADERO	VERDADERO
	floCanEmp	Smallint	FALSO	VERDADERO
	datFecha	Bigint	FALSO	VERDADERO
	strCodRepLub	Varchar	FALSO	FALSO
	intCodRep	Integer	FALSO	FALSO
	strEstado	Bigint	FALSO	VERDADERO
<b>TIPO_VEHICULO</b>	intCodTV	Smallint	VERDADERO	VERDADERO
	strNombre	Varchar	FALSO	VERDADERO
<b>TIPO_REPARACION</b>	intCodTR	Smallint	VERDADERO	VERDADERO
	strNombre	Varchar	FALSO	VERDADERO
<b>USUARIO_SISTEMA</b>	strCI	Varchar	VERDADERO	VERDADERO
	strApellidos	Varchar	FALSO	VERDADERO
	strNombres	Varchar	FALSO	VERDADERO
	strUsuario	Varchar	FALSO	VERDADERO

	strClave	Varchar	FALSO	VERDADERO
	strTipo	Varchar	FALSO	VERDADERO
	strEstado	Varchar	FALSO	VERDADERO
<b>TRABAJOS_MECANICOS</b>	intCodTM	Smallint	VERDADERO	VERDADERO
	strNombre	Varchar	FALSO	VERDADERO
	strDescripcion	Varchar	FALSO	FALSO
<b>REPARACION_TM</b>	intCodRTM	Bigint	VERDADERO	VERDADERO
	intOrdenTrabajo	Integer	FALSO	VERDADERO
	intCodTM	Smallint	FALSO	VERDADERO
	bolEfectuado	Varchar	FALSO	VERDADERO
	strEstado	Varchar	FALSO	VERDADERO
<b>REPARACION_MECANICO</b>	intCodRM	Bigint	VERDADERO	VERDADERO
	intOrdenTrabajo	Integer	FALSO	VERDADERO
	strObservaciones	Varchar	FALSO	FALSO
	strCI	Bigint	FALSO	FALSO
	dtFechaAsignacion	Bigint	FALSO	FALSO
<b>RESP_PROB</b>	intCodRP	Smallint	VERDADERO	VERDADERO
	strProblema	Varchar	FALSO	VERDADERO
	strSolucion	Varchar	FALSO	VERDADERO
	floCerteza	Double precision	FALSO	VERDADERO
	intCodRes	Bigint	FALSO	FALSO
	strDescripcion	Bigint	FALSO	FALSO
<b>OPCIONES_RESPUESTA</b>	intCodOR	Smallint	VERDADERO	VERDADERO
	intCodPreg	Smallint	FALSO	FALSO
	strDecripcion	Varchar	FALSO	VERDADERO
	bolFinal	Boolean	FALSO	VERDADERO
	intRetroal	Smallint	FALSO	FALSO
<b>PREGUNTAS</b>	intCodPreg	Smallint	VERDADERO	VERDADERO
	strEnunciado	Varchar	FALSO	VERDADERO
	intCategoria	Bigint	FALSO	FALSO
<b>RESPUESTAS</b>	intCodRes	Bigint	VERDADERO	VERDADERO
	strResp1	Varchar	FALSO	VERDADERO
	strResp2	Varchar	FALSO	VERDADERO
	strResp3	Varchar	FALSO	VERDADERO
	intCodCat	Bigint	FALSO	FALSO
<b>CATEGORIASOP</b>	intCodOp	Bigint	VERDADERO	VERDADERO
	strNombre	Bigint	FALSO	VERDADERO
<b>CODIGO_REPARACION</b>	intAnio	Smallint	VERDADERO	VERDADERO
	intActual	Integer	FALSO	FALSO
	strEstado	Varchar	FALSO	FALSO
<b>UNIDAD_MEDIDA</b>	strNombre	Varchar	VERDADERO	VERDADERO

Fuente: Patricia Herrera y Vanessa Vargas (Toad Modeler)

**Tabla 2-IV.** Relaciones entre tablas

<b>Id</b>	<b>Entidad Padre</b>	<b>Entidad Hijo</b>
1	DEPARTAMENTO	VEHICULO
2	VEHICULO	REPARACION
3	TIPO_VEHICULO	VEHICULO
4	TIPO_REPARACION	REPARACION
5	REP_LUB	INVENTARIO
6	REPARACION	REPARACION_TM
7	TRABAJOS_MECANICOS	REPARACION_TM
8	REPARACION	REPARACION_MECANICO
9	CHOFER	VEHICULO
10	PREGUNTAS	OPCIONES_RESPUESTA
11	RESPUESTAS	RESP_PROB
12	MECANICO	REPARACION_MECANICO
13	REP_LUB	REPARACION_INVENTARIO
14	REPARACION	REPARACION_INVENTARIO
15	CATEGORIASOP	RESPUESTAS
16	CATEGORIASOP	PREGUNTAS

