

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



PROYECTO DE GRADO

**"SISTEMA DE CONTROL Y SEGUIMIENTO DEL USO DE
VEHÍCULOS VÍA WEB PARA LA EMPRESA CADEB"**

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

AUTORA: NIDIA INGRID MARTINEZ LARICO

TUTOR: LIC. MARIO LOAYZA MOLINA

REVISOR: LIC. GERMAN HUANCA TICONA

LA PAZ - BOLIVIA

2009

DEDICATORIA

*A Dios por estar siempre a mi lado por ser la
luz que guía mi camino.*

*A mis padres Esteban y Lidia por su
comprensión y apoyo incondicional en todo
momento.*

*A mis hermanos Fiorella, Edson y Samantha
por estar ahí brindándome el apoyo y aliento
para seguir adelante.*

Nidia Ingrid Martínez

AGRADECIMIENTOS

- ◆ A mi tutor Lic. Mario Loayza Molina por su colaboración y apoyo para la culminación del presente proyecto.
- ◆ A mi revisor Lic. Germán Huanca Ticona, por sus sugerencias, observaciones y correcciones, brindándome constante ánimo y consejos para culminar el proyecto.
- ◆ Lic. Jorge Vedia Barrón por la confianza, colaboración y la constante disponibilidad en el desarrollo del proyecto.
- ◆ Lic. Javier Machicado por su orientación y colaboración en el desarrollo de todo el proyecto.
- ◆ Al personal de CADEB, por brindarme su ayuda en todo el transcurso del desarrollo del proyecto.
- ◆ A mis padres por la paciencia y comprensión que tuvieron conmigo para la culminación de mis estudios.
- ◆ A mis hermanos por el aliento y ánimo que me dieron para que siga adelante.
- ◆ A mis Amigas Carla Quispe, Lidia Vega, Lisset Chuquimia y Jacky Chuquimia por el apoyo que me brindaron y por estar en los buenos y malos momentos.

Muchas Gracias.

RESUMEN

La Compañía Administradora de Empresas Bolivia - CADEB, es una institución privada que maneja un conglomerado de varias unidades de negocio con una administración centralizada, donde cada unidad se encarga de producir y comercializar sus servicios y productos. El área de CADEB Talleres es una de las áreas importantes dentro de la empresa ya que es la encargada de controlar y hacer un seguimiento de los vehículos que han estado en uso y en mantenimiento para diferentes trabajos.

En el presente proyecto de grado se propone el Sistema de Control y Seguimiento del Uso de Vehículos vía Web, el cual está desarrollado por la necesidad de tener un sistema que brinde un mejor apoyo en la administración de la información, teniéndola así disponible en cualquier momento.

La principal causa para el desarrollo de este proyecto, se basa en: el excesivo tiempo empleado en los procesos de generación de datos y la falta de información del mantenimiento de los vehículos lo que implica la utilización de recursos en demasía para la preparación de reportes y otros informes.

La solución al problema descrito anteriormente, es la implementación de un sistema automatizado diseñado bajo la arquitectura de tres capas, basado en la Web, esta última es una tecnología de última generación ya que su principal ventaja es la centralización de tareas en un servidor de aplicaciones.

Se emplearon diversas técnicas de desarrollo de software, las cuales fueron el uso de nuevas herramientas para el modelado, como es el UML, el uso de nuevas tecnologías como ser Sun Creator para el diseño de aplicaciones Web, y las herramientas de desarrollo (Plataforma, Lenguaje de aplicación, Gestor de Base Datos, Servidor de Páginas Web, etc.)

ÍNDICE

Descripción	Pág.
-------------	------

CAPÍTULO I

MARCO CONTEXTUAL

1.1.Introducción.....	1
1.2.Antecedentes	2
1.3.Antecedentes de Trabajos Similares.....	3
1.4.Análisis de la Problemática Actual.....	4
1.5.Planteamiento del Problema	5
1.5.1. Identificación del Problema.....	5
1.5.2. Problemas Secundarios.....	5
1.6.Objetivos	5
1.6.1. Objetivo General	5
1.6.2. Objetivos Específicos	6
1.7.Justificación.....	6
1.7.1. Justificación Social.....	6
1.7.2. Justificación Tecnológica.....	6
1.7.3. Justificación Económica.....	7
1.8.Alcances	7
1.9.Límites.....	8
1.10.Aportes	8
1.11.Metodología.....	9
1.12.Técnicas y Herramientas	9
1.12.1. Técnicas.....	9
1.12.2. Herramientas	9

CAPÍTULO II

MARCO TEÓRICO

2.1. Introducción al Proceso Unificado	10
2.1.1.Dirigido por casos de Uso	11
2.1.2.Centrado en la Arquitectura.....	11
2.1.3.Iterativo e Incremental.....	12
2.2. Estructura del RUP	13
2.3. Ciclos y Fases del RUP	14
2.3.1.Fase de Inicio.....	15
2.3.1.1.Planificación de las Iteraciones	15
2.3.1.2.Flujos de Trabajo	16
2.3.1.2.1.Recopilación de Requisitos	16
2.3.1.2.2.Análisis	16
2.3.1.2.3.Diseño	16
2.3.1.2.4.Implementación	16
2.3.1.2.5.Pruebas	16

2.3.1.3. Análisis y Gestión de Riesgos	17
2.3.1.4. Criterios de Evaluación	18
2.3.2.Fase de Elaboración.....	18
2.3.2.1.Flujos de Trabajo	19
2.3.2.1.1.Recopilacion de Requisitos	19
2.3.2.1.2.Analisis	19
2.3.2.1.3.Diseño	19
2.3.2.1.4.Implementacion	19
2.3.2.1.5.Pruebas	20
2.3.2.2.Criterios de Evaluación	20
2.3.3.Fase de Construcción.....	20
2.3.3.1.Flujos de Trabajo	21
2.3.3.2.Modelo del Negocio	21
2.3.3.3.Modelo del Dominio.....	21
2.3.3.4.Recopilacion de los requisitos	22
2.3.3.5.Analisis	25
2.3.3.6.Diseño	28
2.3.3.7.Implementacion	29
2.3.3.8.Pruebas	30
2.3.4.Fase de Transición.....	31
2.4.UML(Lenguaje Unificado de Modelado).....	32
2.4.1.Processos Basados en UML	32
2.4.2.Modelos y Diagramas	33
2.4.3.Diagramas que utiliza el UML	34
2.4.4.Elementos Comunes en todos los Diagramas.....	34
2.5.MVC(Modelo Vista Controlador)	44
2.6.Modelo de Tres Capas	45
2.6.1. Ventajas del Modelo Tres Capas	46
2.7. Seguridad.....	47
2.8. Métricas de Calidad.....	48

CAPÍTULO III

MARCO APLICATIVO

3.1.Fase de Inicio.....	52
3.1.1. Funciones Principales.....	52
3.1.2. Recopilación de Requisitos	54
3.1.3. Análisis	55
3.1.3.1. Arquitectura Candidata.....	55
3.1.4.Implementación	55
3.1.5.Pruebas	56
3.1.6.Analisis y Gestión de Riesgos	56
3.1.7.Evaluacion de la Fase de Inicio	57
3.2.Fase de Elaboración.....	58
3.2.1. Recopilación de Requisitos	58
3.2.1.1. Documento de Análisis.....	58
3.2.2.Análisis	60
3.2.2.1 Descripción de Casos de Uso de Primer Nivel.....	61
3.2.3.Diseño.....	64
3.2.4.Implementacion	66

3.2.5.Pruebas	67
3.2.6.Criterios de Evaluación	67
3.3.Fase de Construcción.....	68
3.3.1.Modelo del Negocio	68
3.3.2. Recopilación de Requisitos	69
3.3.3.Análisis	70
3.3.3.1.Construcción del Modelo Conceptual	71
3.3.3.2.Glosario	76
3.3.3.3.Diagramas de Secuencia.....	77
3.3.3.4.Diagramas de Contrato	78
3.3.4.Diseño.....	78
3.3.4.1.Diagramas de Colaboración	78
3.3.4.2.Diagrama de Clases	82
3.3.5.Implementación.....	84
3.3.5.1.Diagramas de Componentes	84
3.3.5.2.Diagramas de Despliegue	84
3.3.6.Pruebas	85
3.3.6.1.Caja Blanca.....	85
3.3.6.2.Caja Negra.....	87
3.3.6.3.Análisis de Resultados.....	89
3.3.6.4.Evaluación.....	89
3.3.7.Diagrama Navegacional	90
3.3.8.Descripción de Pantallas.....	91
3.4. Fase de Transición.....	94
3.4.1. Implementación del Sistema.....	94
3.4.2. Instalación del Sistema	94