Fuente: Patricia Herrera y Vanessa Vargas (Toad Modeler)

## **SCRIPT DE LA BASE DE DATOS**

**Tabla 3-IV.** Script de la base de datos

<b>ID</b>	<b>TABLA</b>	<b>SQL</b>
1	CATEGORIASOP	<pre>CREATE TABLE public.categoriasop (   intcodop smallint NOT NULL,   strnombre character varying,   CONSTRAINT "Codop" PRIMARY KEY (intcodop) ) WITH (   OIDS=FALSE ); ALTER TABLE public.categoriasop OWNER TO postgres;</pre>
2	CHOFER	<pre>CREATE TABLE public.chofer (   strci character varying NOT NULL,   strapellidos character varying NOT NULL,   strnombres character varying NOT NULL,   strtelcon character varying,   strtelcel character varying,   strestado character varying NOT NULL,   CONSTRAINT "CI" PRIMARY KEY (strci) ) WITH (   OIDS=FALSE ); ALTER TABLE public.chofer OWNER TO postgres;</pre>
3	CODIGO_REPARACION	<pre>CREATE TABLE public.codigo_reparacion (</pre>

		<pre> intanio smallint NOT NULL, intactual integer, strestado character varying(10), CONSTRAINT intanio PRIMARY KEY (intanio) ) WITH (   OIDS=FALSE ); ALTER TABLE public.codigo_reparacion   OWNER TO postgres; </pre>
4	DEPARTAMENTO	<pre> CREATE TABLE public.departamento (   intcoddep      bigint      NOT      NULL      DEFAULT nextval('pk_departamento':regclass),   strnombre character varying NOT NULL,   strestado character varying NOT NULL,   CONSTRAINT "CodDep" PRIMARY KEY (intcoddep) ) WITH (   OIDS=FALSE ); ALTER TABLE public.departamento   OWNER TO postgres; </pre>
5	INVENTARIO	<pre> CREATE TABLE public.inventario (   intcodinv      bigint      NOT      NULL      DEFAULT nextval('pk_inventario':regclass),   strcodreplub character varying NOT NULL,   datfechaingreso date NOT NULL,   flocanting double precision NOT NULL,   strdescripcion character varying,   strordenentrega character varying,   CONSTRAINT "CodInv" PRIMARY KEY (intcodinv),   CONSTRAINT "RepLub" FOREIGN KEY (strcodreplub)   REFERENCES public.rep_lub (strcodreplub) MATCH SIMPLE   ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH (   OIDS=FALSE ); ALTER TABLE public.inventario   OWNER TO postgres;  -- Index: public."IX_Relationship37" -- DROP INDEX public."IX_Relationship37";  CREATE INDEX "IX_Relationship37"   ON public.inventario   USING btree   (strcodreplub COLLATE pg_catalog."default");  -- Index: public."fki_RepLub" -- DROP INDEX public."fki_RepLub";  CREATE INDEX "fki_RepLub"   ON public.inventario   USING btree   (strcodreplub COLLATE pg_catalog."default"); </pre>
6	MECANICO	<pre> CREATE TABLE public.mecanico (   strapellidos character varying NOT NULL,   strnombres character varying NOT NULL,   strtelcon character varying,   strtelcel character varying,   strestado character varying NOT NULL, </pre>

		<pre> strfotografia 106 carácter varying, dtfechanacimiento date, strenfermedades text, strtiposangre character varying, strci character varying NOT NULL, CONSTRAINT "CodCi" PRIMARY KEY (strci) ) WITH (   OIDS=FALSE ); ALTER TABLE public.mecanico   OWNER TO postgres; </pre>
7	OPCIONES_RESPUESTAS	<pre> CREATE TABLE public.opciones_respuestas (   intcodor smallint NOT NULL,   intcodpreg smallint,   strdescripcion character varying NOT NULL,   bolfinal boolean NOT NULL,   intretroal smallint,   CONSTRAINT "CodOR" PRIMARY KEY (intcodor) ) WITH (   OIDS=FALSE ); ALTER TABLE public.opciones_respuestas   OWNER TO postgres;  -- Index: public."IX_Relationship1"  -- DROP INDEX public."IX_Relationship1";  CREATE INDEX "IX_Relationship1"   ON public.opciones_respuestas   USING btree   (intcodpreg); </pre>
8	PREGUNTAS	<pre> CREATE TABLE public.preguntas (   intcodpreg smallint NOT NULL,   strenunciado character varying NOT NULL,   intcategoria smallint,   CONSTRAINT "CodPreg" PRIMARY KEY (intcodpreg),   CONSTRAINT categoria FOREIGN KEY (intcategoria)     REFERENCES public.categoriasop (intcodop) MATCH SIMPLE   ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH (   OIDS=FALSE ); ALTER TABLE public.preguntas   OWNER TO postgres;  -- Index: public.fki_categoria  -- DROP INDEX public.fki_categoria;  CREATE INDEX fki_categoria   ON public.preguntas   USING btree   (intcategoria); </pre>
9	REP_LUB	<pre> CREATE TABLE public.rep_lub (   strcodreplub character varying NOT NULL,   strnombre character varying NOT NULL,   strdescripcion character varying,   strtipo character varying NOT NULL,   strunidmedida character varying NOT NULL,   flomindescarga double precision NOT NULL, </pre>

		<pre> flostockbajo double precision NOT NULL, flostockmedio double precision NOT NULL, strestado character varying NOT NULL, flototalbodega double precision NOT NULL DEFAULT 0, CONSTRAINT "CodRepLub" PRIMARY KEY (strcodreplub) ) WITH (   OIDS=FALSE ); ALTER TABLE public.rep_lub   OWNER TO postgres; </pre>
10	REPARACION	<pre> CREATE TABLE public.reparacion (   intordentrabajo integer NOT NULL,   strnumvehiculo integer NOT NULL,   intcodtr smallint NOT NULL,   dtfhingreso date NOT NULL,   dtfhsalida date,   strobservaciones character varying,   bolaccidente boolean,   strestado character varying NOT NULL,   CONSTRAINT "OrdenTrabajo" PRIMARY KEY (intordentrabajo),   CONSTRAINT "TipoReparacion" FOREIGN KEY (intcodtr)     REFERENCES public.tipo_reparacion (intcodtr) MATCH SIMPLE     ON UPDATE NO ACTION ON DELETE NO ACTION,   CONSTRAINT "Vehiculo" FOREIGN KEY (strnumvehiculo)     REFERENCES public.vehiculo (strnumvehiculo) MATCH SIMPLE     ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH (   OIDS=FALSE ); ALTER TABLE public.reparacion   OWNER TO postgres;  -- Index: public."IX_Relationship23" -- DROP INDEX public."IX_Relationship23";  CREATE INDEX "IX_Relationship23"   ON public.reparacion   USING btree   (strnumvehiculo);  -- Index: public."IX_Relationship33" -- DROP INDEX public."IX_Relationship33";  CREATE INDEX "IX_Relationship33"   ON public.reparacion   USING btree   (intcodtr); </pre>
11	REPARACION_INVENTARIO	<pre> CREATE TABLE public.reparacion_inventario (   intcodri          bigint          NOT NULL          DEFAULT nextval('pk_reparacioninventario::regclass),   intcodrep integer NOT NULL,   flocanemp double precision NOT NULL,   strestado character varying NOT NULL,   strcodreplub character varying NOT NULL,   datfecha date,   CONSTRAINT "CodRI" PRIMARY KEY (intcodri),   CONSTRAINT "Reparacion" FOREIGN KEY (intcodrep)     REFERENCES public.reparacion (intordentrabajo) MATCH SIMPLE     ON UPDATE NO ACTION ON DELETE NO ACTION,   CONSTRAINT "RepuestoLubricante" FOREIGN KEY (strcodreplub) </pre>

		<pre> REFERENCES public.rep_lub (strcodreplub) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH (   OIDS=FALSE ); ALTER TABLE public.reparacion_inventario   OWNER TO postgres;  -- Index: public."IX_Relationship29" -- DROP INDEX public."IX_Relationship29";  CREATE INDEX "IX_Relationship29" ON public.reparacion_inventario USING btree (intcodrep);  -- Index: public."fki_RepuestoLubricante" -- DROP INDEX public."fki_RepuestoLubricante";  CREATE INDEX "fki_RepuestoLubricante" ON public.reparacion_inventario USING btree (strcodreplub COLLATE pg_catalog."default"); </pre>
12	REPARACION_MECANICO	<pre> CREATE TABLE public.reparacion_mecanico (   intcodrm          bigint          NOT NULL          DEFAULT nextval('pk_reparacionmecanico':regclass),   intordentrabajo  integer          NOT NULL,   strentado        character varying,   dtfechaasignacion date,   strcimec         character varying,   CONSTRAINT "CodRM" PRIMARY KEY (intcodrm),   CONSTRAINT "Mecanico" FOREIGN KEY (strcimec) REFERENCES public.mecanico (strci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,   CONSTRAINT "Reparacion" FOREIGN KEY (intordentrabajo) REFERENCES public.reparacion (intordentrabajo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH (   OIDS=FALSE ); ALTER TABLE public.reparacion_mecanico   OWNER TO postgres;  -- Index: public."IX_Relationship40" -- DROP INDEX public."IX_Relationship40";  CREATE INDEX "IX_Relationship40" ON public.reparacion_mecanico USING btree (intordentrabajo);  -- Index: public."fki_Mecanico" -- DROP INDEX public."fki_Mecanico";  CREATE INDEX "fki_Mecanico" ON public.reparacion_mecanico USING btree (strcimec COLLATE pg_catalog."default"); </pre>