CAPÍTULO IV

METRICAS DE CALIDAD

4.1.Funcionalidad	95
4.1.1. Métricas Basadas en el Punto Función	95
4.2.Confiabilidad	99
4.3. Facilidad de Mantenimiento	100
4.4. Portabilidad	102

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1.Conclusiones	103
5.2. Recomendaciones	104

Referencias Bibliográficas

Anexos

Documentación

ÍNDICE DE FIGURAS

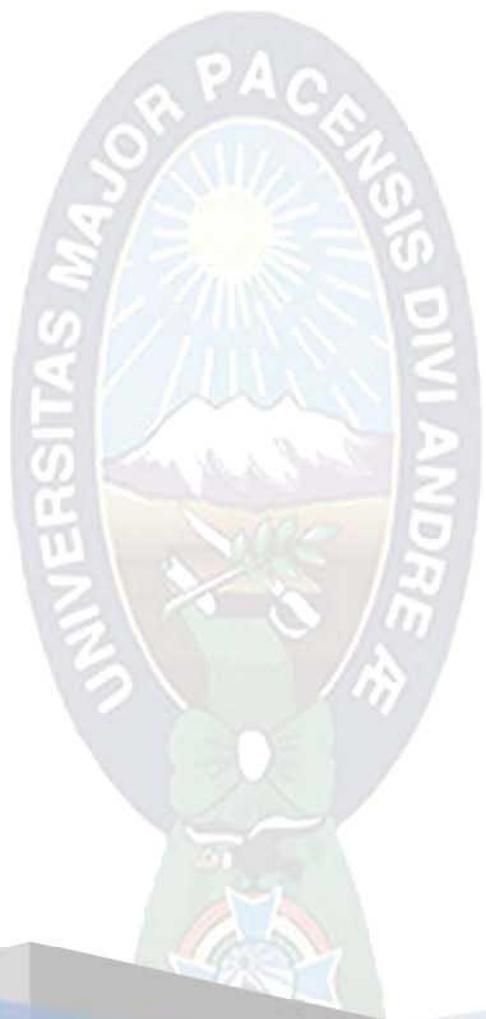
Contenido	Pág.
Figura 2.1 Esquema del Proceso de Desarrollo de Software	10
Figura 2.2 Iterativo Incremental	13
Figura 2.3 Estructura del RUP	14
Figura 2.4 Modelos del RUP	21
Figura 2.5 Modelo Físico	33
Figura 2.6 Diagramas que utiliza el modelo UML	34
Figura 2.7 Ejemplo de Nota	34
Figura 2.8 Ejemplo de Dependencia	35
Figura 2.9 Representación de una Clase	36
Figura 2.10 Ejemplo de Objeto	36
Figura 2.11 Ejemplo de Asociación	37
Figura 2.12 Ejemplo de Agregación	37
Figura 2.13 Ejemplo de Multiplicidad en asociación	38
Figura 2.14 Ejemplo herencia	38
Figura 2.15 Ejemplo de Caso de Uso	39
Figura 2.16 Ejemplo de Diagrama de secuencia	41
Figura 2.17 Ejemplo de Diagrama de colaboración	41
Figura 2.18 Ejemplo de Diagrama de estados	42
Figura 2.19 Ejemplo de Diagrama de actividades	43
Figura 2.20 Ejemplo de Diagrama de componentes	43
Figura 2.21 Ejemplo de Diagrama de despliegue	43
Figura 2.22 Arquitectura del MVC	44
Figura 2.23 Arquitectura de tres capas	45
Figura 3.1 Arquitectura Candidata	55
Figura 3.2 Diagrama de Casos de Uso de Primer Nivel	60
Figura 3.3 Representación de la clase vehículo	64
Figura 3.4 Diagrama de colaboración Registro Grupo de vehículos	66
Figura 3.5 Diagrama de colaboración Registro de Combustible	66
Figura 3.6 Modelo de casos de Uso del Negocio	68
Figura 3.7 Diagrama de Casos de Uso de Segundo Nivel	69
Figura 3.8 Modelo Conceptual Inicial del Dominio	73
Figura 3.9 Atributos en el Modelo Conceptual Inicial	73
Figura 3.10 Operaciones de las clases en el Modelo Conceptual Inicial	74
Figura 3.11 Modelo Conceptual	75
Figura 3.12 Diagrama de secuencia registro del uso de Vehículos	77
Figura 3.13 Diagrama de secuencia mantener información Vehículos	77

Figura 3.14 Diagrama de colaboración Registro del Uso de Vehículos	79
Figura 3.15 Diagrama de colaboración mantener información de vehículos.....	79
Figura 3.16 Diagrama de Clases.....	82
Figura 3.17 Modelo Objeto Relacional	83
Figura 3.18 Diagrama de Componentes	84
Figura 3.19 Diagrama de Despliegue	84
Figura 3.20 Grafo complejidad Ciclomática	86
Figura 3.21 Modelo de Prueba de Caja Negra.....	87
Figura 3.22 Diagrama de Navegación	90
Figura 3.23 Pantalla de Ingreso al Sistema.....	91
Figura 3.24 Pantalla del listado General de los Vehículos	92
Figura 3.25 Pantalla de registro de Vehículos	92
Figura 3.26 Pantalla de visualización de reportes de Vehículos.....	93
Figura 3.27 Pantalla de Factores de Distribución.....	93



ÍNDICE DE TABLAS

Contenido	Pág.
Tabla 2.1 Estructura de caso de Uso en formato de Alto Nivel.....	23
Tabla 2.2 Estructura de caso de Uso en formato expandido.....	24
Tabla 2.3 Pasos para describir casos de uso en su totalidad	24
Tabla 2.4 Pasos para construir un modelo conceptual	26
Tabla 2.5 Estructura de términos en un glosario	26
Tabla 2.6 Descripción de un contrato.....	28
Tabla 2.7. Parámetros de medición	50
Tabla 3.1 Planificación Fase de Inicio.....	54
Tabla 3.2 Tabla de Riesgos.....	57
Tabla 3.3 Caso de uso de Distribución de Gastos.....	61
Tabla 3.4 Caso de uso Mantener Información de Vehículos	61
Tabla 3.5 Caso de uso Registro de Vehículos	61
Tabla 3.6 Caso de uso Determinar Factores de Distribución.....	62
Tabla 3.7 Caso de uso Registrar consumo de Combustible.....	62
Tabla 3.8 Caso de uso Registrar mantenimiento	62
Tabla 3.9 Caso de uso Registrar parámetros	62
Tabla 3.10 Caso de uso Generar Asientos contables	63
Tabla 3.11 Tabla que representa al actor Jefe de Talleres	63
Tabla 3.12 Tabla que representa al actor Responsable de contabilidad.....	63
Tabla 3.13 Tabla que representa al actor Transcriptor	63
Tabla 3.14 Tabla para representar la interfaz del Jefe de Talleres.....	63
Tabla 3.15 Tabla para representar la interfaz del responsable de contabilidad.....	64
Tabla 3.16 Tabla para representar la interfaz del Transcriptor.....	64
Tabla 3.17 Programación y Planificación de pruebas.....	67
Tabla 3.18 Caso de Uso extendido Mantener Información Vehículos	70
Tabla 3.19 Caso de Uso extendido Registrar Uso de Vehículos	71
Tabla 3.20 Categoría de Conceptos Sistema de Control y Seguimiento	72
Tabla 3.21 Glosario de Términos	76
Tabla 3.22 Contrato Registrar Uso de Vehículos	78
Tabla 3.23 Caminos de la complejidad ciclomática	87
Tabla 3.24 Caso de Prueba Registro del uso de Vehículos.....	88
Tabla 3.25 Análisis de Resultados.....	89
Tabla 4.1 Parámetros de Medición.....	97
Tabla 4.2 Escala de Ajuste	98
Tabla 4.3 Valores de Ajuste de Complejidad	98
Tabla 4.4 Indicador de estabilidad del Software.....	101



**Marco
Contextual**

CAPÍTULO 1

MARCO CONTEXTUAL

1.1. Introducción

Cada Institución, Empresa u Organización busca mejorar el desempeño de sus actividades así como el desarrollo de las mismas, de manera que hoy en día resulta imprescindible que la información sea rápida efectiva y exacta, esto con el fin de prever buenos servicios al cliente más aun si se ofrece un producto necesario para su diario vivir.