13	REPARACION_TM	<pre> CREATE TABLE public.reparacion_tm (   intcodrtm      bigint      NOT      NULL      DEFAULT nextval('pk_reparaciontm':regclass),   intordentrabajo integer NOT NULL,   intcodtm       smallint NOT NULL,   bolectuado    character varying NOT NULL,   strechado     character varying NOT NULL,   CONSTRAINT "CodRTM" PRIMARY KEY (intcodrtm),   CONSTRAINT "Reparacion" FOREIGN KEY (intordentrabajo) REFERENCES public.reparacion (intordentrabajo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,   CONSTRAINT "TrabajoMecanico" FOREIGN KEY (intcodtm) REFERENCES public.trabajo_mecanico (intcodtm) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH (   OIDS=FALSE ); ALTER TABLE public.reparacion_tm OWNER TO postgres;  -- Index: public."IX_Relationship38"  -- DROP INDEX public."IX_Relationship38";  CREATE INDEX "IX_Relationship38" ON public.reparacion_tm USING btree (intordentrabajo);  -- Index: public."IX_Relationship39"  -- DROP INDEX public."IX_Relationship39";  CREATE INDEX "IX_Relationship39" ON public.reparacion_tm USING btree (intcodtm); </pre>
----	---------------	--

14	RESP_PROB	<pre> CREATE TABLE public.resp_prob (   intcodrp smallint NOT NULL,   strproblema character varying NOT NULL,   strsolucion character varying,   flocerteza double precision,   intcodresp smallint,   strdescripcion text,   CONSTRAINT "CodRP" PRIMARY KEY (intcodrp),   CONSTRAINT respuestas FOREIGN KEY (intcodresp)     REFERENCES public.respuestas (intcodresp) MATCH SIMPLE     ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH (   OIDS=FALSE ); ALTER TABLE public.resp_prob   OWNER TO postgres;  -- Index: public.fki_respuestas  -- DROP INDEX public.fki_respuestas;  CREATE INDEX fki_respuestas   ON public.resp_prob   USING btree   (intcodresp); </pre>
15	RESPUESTAS	<pre> CREATE TABLE public.respuestas (   intcodresp smallint NOT NULL,   strresp1 character varying,   strresp2 character varying,   strresp3 character varying,   intcodcat smallint,   CONSTRAINT codresp PRIMARY KEY (intcodresp),   CONSTRAINT categoriaresp FOREIGN KEY (intcodcat)     REFERENCES public.categoriasop (intcodop) MATCH SIMPLE     ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH (   OIDS=FALSE ); ALTER TABLE public.respuestas   OWNER TO postgres;  -- Index: public.fki_categoriaresp  -- DROP INDEX public.fki_categoriaresp;  CREATE INDEX fki_categoriaresp   ON public.respuestas   USING btree   (intcodcat); </pre>
16	TIPO_REPARACION	<pre> CREATE TABLE public.tipo_reparacion (   intcodtr smallint NOT NULL DEFAULT nextval('pk_tiporeparacion':regclass),   strnombre character varying NOT NULL,   CONSTRAINT "CodTR" PRIMARY KEY (intcodtr) ) WITH (   OIDS=FALSE ); ALTER TABLE public.tipo_reparacion   OWNER TO postgres; </pre>

17	TIPO_VEHICULO	<pre>CREATE TABLE public.tipo_vehiculo (   intcodtv      smallint      NOT      NULL      DEFAULT nextval('pk_tipovehiculo':regclass),   strnombre character varying NOT NULL,   CONSTRAINT "CodTV" PRIMARY KEY (intcodtv) ) WITH (   OIDS=FALSE ); ALTER TABLE public.tipo_vehiculo OWNER TO postgres;</pre>
18	TRABAJO_MECANICO	<pre>CREATE TABLE public.trabajo_mecanico (   intcodtm      smallint      NOT      NULL      DEFAULT nextval('pk_trabajomecanico':regclass),   strnombre character varying NOT NULL,   strdescripcion character varying,   CONSTRAINT "CodTM" PRIMARY KEY (intcodtm) ) WITH (   OIDS=FALSE ); ALTER TABLE public.trabajo_mecanico OWNER TO postgres;</pre>
19	UNIDAD_MEDIDA	<pre>CREATE TABLE public.unidad_medida (   strnombre character varying NOT NULL,   CONSTRAINT unidad_medida_pkey PRIMARY KEY (strnombre) ) WITH (   OIDS=FALSE ); ALTER TABLE public.unidad_medida OWNER TO postgres;</pre>
20	USUARIO_SISTEMA	<pre>CREATE TABLE public.usuario_sistema (   strci character varying NOT NULL,   strapellidos character varying NOT NULL,   strnombres character varying NOT NULL,   strusuario character varying NOT NULL,   strclave character varying NOT NULL,   strtipo character varying NOT NULL,   strestado character varying NOT NULL,   CONSTRAINT "CI_us" PRIMARY KEY (strci) ) WITH (   OIDS=FALSE ); ALTER TABLE public.usuario_sistema OWNER TO postgres;</pre>
21	VEHICULO	<pre>CREATE TABLE public.vehiculo (   strnumvehiculo integer NOT NULL,   intcoddep bigint NOT NULL,   intcodtv smallint NOT NULL,   strci character varying NOT NULL,   intanio smallint NOT NULL,   strmarca character varying NOT NULL,   strtipoespecifico character varying NOT NULL,   strnummotor character varying NOT NULL,   intmatricula smallint NOT NULL,   strnumchasis character varying NOT NULL,   strplaca character varying NOT NULL,   strcolor character varying NOT NULL,   intkmgarantia bigint,   strdisponibilidad character varying NOT NULL,</pre>

		<pre> strestado character varying NOT NULL, CONSTRAINT "NumVehiculo" PRIMARY KEY (strnumvehiculo), CONSTRAINT "Chofer" FOREIGN KEY (strci) REFERENCES public.chofer (strci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT "Departamento" FOREIGN KEY (intcoddep) REFERENCES public.departamento (intcoddep) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT "TipoVehiculo" FOREIGN KEY (intcodtv) REFERENCES public.tipo_vehiculo (intcodtv) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH ( OIDS=FALSE ); ALTER TABLE public.vehiculo OWNER TO postgres;  -- Index: public."IX_Relationship32"  -- DROP INDEX public."IX_Relationship32";  CREATE INDEX "IX_Relationship32" ON public.vehiculo USING btree (intcodtv);  -- Index: public."IX_Relationship42"  -- DROP INDEX public."IX_Relationship42";  CREATE INDEX "IX_Relationship42" ON public.vehiculo USING btree (strci COLLATE pg_catalog."default");  -- Index: public."IX_tiene"  -- DROP INDEX public."IX_tiene";  CREATE INDEX "IX_tiene" ON public.vehiculo USING btree (intcoddep);  -- Index: public."fki_Chofer"  -- DROP INDEX public."fki_Chofer";  CREATE INDEX "fki_Chofer" ON public.vehiculo USING btree (strci COLLATE pg_catalog."default"); </pre>
--	--	---

**Fuente:** Patricia Herrera y Vanessa Vargas (pgAdmin)

**Anexo V: Encuesta para desarrollo de Módulo de Diagnóstico Vehicular**



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA  
ESCUELA DE INGENIERÍA EN SISTEMAS

---

**ENCUESTA DE INVESTIGACIÓN**

**ANTECEDENTES:** EN EL DESARROLLO DEL PROYECTO DE TESIS *"IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN VEHICULAR CON JAVA SERVER FACES PARA EL DEPARTAMENTO DE MANTENIMIENTO VEHICULAR DE LA ESPOCH"*, SE REQUIERE CONOCER LA PERCEPCIÓN DE LOS ESTUDIANTES DE LA ESCUELA DE INGENIERÍA AUTOMOTRIZ RESPECTO AL DISEÑO DEL MÓDULO DE DIAGNÓSTICO VEHICULAR, EL CUAL FORMA PARTE DEL SISTEMA.

**INSTRUCCIONES:**

- LEA DETENIDAMENTE LAS PREGUNTAS Y CONTESTE EN LOS ESPACIOS PROVISTOS PARA SUS RESPUESTAS.
- EVITE DEJAR ALGUNA PREGUNTA SIN SU RESPECTIVA RESPUESTA

**DATOS GENERALES:**

**Nivel Actual (Semestre):** \_\_\_\_\_

**CUESTIONARIO:**

1. **¿Cuáles son las áreas fundamentales sobre las cuales se puede efectuar un diagnóstico vehicular? Cite al menos 2.**  
\_\_\_\_\_  
\_\_\_\_\_
2. **¿Qué tipo de información considera ud. sería útil incluir dentro de un módulo informatizado de diagnóstico vehicular, una vez que se ha detectado el posible problema del vehículo?**  
\_\_\_\_\_  
\_\_\_\_\_
3. **¿Considera útil la generación de este módulo? Justifique su respuesta.**