Viendo el impacto actual de el Internet, éste se ha convertido en un medio común para la mayoría de las empresas las cuales utilizan este instrumento para ofrecer sus productos y servicios mediante la Web, en consecuencia, a causa de la evolución rápida de el Internet y de la tecnología aparecen nuevas herramientas de aplicaciones de manera que cada organización se ve en la necesidad de ingresar a este mercado y de esta manera ser más competitiva.

Tal es el caso para la realización de este proyecto que considera como área de trabajo la Empresa "CADEB" dentro del cual el área de CADE talleres maneja actualmente un sistema de uso de vehículos, en la que se ha visto que los procesos específicos de información sobre el uso de los vehículos tardan por lo cual se ha propuesto mejorar su actual proceso en el seguimiento y control del uso de vehículos, utilizando nuevas tecnologías y herramientas que permitan acceder a la información de manera rápida y disponible en cualquier momento. Haciendo que de esta forma la empresa pueda tomar decisiones oportunas y adecuadas para su beneficio.

1.2. Antecedentes

El 5 de enero de 1996 bajo las leyes de Bolivia se funda La Compañía Administradora de Empresas Bolivia S.A. (CADEB) la cual nace de la disolución de la sociedad HEMISPHERE MANAGEMENT COMPANY de Estados Unidos. CADEB mantiene las actividades de su antecesora, proveyendo postes de concreto y arrendamiento de inmuebles a las nuevas empresas de distribución eléctricas ELECTROPAZ y ELFEO.

Como parte de un proceso de tercerización en ELECTROPAZ, en septiembre de 1996 CADEB se hace cargo de la División Informática y en agosto de 1997 de la División de Talleres y Servicios. En la Gestión 1999, CADEB empieza a ofrecer otros servicios, creando un Call Center para gestionar los reclamos de los Clientes de ELECTROPAZ.

Hoy en día CADEB es un conglomerado de varias unidades de negocios con una administración centralizada. Cada unidad se encarga de producir y comercializar sus servicios y productos a través de un contacto directo con el cliente.

Las unidades de negocios o divisiones son:

División Industrial, División Informática, División Inmobiliaria, División de Talleres y Servicios, División de Call Center, División de Tecnología de la Información.

Sus principales clientes son: Electropaz, Elfeo, Edeser, Cobee, Cessa, ENTEL, Aguas del Illimani, Soboce, Gobiernos Municipales de La Paz y El Alto.

CADEB, pertenece al grupo internacional IBERDROLA de España, dentro del cual están (EDESER, ELECTROPAZ, ELFEO, CADEB S.A. y CESSA).

El área de CADEB Talleres esta a cargo del mantenimiento de los vehículos de la Empresa Electropaz así como de las otras empresas del grupo IBERDROLA de manera que los vehículos de Electropaz y de las otras empresas son revisados en CADE Talleres el cual envía un informe sobre el mantenimiento de los mismos, por otro lado también CADE Talleres maneja el sistema de control y uso de los vehículos tanto de transporte y el uso de combustible de los vehículos dentro del cual también se envía un informe resumen sobre el mantenimiento de los vehículos.

1.3. Antecedentes de Trabajos Similares

De acuerdo a los diferentes requerimientos de cada Institución ya sea pública o privada y del cual se consultaron e inspiraron algunas ideas se tienen como antecedentes de trabajos similares a los siguientes:

1. Título: Sistema de Información Vehicular

Autor: Angela Mareth Buitron Flores

Resumen: Sistema que permite almacenar, consultar y actualizar

datos e información del transporte público y particular y fiscal.

Este sistema se desarrollo para resolver el problema fundamental de la GMEA con lo se aporto con la construcción de este sistema utilizando la metodología RUP que ayudó a la Planificación y construcción del sistema para la unidad de transporte automotor del GMEA.

2. Título: Sistema de Información de Mantenimiento de Vehículos

para la Cascada S.A.

Autor: Miguel Angel Condori Aroa

Resumen: Sistema desarrollado para tener un seguimiento

detallado del mantenimiento de sus vehículos de manera que

realiza un control de los mismos proponiendo con su sistema así mejorar los tiempos de registros de los vehículos y la información del mantenimiento del mismo.

3. Título: Sistema de Información Integrado de Control y Administración

de Empresas de Transporte – Caso Trans Illimani

Autor: Gregoria Nancy Blanco Acarapi

Resumen: El sistema desarrollado es un aporte para Trans Illimani el

cual automatiza todos los procesos manejados dentro del mismo

integrando de esta manera la parte administrativa del transporte.

1.4. Análisis de la Problemática Actual

La Compañía Administradora de Empresas Bolivia S.A. (CADEB) actualmente cuenta con un Sistema integrado llamado SICAF (Sistema Comercial, Administrativo y Financiero) el cual está constituido por varios módulos que permiten a cada área cumplir con sus funciones específicas diariamente. Una de las áreas la cual es objeto de estudio para el presente Proyecto es el área de Talleres que actualmente maneja el sistema de control y seguimiento del Uso de Vehículos para las empresas del Grupo Iberdrola.

El Sistema lleva un control del Uso de los Vehículos de la empresa como de otras empresas, en la cual lleva a cabo la distribución y cargos por gasto de combustible y transporte a cada uno de los centros de costo y proyectos que hayan utilizado el vehículo.

El sistema actual de Vehículos realiza las siguientes tareas:

- Entrega en línea del presupuesto por el uso de cada vehículo a contabilidad
- Control de consumo de combustible
- Control y Seguimiento del Uso de Vehículos.
- Distribución de gastos de consumo de combustible y mantenimiento de uso de vehículos a centros de costo y proyectos.
- Generación de asientos contables basado en la distribución de gastos de combustible.

Habiendo descrito la forma de trabajo del sistema anterior se observa que la información que se maneja no es oportuna en su debido momento y además que no se esta agilizando los procesos necesarios que se necesita para llevar a cabo un buen Control y Seguimiento del uso de vehículos. Tomando en cuenta de que hoy en día la tecnología esta apuntando a sistemas vía Web lo cual hace que la empresa tiene como objetivo llegar a implementar el sistema bajo la Web, teniendo seguridad en la información, mejorando el rendimiento de sus procesos y resolviendo los problemas que presenta el actual sistema.

1.5. Planteamiento del Problema

1.5.1. Identificación del Problema:

Habiendo hecho un análisis de la problemática actual se logró identificar el problema central la cual se plantea a continuación:

¿El sistema de Control y Seguimiento del Uso de Vehículos propuesto para CADEB mejorará y agilizará los procesos que actualmente se manejan para, así proporcionar la información de manera oportuna y confiable que ayuden en la toma de decisiones de la Empresa?

1.5.2. Problemas Secundarios

Con respecto al problema central, se requiere el diseño de un sistema que abarque los requerimientos para el control y seguimiento de uso de vehículos de los cuales se detectaron los siguientes problemas Secundarios: [Ver. Anexo A]

- *Procesos lentos al realizar la Distribución:* Al generar la distribución de gastos de consumo de combustible y mantenimiento por el uso de Vehículos demora demasiado.
- *Registro de Vehículos, consumo de Combustible y Tarjetas de Uso de Vehículos:* El registro de vehículos así como el registro de consumo de combustible y de las tarjetas de uso de Vehículos se lo hace manualmente ocasionando errores en los datos.
- *Obtención de Reportes Lenta:* La generación de reportes es demasiado lenta y no oportuna para tomar decisiones.
- *La información que se envía del área de mantenimiento es manual:* Se envía información de manera global de la parte de mantenimiento al sistema actual lo que conlleva a no tener una información más a detalle de mantenimiento.
- Tardanza en la carga de datos de las tarjetas del uso de vehículos.

1.6. Objetivos

1.6.1. Objetivo General

Desarrollar e implementar el Sistema de Control y Seguimiento del Uso de Vehículos Vía Web; que permita administrar la información de manera eficiente, agilizando y mejorando los procesos que se realizan en la institución para una buena toma de decisiones.

1.6.2. Objetivos Específicos:

Los siguientes objetivos específicos planteados son: [Ver. Anexo B]

- Agilizar la generación del proceso de distribución de gastos de combustible y mantenimiento por el uso de vehículos.
- Registrar los vehículos, consumo de combustible y tarjetas de uso de vehículos de manera automática.
- Emitir reportes que contengan información detallada y continua del uso y del mantenimiento de los vehículos de acuerdo a los requerimientos.
- Integrar el sistema de manera automática con el área de mantenimiento obteniendo la información de manera más detallada sobre los vehículos que estuvieron en mantenimiento.
- Agilizar la carga de los datos de tarjetas de uso de vehículos para obtener la información de manera rápida y efectiva.