Si  No

\_\_\_\_\_  
\_\_\_\_\_

**GRACIAS POR SU COLABORACIÓN**



**SISTEMA DE  
CONTROL VEHICULAR**

# MANUAL DE USUARIO

MANEJO DEL SOFTWARE



ESCUELA SUPERIOR POLITÉCNICA  
DE CHIMBORAZO



## AVISOS

---

© 2016 Escuela Superior Politécnica de Chimborazo. Todos los derechos reservados. Ninguna parte de este manual podrá ser reproducida o transcrita.

SISCOVE es un sistema web desarrollado en colaboración con la Escuela Superior Politécnica de Chimborazo.

**Colaboradores:**

Tlgo. Gustavo Tapia (Usuario Principal del Proyecto)

Ing. Raúl Rosero (Director del Proyecto)

Patricia Herrera y Vannesa Vargas (Programadoras)

# TABLA DE CONTENIDO

<b>INTRODUCCIÓN</b> .....	4
<b>CAPÍTULO I: REQUISITOS</b> .....	5
HARDWARE .....	5
SOFTWARE.....	5
<b>CAPÍTULO II: EJECUCIÓN E INGRESO AL SISTEMA</b> .....	6
EJECUCIÓN DEL SISTEMA .....	6
INGRESO AL SISTEMA .....	6
<b>CAPÍTULO III: INTERFAZ DE USUARIO</b> .....	7
MENÚS [ Navegación por el sistema ] .....	7
FORMULARIOS [ Gestión de la información y generación de reportes ].....	7
MENSAJES DEL SISTEMA .....	8
ACTIVIDADES DEL GESTIÓN Y SUS COMANDOS.....	9
<b>CAPÍTULO IV: MÓDULOS DEL SISTEMA</b> .....	11
GESTIÓN GENERAL DE LA INFORMACIÓN .....	11
GESTIÓN DE PROCESOS DEL TALLER .....	11
REPORTES DEL SISTEMA Y FORMATOS.....	11
DIAGNÓSTICO VEHICULAR .....	11
<b>RECOMENDACIONES DE USO</b> .....	12
<b>SERVICIO AL CLIENTE</b> .....	13

# INTRODUCCIÓN

SISCOVE es un sistema web que automatiza el control del mantenimiento vehicular y del inventario del taller de mecánica de la ESPOCH. Su interfaz gráfica es intuitiva y amigable, lo cual facilita su uso. El presente manual permite un aprendizaje guiado sobre el manejo de sus diferentes módulos, así como la solución de problemas.

En el capítulo I encontrará los requisitos de instalación del sistema. En el capítulo II aprenderá a ejecutar la aplicación y acceder al sistema con un usuario y contraseña. El capítulo III describe los componentes de la interfaz de usuario y su correcta utilización. En el capítulo IV encontrará información sobre el manejo de los módulos del sistema y su lógica de negocios. Finalmente en el capítulo V se detallan ciertas recomendaciones a considerarse antes del uso del sistema. Si encuentra problemas que no pueda solucionar, sírvase encontrar en el capítulo VI información de contacto de técnicos especializados en el sistema.

# CAPÍTULO I: REQUISITOS

## HARDWARE

El sistema ha sido probado satisfactoriamente en un computador que cumple los siguientes requerimientos:

CARACTERÍSTICA	DESCRIPCIÓN
RAM	1GB de RAM
PROCESADOR	Intel Core 2duo de 2.8 GHz
Disco Duro	500GB

**TABLA 1:** Requisitos Hardware

## SOFTWARE

Para el correcto funcionamiento del sistema, el equipo servidor debe tener instalado los siguientes programas mínimos:

PROGRAMA	FUNCIONALIDAD	VERSIÓN (validad en pruebas)
Windows	Sistema Operativo	7 Profesional 64 bits
PostgreSQL	Motor de Base de Datos	9.2
Glassfish Server	Servidor de Páginas Web para Java	4.1
Mozilla Firefox	Navegador Web	47.0

**TABLA 2:** Requisitos Software Servidor

En cuanto al equipo cliente, este debe tener instalado un navegador web, de preferencia Firefox o Google Chrome debidamente actualizados.

# CAPÍTULO II: EJECUCIÓN E INGRESO AL SISTEMA

## EJECUCIÓN DEL SISTEMA

La ejecución correcta del sistema depende de la configuración del equipo servidor. En caso de que todos los servicios se encuentren correctamente configurados, la disponibilidad del sistema en la web será inmediata, luego de encender el servidor.

## INGRESO AL SISTEMA

1 En el equipo cliente, abra el navegador web de su preferencia y coloque la URL de la aplicación web. En caso de que el cliente y el servidor corran en el mismo computador, podrá accederse al sistema mediante la siguiente dirección:

<http://localhost:8080/siscove>

2 Seleccione una de las siguientes opciones, según su perfil de usuario: Formulario de Login (para el acceso al sistema del “administrador” y usuario “formato”) o Diagnóstico Vehicular (cualquier usuario o invitado).



FIGURA 1: Opciones de acceso al sistema

Para el ingreso al formulario de login, debe digitar el “nombre de usuario” y “clave” provista por el administrador. Para ingresar al módulo de diagnóstico vehicular únicamente debe hacer clic en el botón “simulador”. Finalmente, si el ingreso fue satisfactorio, podrá observar una pantalla como la figura 2:



FIGURA 2: Acceso satisfactorio al sistema

# CAPÍTULO III: INTERFAZ DE USUARIO

## MENÚS [ Navegación por el sistema ]

### MENÚ PRINCIPAL

La navegación a través del sistema puede realizarse mediante el “menú principal”. Éste se encuentra en la parte superior central de la pantalla, debajo del logotipo. Al hacer clic en alguna de sus opciones, se despliega un submenú el cual, al seleccionarlo, lo redirigirá a otras pantallas del sistema.



FIGURA 3: Menú principal

### MENÚ SUPERIOR

Opciones del usuario del sistema, como “cerrar sesión” o “modificar clave”, se encuentran en la parte superior izquierda de la pantalla, en el “menú superior”. Al hacer clic en “cerrar sesión”, se abrirá el index o home del sistema. La opción “modificar clave” nos mostrará un formulario para el cambio de contraseña.



FIGURA 4: Menú superior

## FORMULARIOS [ Gestión de la información y generación de reportes ]

### FORMULARIOS DE GESTIÓN Y BOTONES

Cuando haga clic en botones como “nuevo” o “editar”, le aparecerá una ventana flotante en la cual deberá ingresar la información solicitada (etiquetas verdes: campos obligatorios, etiquetas grises: campos no obligatorios).



FIGURA 5: Formulario de Gestión

### FORMULARIOS DE GENERACIÓN DE REPORTE

En el proceso de generación de reportes, deberá ingresarse la información solicitada en los formularios de datos. Estos datos ingresados actúan como filtros, los cuales priorizan ciertos datos de búsqueda. Generalmente se exportarán en formato pdf.

IMPORTANTE: Los campos marcados (\*) son requeridos.

	DESDE	HASTA
INGRESO VEHÍCULO(*)	2016/07/06	2016/07/06
SALIDA VEHÍCULO(*)	2016/07/06	2016/07/06
MECÁNICO(*)	TODOS	
VEHÍCULO(*)	TODOS	
TIPO DE REPARACIÓN(*)	TODOS	
ACCIDENTE(*)	<input checked="" type="radio"/> TODOS <input type="radio"/> SI <input type="radio"/> NO	
TRABAJO SOLICITADO(*)	TODOS	
TRABAJO EFECTUADO(*)	TODOS	

Aceptar Cancelar

FIGURA 6: Formulario de generación de reportes

### MENSAJES DEL SISTEMA

El sistema lo alertará en caso de haber cometido errores en el ingreso de datos en los formularios. Existen dos tipos de alertas o mensajes: mensajes textuales (figura 7) y paneles de alerta (figura 8).



FIGURA 7: Mensajes textuales de error/acierto del sistema

Esos mensajes desaparecerán una vez se actualice la página web (F5), o cuando el proceso termine satisfactoriamente y sin errores.



FIGURA 8: Paneles de alerta/información del sistema

## ACTIVIDADES DEL GESTIÓN Y SUS COMANDOS

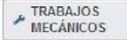
En la interfaz del sistema encontrará varios elementos o comandos que le ayudarán a la realización de las actividades que desee realizar. En la tabla 3 se resumen las acciones generales que puede realizar en el sistema.