1.7. Justificación:

1.7.1. Justificación Social:

Se justifica socialmente por que beneficiará a los usuarios ya que estos están dispuestos a aceptar el cambio que facilitará el desempeño de su trabajo de manera eficaz y eficiente además de crear un sentimiento de conformidad y aceptación para beneficio de los mismos generando así la satisfacción de trabajar de manera mas transparente sin tener problemas que anteriormente perjudicaban el trabajo de los usuarios impidiendo que no cumplan con sus actividades diarias y no lograr sus objetivos.

1.7.2. Justificación Tecnológica:

Se justifica Tecnológicamente porque la empresa CADEB satisface los requerimientos del sistema ya que se adecua a la nueva tecnología para mejorar sus servicios, teniendo todos los recursos tecnológicos disponibles tanto en Hardware y Software necesarios para la realización del presente

proyecto del cual se menciona a los siguientes recursos un servidor en el cual se harán las pruebas y en la cual estará albergado el sistema Sun Aplicación Server (SUNWAPP SERVER) como servidor de páginas Web, una computadora que tiene instalado todas las herramientas necesarias para la realización del proyecto en la cual se encuentran instalados Sun Creador como IDE para el desarrollo de la parte de la interfaz, Sun Java Estudio Enterprise como IDE para el desarrollo de toda la lógica de negocios (basado en el desarrollo de los componentes que serán reutilizados), Ireport como generador de reportes las cuales son de uso libre , plataforma Solaris sobre la cual se trabajara para agilizar el avance del todo el proyecto y así poder realizar de manera eficiente el presente trabajo.

1.7.3. Justificación Económica:

Se justifica económicamente porque con la implementación de este nuevo sistema se reducirá el tiempo de acceso a la información además de que estará disponible en cualquier momento para el usuario, también apreciar el uso de software libre lo que permite ahorrar recursos económicos para la empresa sin que esta incurra en gastos y así poder beneficiarse para lograr sus metas.

1.8. Alcances

El Sistema de Control y Seguimiento del Uso de Vehículos Vía Web para la empresa CADEB comprenderá los siguientes módulos:

Distribución: Este modulo se encargara de realizar la distribución a los distintos centros de costo, proyectos y cuentas de terceros dependiendo del uso que haya tenido el vehículo.

- Distribución
 - ✓ Distribución de Gastos
 - ✓ Consultas de Distribución
 - ✓ Reportes de Distribución

Parámetros: Este módulo se encargará de realizar las distintas actividades

- Parámetros
 - ✓ Registro de Empresas Externas
 - ✓ Registro de Grupo Vehículos
 - ✓ Registro de Odómetro
 - ✓ Registro de Parámetros Generales
 - ✓ Registro de Subgrupo Vehículos
 - ✓ Registro de surtidores

- ✓ Registro de Tipos de Combustible
- ✓ Registro de Vehículos

Procesos: Este módulo se encargara de realizar los siguientes procesos

- Procesos
 - ✓ Consumo de Combustible
 - ✓ Factores de Distribución
 - ✓ Gastos por mantenimiento
 - ✓ Giro o cambio Odómetro
 - ✓ Tarjetas de uso de Vehículos

Reversiones: Este módulo registrara las nuevas gestiones mensuales de vehículos especificando el comportamiento de las mismas y realizando el cierre respectivo.

- Reversiones
 - ✓ Reversión Contable
 - ✓ Reversión Talleres

Mantenimiento: Este módulo obtendrá la información del mantenimiento de los vehículos

- Mantenimiento
 - ✓ Mano de Obra
 - ✓ Repuestos
 - ✓ Trabajos Externos

1.9. Límites

El proyecto se limita solo para el área de talleres tomando en cuenta que el sistema de control y seguimiento de uso de vehículos abarcara todo lo que es referente al uso de los vehículos y parte de mantenimiento de los mismos esto con el fin de tener la información global de ambas partes que este disponible en todo momento.

1.10. Aportes

El presente trabajo será un aporte para la empresa CADEB donde se logrará la sistematización de los procesos actuales de acuerdo a los requerimientos de la empresa, mejorando así de una manera eficiente y confiable la información que se maneja.

Otro aporte es la utilización de la metodología RUP usando a la vez el MVC(Modelo-Vista-Controlador) modelo orientado a objetos, para la realización de una buena arquitectura de sitios web, lo cual será un beneficio en la parte de desarrollo ya que de esta forma se reducirá el esfuerzo de programación que se requiere al construir un sistema.

También a través de la realización de este proyecto se pretende mostrar el uso y manejo de la metodología RUP ya que este se define mediante la combinación de flujos de trabajo. Al ser iterativo, RUP promueve que se minimice el riesgo de obtención de un mal producto (o un producto no deseado). Potenciando así la robustez del producto que se va a desarrollar.

1.11. Metodología

Se empleará la teoría del marco lógico para poder examinar con éxito todas las fases de preparación e implementación del sistema.

La metodología que se empleará para el análisis, diseño e implementación del sistema de control y seguimiento del Uso de Vehículos vía WEB será el Proceso Unificado Rational (RUP).

1.12. Técnicas y Herramientas

1.12.1. Técnicas

- Técnicas de observación, entrevistas y recolección de datos.
- Técnicas de selección bibliográfica

1.12.2. Herramientas

Para la programación del sistema se utilizará Java como lenguaje de programación y como instrumento para el desarrollo Sun Studio Creator que es el adecuado para crear aplicaciones para Internet, Studio Enterprise que es la herramienta adecuada para la parte del desarrollo de la lógica de negocios. Se utilizará como gestor de base de datos la AS400 de IBM gestor que utiliza actualmente la empresa.

Para el desarrollo de análisis y diseño se utilizará el Enterprise Architec que es una herramienta CASE que sirve para modelar el análisis y diseño de sistemas basado en objetos, servidor de aplicaciones de la SUN, sistema operativo LINUX para el desarrollo del sistema.



**Marco
Teórico**

CAPÍTULO 2

MARCO TEÓRICO

2.1. Introducción Al Proceso Unificado (RUP)

El Proceso Unificado es un proceso de desarrollo de software, un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. (Figura 2.1) y está basado en componentes de software interconectados a través de interfaces.

El proceso Unificado es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software para diferentes áreas de aplicaciones, diferentes tipos de organizaciones y diferentes tamaños de proyecto.

El proceso Unificado de Desarrollo nos proporcionará una aproximación disciplinada para asignar tareas y responsabilidades dentro de la organización con el fin de asegurar la producción de software de alta calidad que satisfaga los estándares de los usuarios finales, dentro de un horario y presupuesto predecible. [JACOB, 1999]

Los aspectos principales del Proceso Unificado de Desarrollo son los siguientes:

- ◆ Dirigido por Casos de Uso
- ◆ Centrado en la Arquitectura
- ◆ Iterativo e Incremental

Lo que hace único al proceso unificado.



Figura. 2.1. Esquema del Proceso de Desarrollo de Software [JACOB, 1999]

Es preciso mencionar que el RUP define claramente quien, como, cuando y que debe hacerse; y como su enfoque esta basado en modelos que utilizan un lenguaje bien definido, como es el Lenguaje Unificado de Modelado(UML) el cual prepara todos los esquemas de un software.

2.1.1. Dirigido por Casos de Uso:

En el Proceso Unificado los casos de uso definen el comportamiento de un sistema que son la base para el proceso de Desarrollo entero. También es la entrada fundamental cuando se especifican y verifican clases, subsistemas e interfaces, cuando se especifican casos de prueba, cuando se planifican iteraciones de desarrollo y la integración del sistema.

El conjunto de los casos de uso representa los requisitos funcionales. Un caso de uso es un fragmento de funcionalidad del sistema y todos los casos de uso constituyen el modelo de casos de uso el cual describe la funcionalidad de todo el sistema.

Los casos de uso no solo especifican los requerimientos de los sistemas si no que también juegan un papel importante en cada uno de los componentes de ingeniería: Modelo de Negocio, Requerimientos, Análisis, Diseño, Implementación y Prueba los cuales se basan en el modelo de casos de uso lo que también guiará la arquitectura del sistema y que influirá en la selección de casos de uso a utilizar. [JACOB, 1999]

2.1.2. Centrado en la Arquitectura:

La arquitectura¹ de software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios y los inversores, y se refleja en los casos de uso. Sin embargo se ve influida por muchos otros factores como ser:

- Plataforma en la que tiene que funcionar el software
 - ◆ Arquitectura hardware
 - ◆ Sistema Operativo
 - ◆ Sistema de Gestión de Base de Datos
 - ◆ Protocolos para comunicaciones en red.