ACCIÓN	DESCRIPCIÓN	COMANDO (ELEMENTO HTML)
NUEVO	Ingresar un registro (por ejemplo: un nuevo chofer o vehículo).	Botón NUEVO
EDITAR	Modificar de los datos de un registro (por ejemplo, cambiar la “marca” de un vehículo, debido a errores de digitación anteriores).	Botón EDITAR
VER DETALLE	Ver los datos del registro seleccionado.	Botón VER DETALLE
ACTIVAR / DESACTIVAR	Cambiar el “estado” de un registro. Si se “desactiva”, por ejemplo un mecánico, éste no podrá ser utilizado en ningún proceso (como el proceso de asignación de mecánicos a reparaciones vehiculares), hasta que sea activado.	Botón ACTIVAR DESACTIVAR
ELIMINAR	Eliminar un registro del sistema; es un proceso irreversible, por lo cual se encuentra disponible solo en algunas pantallas.	Botón ELIMINAR
FILTRADO DE REGISTROS	Obtener los registros de datos que cumplan con una serie de restricciones o filtros, como por ejemplo, todos los que tengan el “estado=activo”.	Filtro (varios tipos) ESTADO TODOS

REPORTES RESUMEN	Descargar reportes en Excel o pdf de los registros visualizados en pantalla (incluye la aplicación de filtros)	Enlaces [PDF] [EXCEL]
---------------------	--	--------------------------

**TABLA 3:** Actividades de gestión y sus comandos

Otros comandos de uso específico que ud. encontrará en el sistema, se resumen en la tabla 4.

ACCIÓN	DESCRIPCIÓN	MÓDULO / SUBMÓDULO	COMANDO (ELEMENTO HTML)
VER MECÁNICOS / REPUESTOS / TRABAJOS MECÁNICOS	Visualizar información relacionada a la orden de trabajo, pero ingresada en otras pantallas u opciones	Reparación Vehicular / Órdenes de trabajo	Botón   
DESCARGAR INFORMES ESPECÍFICOS	Obtener informes de control, en formato pdf o word.	Reparación Vehicular / Órdenes de trabajo  Reportes del sistema / ordenes de trabajo	Enlaces <a href="#">INFORME DE REPARACIÓN/P</a> <a href="#">INFORME DE REPARAC.WORD</a>
FILTRAR INFORMACIÓN PARA LA GENERACIÓN DE INFORMES	Ingresar datos requeridos en un formulario, a fin de que el sistema genere un reporte en baso a esos filtros.	Reportes del sistema / [Informes]	Formulario 

**TABLA 4:** Actividades de uso específico y sus comandos

# CAPÍTULO IV: MÓDULOS DEL SISTEMA

## GESTIÓN GENERAL DE LA INFORMACIÓN

---

Para gestionar la información general del sistema, como choferes, vehículos, dependencias, entre otros, se han implementado dos opciones en el menú principal:

**INFORMACIÓN GENERAL:** gestión de usuarios del sistema.

**GESTIÓN DEL TALLER:** gestión de mecánicos, choferes, vehículos, repuestos/lubricantes/aditivos y dependencias.

## GESTIÓN DE PROCESOS DEL TALLER

---

Para gestionar información específica de los procesos del taller (como órdenes de trabajo e inventarios), se han implementado dos opciones en el menú principal:

**BODEGA:** gestión de ingresos y egresos a bodega.

**REPARACIÓN VEHICULAR:** gestión de órdenes de trabajo, incluidos mecánicos y trabajos mecánicos asignados a cada reparación.

## REPORTES DEL SISTEMA Y FORMATOS

---

Para la generación y descarga de informes, el sistema tiene implementado en el menú principal la opción de "REPORTES DEL SISTEMA", mediante la cual se pueden generar los siguientes reportes:

**FORMATOS GENERALES:** descarga de documentos del sistema para su llenado manual (formato pdf) o mediante otro software informático (formato word).

**INFORMES:** descarga de documentos que contienen información registrada en el sistema (reportes preimpresos).

## DIAGNÓSTICO VEHICULAR

---

El sistema de diagnóstico vehicular permite, a cualquier usuario externo (usuario invitado), ingresar a un proceso de preguntas y respuestas, a fin de concluir con la falla mecánica de un vehículo. El proceso termina una vez que el sistema determina los posibles problemas del auto.

## RECOMENDACIONES DE USO

- En caso de que el sistema presente retardo en los tiempos de respuesta, se recomienda reiniciar tanto el equipo cliente como el servidor.
- Se sugiere emplear el navegador Mozilla Firefox para el acceso al sistema, además de actualizarlo frecuentemente.
- Si van a existir periodos amplios de inactividad, se sugiere cerrar la sesión una vez terminados los procesos planificados. Usted podrá volver a ingresar al sistema nuevamente sin problemas.

## SERVICIO AL CLIENTE

En caso requiera soporte técnico sobre el manejo del sistema, puede contactar a los desarrolladores a través de los siguientes medios:

**Técnico 1:** Patricia Herrera (0988103805)

**Técnico 2:** Vannesa Vargas (0999979648)

No se permite la reproducción total ni parcial de este manual sin la autorización de la ESPOCH

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

DESARROLLADO EN RIOBAMBA, ECUADOR 2016