¹ Conjunto de decisiones significativas acerca de la organización de un sistema software, selección de elementos estructurales a partir de los cuales se compone el sistema

- Los Bloques de construcción reutilizables de que se dispone
- Consideraciones de implantación
- Sistemas Heredados
- Requisitos no funcionales:
 - ◆ Rendimiento
 - ◆ Fiabilidad

La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado. Cada producto tiene una función y una forma, la función corresponde a los casos de uso y la forma a la arquitectura. Por un lado los casos de uso deben encajar en la arquitectura cuando se llevan a cabo. Por otro lado los casos de uso y la arquitectura deben permitir el desarrollo de todos los casos de uso requeridos. [JACOB, 1999]

La arquitectura debe diseñarse para permitir que el sistema evolucione no solo en el desarrollo inicial, sino también a lo largo de futuras generaciones, para encontrar esta forma se debe trabajar sobre los casos de uso claves del sistema.

Se puede especificar los beneficios del proceso unificado centrado en la arquitectura:

- ◆ Comprender el sistema.
- ◆ Organizar el desarrollo.
- ◆ Fomentar la reutilización.
- ◆ Hacer evolucionar el sistema.

2.1.3. Iterativo e Incremental:

El proceso de desarrollo iterativo e incremental de un producto software comercial supone un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año o más. Es en este sentido que el software no es liberado de una sola vez al final del proyecto, mas bien se divide el trabajo en partes mas pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los

incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas; esto es deben seleccionarse y ejecutarse de una forma planificada.

Cada iteración contiene todas las fases del ciclo de vida del proyecto Requerimientos, Análisis, Diseño, Implementación y Prueba (Figura 2.2.), que termina convirtiéndose en código ejecutable.



Figura. 2.2. Iterativo Incremental [EVANS, 1999]

Los beneficios de un proceso iterativo son los siguientes:

- ◆ Reduce el costo de riesgo y coste de una sola iteración.
- ◆ Se reduce el riesgo de no sacar el producto en el tiempo previsto. Mediante la identificación de riesgos en fases tempranas de desarrollo resolviéndolos al principio de la planificación.

2.2. Estructura del RUP

El proceso puede describirse en dos dimensiones, o a lo largo de dos ejes como se muestra en la Figura 2.3.

- El eje horizontal representa tiempo y muestra el aspecto dinámico del proceso, expresado en términos de ciclos, fases, iteraciones, y metas.
- El eje vertical representa el aspecto estático del proceso; como está descrito en términos de actividades, artefactos, trabajadores y flujos de trabajo.

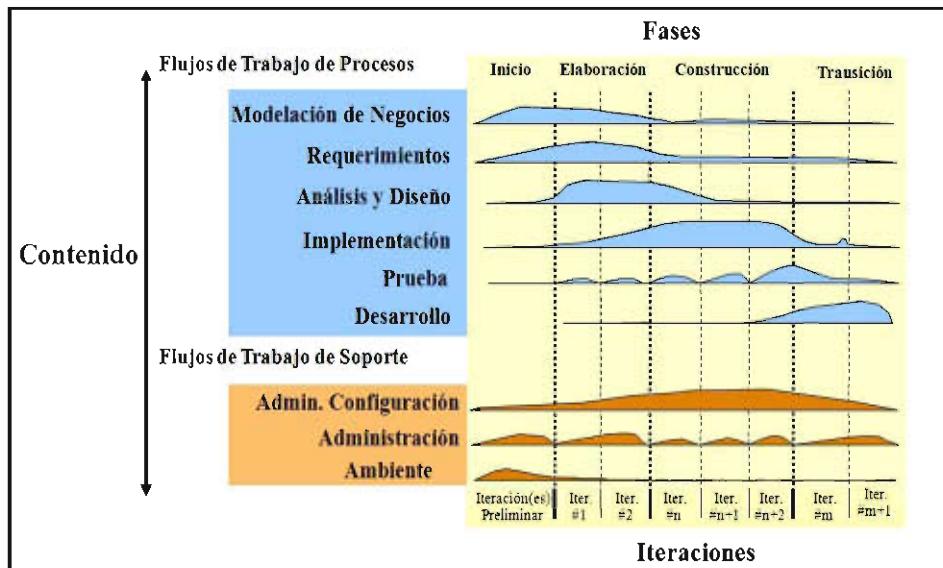


Figura. 2.3. Estructura del RUP [EVANS, 1999]

2.3. Ciclos y fases del RUP

RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo. Cada ciclo se divide en cuatro fases consecutivas:

- ✓ Inicio
- ✓ Elaboración
- ✓ Construcción
- ✓ Transición

Cada fase concluye con un hito² bien definido donde deben tomarse ciertas decisiones dentro del ciclo del desarrollo del proyecto y cada ciclo está compuesto por cinco flujos de trabajo: requisitos, análisis, diseño, implementación y pruebas. Cada fase se subdivide a su vez en iteraciones, donde cada iteración debe pasar por los cinco flujos de trabajo.

Para iniciar un ciclo, los desarrolladores necesitan todas las representaciones del producto software, es decir modelos de casos de uso, análisis, diseño, implementación y pruebas, como también una representación de la arquitectura y de los artefactos modelados por UML.

² Punto en el que han de tomarse importantes decisiones de negocio.

2.3.1. Fase de Inicio:

El objetivo principal de la fase de inicio es desarrollar el análisis del negocio hasta el punto necesario para justificar la puesta en marcha del proyecto. En esta fase se establece la planificación del proyecto y su alcance, donde es necesario:

- Determinar principales funciones del sistema.
- Esquema tentativo de la arquitectura (arquitectura candidata). Se debe tener en claro cuál es el ámbito.
- Definir los límites dentro los cuales se busque los riesgos críticos
- Plan de Desarrollo del Proyecto.
- Delimitación de estimaciones de Coste, agenda y recuperación de la inversión

Las actividades relacionadas con esta fase son:

- Planificación de las iteraciones.
- Desarrollo de los cinco flujos de trabajo.
- Análisis y gestión de Riesgos.
- Criterios de evaluación.

El desarrollo de los flujos de trabajo para la fase de inicio será respaldada por la ingeniería del producto³, la cual utiliza la ingeniería de requisitos, para recopilar, analizar, modelar, validar e implementar y realizar las pruebas necesarias haciendo una gestión de los mismos.

2.3.1.1. Planificación de las Iteraciones

La planificación de iteraciones se basa en asignaciones de tiempos, hitos principales, numero de iteraciones por fase, para lograr un plan de proyecto que esboce un mapa de actividades que cubra fechas y criterios de los objetivos principales y la correspondiente división por fases.

³ R. Pressman "Ingeniería del Software" Ed 2003, pp 171-178

2.3.1.2. Flujos de Trabajo

A continuación se describirá los cinco flujos de trabajo desde el punto de vista de la fase de inicio, con el propósito de mostrar en qué flujo de trabajo se debe hacer más énfasis. [JACOB, 1999]

2.3.1.2.1. Recopilación de Requisitos: La recopilación de requisitos en esta fase reside en el primer flujo de trabajo. Este flujo incluye identificar y detallar los casos de uso pertenecientes a esta fase. En los cuales se tiene los siguientes aspectos:

- ◆ Enumerar requisitos candidatos
- ◆ Comprender el contexto del sistema.
- ◆ Representar los requisitos funcionales pertenecientes como casos de uso
- ◆ Recoger requisitos funcionales relacionados.

2.3.1.2.2. Análisis: Los objetivos generales del flujo de trabajo de análisis en esta fase son analizar los requisitos afinarlos y estructurarlos en un modelo de objetos que sirva como primera impresión del modelo de diseño. El resultado es un modelo inicial del análisis para definir con precisión los casos de uso y el establecimiento de la arquitectura candidata

2.3.1.2.3. Diseño: En esta fase, el objetivo principal del flujo de trabajo de diseño es esbozar un modelo de diseño de la arquitectura candidata, con el objetivo de incluirlo en la descripción de la arquitectura preliminar.

2.3.1.2.4. Implementación: En este flujo de trabajo se finalizará la fase con la descripción de la arquitectura candidata en cuyo caso seguir con el flujo de trabajo de implementación no es necesario.

2.3.1.2.5. Pruebas: Durante la fase de inicio no se realiza un trabajo significativo de pruebas, ya que el prototipo desarrollado hasta este momento tiene más carácter exploratorio que operativo.

2.3.1.3. Análisis y Gestión de Riesgos

El análisis y la gestión de riesgo, es una actividad de vital importancia, para determinar los riesgos posibles en una etapa temprana del desarrollo de software. [PRESMAN, 2005]

La gestión de riesgos sigue los siguientes pasos:

- Identificación del riesgo
- Proyección del riesgo
- Refinamiento del riesgo
- Reducción, supervisión y gestión del riesgo.

a) Identificación del riesgo

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan del proyecto. Identificando los riesgos conocidos y predecibles, para evitarlos cuando sea posible y controlarlos cuando sea necesario.

b) Proyección del Riesgo

La proyección del riesgo, es denominada también estimación del riesgo, intenta medir cada riesgo con la probabilidad de que el riesgo sea real y/o las consecuencias de los problemas asociados con los riesgos identificados.

c) Refinamiento del Riesgo

Una forma de refinar el riesgo, según Gluch y Pressman es predecir el riesgo de la forma Condición-Transición-Consecuencia (CTC), donde el riesgo es sometido a un refinamiento basado en subcondiciones que ayuden a aislar los riesgos identificados y poder producir un análisis y respuesta mas sencilla.

d) Reducción, supervisión y gestión del riesgo

La estrategia sugerida por Pressman para tratar los riesgos, comprende los siguientes aspectos:

- Evitar el riesgo: se consigue desarrollando un plan de reducción de riesgo.
- Supervisar el riesgo: una vez que comienza el proyecto, comienzan las actividades de supervisión del riesgo.

- Gestionar el riesgo y planes de contingencia: asumen que los riesgos de reducción han fracasado y que el riesgo se ha convertido en una realidad, pero si se ha seguido la estrategia de reducción se tendrá copias de seguridad disponibles, información documentada y el conocimiento de cada miembro del equipo.

2.3.1.4. Criterios de evaluación

Los criterios de evaluación con respecto a la fase de inicio son los siguientes:

- Con respecto al ámbito del proyecto.
- Con respecto a resolver las ambigüedades en los requisitos.
- Con respecto a determinar una arquitectura candidata.
- Con respecto a mitigar los riesgos críticos.

2.3.2. Fase de Elaboración:

En la fase de elaboración los objetivos principales son:

- Recopilar la mayor parte de los requisitos que aun queden pendientes, formulando los requisitos funcionales como casos de uso.
- Establecer una base de la arquitectura sólida para guiar el trabajo durante las fases de construcción y transición.
- Continuar la observación y control de los riesgos críticos que aun queden, e identificar los riesgos significativos hasta el punto de que podamos estimar su impacto en el análisis del negocio y en particular en propuesta económica.
- Completar los detalles del plan del proyecto.

Las actividades a realizarse en esta etapa son las siguientes:

- Desarrollo de los cinco flujos de trabajo.
- Criterios de evaluación
- Preparación del entorno de desarrollo.

2.3.2.1. Flujos de Trabajo

A continuación describiremos los cinco flujos de trabajo desde el punto de vista de la fase de elaboración con el propósito de mostrar en que flujo de trabajo se debe hacer más énfasis.

2.3.2.1.1. Recopilación de Requisitos: En esta etapa daremos prioridad a estructurar los casos de uso. En la cual se debe considerar lo siguiente:

- Encontrar casos de uso y actores: se debe identificar la mayor parte de los casos de uso y los actores.
- Desarrollar prototipos de las interfaces: identificación de las interfaces del usuario, interesantes desde el punto de vista de la arquitectura. Por norma general no es necesario desarrollar prototipos de las interfaces de usuario durante la fase de elaboración a menos que sean únicas o se cree un propio marco de trabajo de las interfaces de usuario.
- Determinar la prioridad de los casos de uso: completando los casos de uso y trabajando en la línea base de la arquitectura, ambas influenciadas por las prioridades de los riesgos percibidos y el orden de desarrollo.
- Detallar un caso de uso: enfocándose a la construcción de los escenarios para esta fase.
- Estructurar el modelo de casos de uso: buscando similitudes, simplificaciones y oportunidades para mejorar la estructura del modelo de casos de uso.

2.3.2.1.2. Análisis: Durante la fase de inicio se comenzó hacer un borrador del modelo de análisis. Ahora se empezará a trabajar con los casos de uso que son más significativos desde el punto de vista de la arquitectura.

2.3.2.1.3. Diseño: En esta fase diseñaremos e implementemos menos del diez por ciento de los casos de uso. Este pequeño porcentaje es solo una fracción del total de casos de uso identificados desde el punto de vista arquitectónico. Esto quiere decir que diseñaremos los casos de uso, clases y subsistemas que sean arquitectónicamente significativos.

2.3.2.1.4. Implementación: Este flujo de trabajo implementa y prueba los componentes arquitectónicamente significativos a partir de los elementos de diseño más significativos.

2.3.2.1.5. Pruebas: El objetivo es asegurar que los subsistemas de todos los niveles y de todas las capas funcionen. Solo podemos probar los componentes ejecutables. Si funcionan, tendremos cierta seguridad de que otras cosas también funcionaran.

2.3.2.2. Criterios de Evaluación:

En esta etapa se deben evaluar los siguientes aspectos:

- ◆ En los requisitos
- ◆ En la definición de la línea base de la arquitectura.
- ◆ En los riesgos significativos.
- ◆ En la validez del análisis del negocio.

2.3.3. Fase de Construcción:

En la fase de construcción el objetivo principal es dejar listo un producto de software, a partir de una línea base de la arquitectura ejecutable, y trabajando a través de una serie de iteraciones e incrementos, desarrollando un producto software listo para su operación inicial en el entorno de usuario, llamada versión beta.

En esta fase se detallan los casos de uso y escenarios restantes, se modifica si es necesario la descripción de la arquitectura y se continúa con los flujos de trabajo a través de iteraciones adicionales dejando así terminados los modelos de análisis, diseño e implementación. Además de integrar los subsistemas y probarlos.

Esta fase también describe los cinco flujos de trabajo los cuales son complementados con el proceso de ingeniería el cual tiene como puntos principales: el modelo del negocio, captura de requisitos, análisis y diseño, implementación y pruebas como se ve en la Figura 2.4.

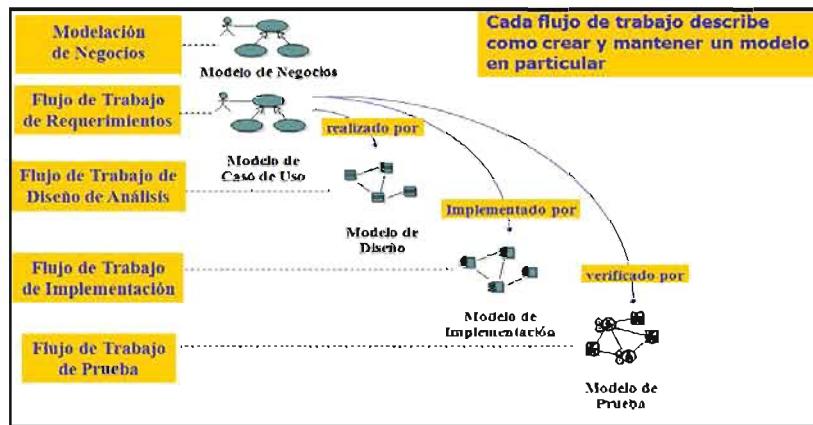


Figura 2.4. Modelos del RUP [EVANS, 1999]

2.3.3.1. Flujos de Trabajo

Describiremos los cinco flujos de trabajo desde el punto de vista de la fase de construcción. El proceso de ingeniería describe como crear y mantener un modelo del cual se puede apreciar los siguientes conceptos que son importantes para el desarrollo de la recopilación de requisitos en el cual se plasmará la idea fundamental de los requerimientos para un mejor análisis en esta fase.

2.3.3.2. Modelo del Negocio:

En primer lugar un modelo de casos de uso del negocio describe los procesos de negocio de una empresa en términos de casos de uso del negocio y actores del negocio que se corresponden con los procesos del negocio y los clientes, respectivamente. Al igual que el modelo de casos de uso para un sistema software, el modelo de casos de uso del negocio presenta un sistema desde la perspectiva de su uso, y esquematiza como proporciona valor a sus usuarios (clientes y socios). [JACOB, 1999]

2.3.3.3. Modelo del Dominio:

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema.

Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio.

2.3.3.4. Recopilación de Requisitos:

En esta etapa debemos establecer los requisitos funcionales y no funcionales del sistema realizando una recopilación completa. A partir del modelo de negocio podemos obtener el conjunto inicial de los casos de uso del sistema, y el modelo conceptual inicial del sistema, además de construir prototipos de las interfaces de usuario detallando y estructurando los casos de uso.

La captura de requisitos es el proceso de averiguar, lo que se debe hacer es decir las funciones y capacidades que el sistema debe de cumplir y mediante los casos de uso se puede tener una visión más clara de cómo será el sistema a construir.

➤ **Diagrama de casos de uso:**

El Diagrama de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas. [LARMAN, 1999] para una mejor comprensión más adelante se tiene una definición completa de diagrama de casos de uso.

➤ **Procedimiento Mediante Casos de Uso:**

Según Larman describiremos los casos de Uso, mediante una plantilla. Dado que UML no propone ningún modelo de plantilla para este proceso, hemos decidido utilizar la plantilla propuesta por Larman que a continuación presentamos la plantilla de casos de uso que utilizaremos en nuestro proceso de software. Para complementar cada plantilla de caso de uso partiremos de la especificación del contrato correspondiente a la actividad de la que proviene y que se encuentra en el diccionario de actividades. [LARMAN, 1999]

La identificación y descripción de casos de uso no es una actividad que se realiza de una sola vez, sino que a lo largo de las iteraciones del proceso de los casos de uso van refinándose, añadiéndose detalles y pueden ir surgiendo nuevos casos de uso no detectados en un principio o que surgen por cambios en el proceso de negocio. También es posible que sean identificados aspectos comunes a varios casos de uso de forma que surjan nuevos casos de uso ligados a los primeros mediante relaciones incluye, o bien puedan usarse relaciones extends para expresar los cursos alternativos u opcionales de un caso de uso se simplificara los casos de uso a lo largo de los sucesivos refinamientos buscando la claridad y la concreción, pero sin olvidar que el modelado debe ser completo y reflejar todos los requisitos del proceso de negocio correspondiente.

La descripción de casos de uso es un documento narrativo que describe la secuencia de eventos entre actores y el sistema para desarrollar un determinado proceso. Los casos de uso pueden ser expresados en dos formatos:

- Descripción de caso de uso en formato de alto nivel.
 - Descripción de caso de uso en formato expandido o de segundo nivel.
-
- **Descripción de caso de uso en formato de alto nivel.** Permiten entender a los principales procesos globales. Larman sugiere la estructura como se muestra en la Tabla 2.1

Tabla 2.1. Estructura de caso de Uso en formato de Alto Nivel [LARMAN, 1999]

CASOS DE USO:	Nombre de Caso de Uso
ACTORES:	Lista de actores.
TIPO:	1. Primario, secundario u opcional 2. Esencial o real.
DESCRIPCION:	Resumen del proceso concerniente al caso de uso.

- **La descripción de casos de uso en formato expandido o de segundo nivel.** Permiten alcanzar un conocimiento más profundo del proceso que se está tratando. Craig Larman propone el formato como se muestra en la Tabla 2.2. Correspondiente a casos de uso.

Tabla 2.2 Estructura de caso de Uso en formato expandido [LARMAN, 1999]

CASOS DE USO	Nombre del caso de uso
ACTORES	Lista de actores
PROPOSITO	Intención de caso de uso
RESUMEN	Repetición del caso de uso de alto nivel
TIPO	1. Primario secundario u opcional 2. Esencial o real.
REFERENCIA CRUZADA	Casos relacionados de uso y funciones también relacionadas de sistemas
CURSO NORMAL DE ACCIONES:	
ACCION DEL ACTOR	RESPUESTA DEL SISTEMA
Acciones numeradas de los actores	Descripción numerada de la respuesta del sistema.
CURSOS ALTERNATIVOS	
Alternativas que pueden ocurrir en el número de línea. Descripción excepción.	

Descripción de casos de uso en su totalidad

Se realizara la descripción de casos de uso en su totalidad, para poder ver como se relacionan los casos de uso entre sí y con los actores.

En este paso se pretende explicar el modelo de casos de uso como un todo. Primero se identifican actores y casos de uso para:

- Delimitar el sistema de su entorno.
- Esbozar quien y que actores interactúan con el sistema y la funcionalidad que se espera del sistema.
- Capturar y definir un glosario de términos.

Tabla 2.3. Pasos para describir casos de uso en su totalidad [LARMAN, 1999]

1	Encontrar actores
2	Encontrar casos de uso
3	Describir brevemente cada caso de uso
4	Describir el modelo de casos de uso completo

2.3.3.5. Análisis: en esta fase se consideran de nuevo las actividades de análisis de la arquitectura, analizar un caso de uso, analizar una clase y analizar un paquete lo cual al finalizar la construcción se tendrá un modelo del análisis completo del proyecto en ejecución. Para la parte del análisis se hace énfasis en conceptos basados en UML que apoyan a este flujo de trabajo los cuales nos ayudara en la construcción del sistema.

➤ **Modelo Conceptual:**

El modelo conceptual representa entidades o conceptos del mundo real concerniente al dominio del problema es el artefacto mas importante a crear durante el análisis orientado a objetos y por tanto es un a buena base para crear el modelo conceptual inicial. Debe ser realizado con poca o ninguna importancia de software y podemos considerarlo independiente del lenguaje, interfaz de usuario, base de datos y procesos o métodos de las clases. Este incluirá las clases del dominio y las relaciones entre ellas, y se representa mediante un diagrama de clases de UML.

El análisis estructurado se desarrolla mediante identificación de procesos y funciones. En cambio el análisis orientado a objetos, se lleva a cabo fundamentalmente por conceptos(objetos de problema real). Por tanto identificaremos conceptos del dominio del problema y documentaremos los resultados en un modelo conceptual.

Comenzamos a construir el modelo conceptual a partir de la lista de informaciones que hemos obtenido del diagrama de actividades. A cada información contenida en el diccionario de informaciones se le asociara una clase siempre y cuando esa información vaya a ser tratada por el sistema software. A partir de la especificación de una información extraeremos las propiedades de la correspondiente clase, atributos, asociaciones con otras clases, responsabilidades y restricciones.

Este modelo conceptual creado puede refinarse sucesivamente hasta conseguir el nivel de corrección deseado. En este refinamiento podemos encontrar nuevos conceptos que habrá que incluir también en el diccionario de informaciones para encontrar esos nuevos conceptos podemos utilizar el método de identificación de nombres y/o la lista de categorías.

Tabla 2.4. Pasos para construir un modelo conceptual [LARMAN, 1999]

1	Liste los conceptos idóneos utilizando la lista de categorías de conceptos la frase nominal relacionada con los requerimientos en cuestión.
2	Dibújelos en un modelo conceptual.
3	Incorpore las asociaciones necesarias para registrar las relaciones para las cuales debe reservar un espacio en memoria.
4	Agregue los atributos necesarios para cumplir con las necesidades de información.

La asociación es una relación entre dos conceptos que indica alguna conexión significativa entre ellos es necesario identificar asociaciones para satisfacer los requerimientos de información de los casos de uso en cuestión y los que contribuyen a entender el modelo conceptual.

Registro de términos en glosario

El glosario es un documento simple en el cual se definen términos que requieren explicación para mejorar la comunicación y aminorar el riesgo de malos entendidos.

Cuando se restringe términos en el glosario, se debe contemplar los puntos como se muestra en la Tabla 2.5.

Tabla 2.5. Estructura de términos en un glosario [LARMAN, 1999]

TERMINO	CATEGORIA	COMENTARIO
Nombre del termino	Caso de Uso/Atributo/Tipo	Breve explicación del termino

UML nos ofrece una notación que nos permitirá modelar y construir los artefactos del sistema. Considerando el tipo de diagrama que se utilice en la parte de Análisis, Diseño e Implementación a continuación se describirá de forma general los diagramas que se apliquen en cada etapa de esta fase, más adelante se describe cada uno de los diagramas de UML con más profundidad.

Comportamiento mediante diagramas de secuencia

Los diagramas de secuencia nos permiten mostrar gráficamente los eventos que fluyen de los actores al sistema con el propósito de conocer el comportamiento del sistema, y la descripción de lo que hace el sistema sin explicar como lo hace.

Los casos de uso nos muestran como los actores interactúan con el sistema de software que es lo que deseamos crear. El diagrama de secuencias de un sistema es una representación que muestra, la determinada instancia de un caso de uso. Por tanto existe una estrecha relación entre los diagramas de casos de uso y los diagramas de secuencia ya que los primeros generan a los segundos.

Diagramas de Secuencia

Este diagrama (llamado también diagrama de interacción) muestra las interacciones entre un conjunto de objetos (clases, actores), ordenadas según el tiempo en que tienen lugar. Es decir, muestra el orden de las llamadas en el sistema.

Se utiliza un diagrama para cada llamada a representar. Es imposible representar en un solo diagrama la secuencia de todas las llamadas posibles del sistema, es por ello que se escoge un punto de partida.

El diagrama de secuencia con los objetos que forman parte de la secuencia, estos se sitúan en la parte superior de la pantalla, normalmente a la izquierda se sitúa el que inicia la acción. De estos objetos sale una línea que indica su vida en el sistema. Esta línea se convierte en una línea gruesa cuando representa que el objeto tiene el foco del sistema, es decir cuando el está activo.

Este diagrama será útil para observar la vida de los objetos en el sistema, identificar llamadas a realizar o posibles errores del modelo estático, que imposibiliten el flujo de información o de llamadas entre los componentes del sistema.

Comportamiento del sistema de contratos

Los contratos nos ayudaran a definir el comportamiento de un sistema; describe el efecto que tienen las operaciones sobre comportamiento del sistema a partir de cómo cambia el estado cuando se llama a una operación.

En términos generales un contrato es un documento que muestra lo que una operación se propone lograr. Se expresa a partir de cambios de estado de las precondiciones y poscondiciones. A continuación se muestra el esquema de un contrato (véase la Tabla 2.6.)

Tabla 2.6. Descripción de un contrato [LARMAN, 1999]

CONTRATO	
NOMBRE:	Nombre de la operación y parámetros.
RESPONSABILIDADES:	Descripción informal de las responsabilidades que debe cumplir la operación.
TIPO:	Nombre del tipo(concepto, clase de software, interfaz)
REFERENCIAS CRUZADAS:	Números de referencia de la funciones del sistema.
NOTAS:	Notas del diseño, algoritmos o información afín.
EXCEPCIONES:	Casos excepcionales.
SALIDA:	No salida de interfaz del usuario; por ejemplo, mensajes registros que se envían fuera del sistema
PRECONDICIONES:	Suposiciones acerca del estado del sistema antes de ejecutar la operación.
POSCONDICIONES:	El estado del sistema después de la operación

2.3.3.6. Diseño: en esta fase se diseña e implementa el noventa por ciento restante de los casos de uso, aquellos que no fueron utilizados para desarrollar la línea base de la arquitectura.

Para esta etapa se define para cada caso de uso una colaboración, tanto en su parte estructural como dinámica, que especifique como el sistema realizara el caso de uso a través de una interacción entre un grupo de objetos. Esto implica crear escenarios que describan el

comportamiento asociado a cada caso de uso, que serán refinados sucesivamente, y trabajar con el modelo conceptual hasta llegar a obtener las clases del sistema y elaborar el esquema conceptual de datos.

Diagramas de colaboración

Este diagrama muestra la interacción entre varios objetos y los enlaces que existe entre ellos, representa las interacciones entre los objetos organizados alrededor de los objetos y sus vinculaciones. Los diagramas de secuencias y los diagramas de colaboraciones expresan información similar, pero en forma diferente.

Diagramas de Clases

En el diagrama de clases definiremos las características de cada una de las clases y relaciones de dependencia y generalización. Es decir es donde daremos rienda suelta a nuestros conocimientos de diseño orientado a objetos definiendo clases e implementando las típicas relaciones de herencia y agregación.

Este diagrama sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, herencia y contenido. Los diagramas de clase se utilizan cuando necesitamos realizar un análisis de dominio, con el objetivo de conocer las entidades principales en el dominio para luego convertirse en clases.

Los componentes de un diagrama de clases son:

Clase: es la unidad básica que encapsula toda la información de un objeto. A través de ella podemos modelar el objeto de estudio.

Atributo: un atributo representa la propiedad de la clase los atributos pueden representarse mostrando solo su nombre y su tipo e incluso su valor por defecto.

2.3.3.7. Implementación: este flujo de trabajo implementa y lleva a cabo las pruebas de unidad de todos los componentes, trabajando principalmente a partir del modelo de diseño.

Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, solo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

Normalmente los diagramas de componentes se utilizan para modelar código fuente, versiones ejecutables, bases de datos físicas, entre otros.

El diagrama de componentes se usa para modelar la estructura del software, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables. En el diagrama de componentes se modela componentes del sistema, a veces agrupados por paquetes, y las dependencias que existen entre componentes(y paquetes de componentes)

Diagrama de despliegue: el diagrama de despliegue muestra la arquitectura física del hardware y el software del sistema. Se pueden mostrar las computadoras y los dispositivos, junto a las conexiones que tienen unos con otros; también se puede mostrar el tipo de conexión.

2.3.3.8. Pruebas: en esta última etapa se comprobara que el sistema cumple con los requisitos establecidos y que se comporta de acuerdo a las especificaciones indicadas.

Para ello se utilizaran los casos de uso del sistema, de tal manera que para cada uno de ellos se comprobara que el sistema se comporta de la forma esperada. También habrá que validar los requisitos no funcionales que hubieren sido establecidos. Debemos decir que entendemos que cada implementador debe ser responsable de testear su propio trabajo, intentando siempre alcanzar los objetivos de calidad que se proponen en cualquier proyecto.

2.3.4 Fase de Transición:

En esta fase la actividad es baja en los cinco flujos de trabajo ya que todo el trabajo se realizó en las fases anteriores, el nivel de actividad es bajo ya que el trabajo se concentra en implementar el producto en su entorno de operación y corregir los problemas encontrados en el entorno del usuario.

Las actividades llevadas a cabo en esta fase son las siguientes:

- Preparar la versión beta (pruebas de aceptación) a partir de la versión con capacidad operativa inicial producida durante la fase de transición.
- Instalación de la versión beta.
- Instalar la versión en los lugares elegidos
- Actuar a partir de la información recogida en las instalaciones de pruebas
- Adaptar el producto corregido a las circunstancias de los usuarios.
- Completar artefactos del proyecto.
- Determinar cuando se acaba el proyecto.

2.4. UML (Lenguaje Unificado de Modelado)

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

2.4.1. Procesos Basados en UML

Desde la consolidación de UML como lenguaje estándar para el modelado se ha definido un buen número de procesos para el desarrollo de aplicaciones orientadas a objetos que utilizan este lenguaje como medio de expresión de los diferentes modelos que se crean durante el ciclo de vida. Las características principales deseables en cualquier proceso software basado en UML son:

- Un proceso debe ser iterativo e incremental, y debe centrarse en los aspectos críticos en las primeras iteraciones para minimizar riesgos.
- Es necesario un enfoque industrial para la producción de software:
“capacidad de producir productos de alta calidad a bajo coste”.
- Debe estar guiado por los requisitos (Casos de uso). Los requisitos cambian a lo largo del desarrollo del proyecto y el proceso debe estar preparado para identificar nuevos requisitos a lo largo de todo el ciclo de vida, ya que es muy difícil que puedan capturarse todos los requisitos antes de empezar la implementación.
- Debe utilizar arquitectura basadas en componentes.
- Modelado software visual, esto facilita la gestión de los modelos, pues ayuda a mantener la consistencia entre los elementos del sistema y ayuda a mejorar la habilidad de un equipo de desarrollo para manejar la complejidad del software.
- Permite verificar la calidad del software. Es importante evaluar continuamente la calidad de un sistema con respecto a su funcionalidad, fiabilidad, rendimiento de la aplicación y rendimiento del sistema.
- Debe existir un control de cambios del software. la ausencia de un control de cambios hace que el proceso degenera rápidamente en un caos. Si se hace un control de cambios se solucionan parte de las dificultades principales del desarrollo de software, como la

comunicación entre equipos de desarrollo, la consistencia, la interferencia entre miembros de un equipo que trabajan en paralelo.

En todos los procesos basados en UML el concepto de caso de uso juega un papel primordial, ya que se emplea para definir los requisitos funcionales del sistema y en torno a ellos se articulan todas las etapas del proceso.

UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

2.4.2. Modelos y Diagramas:

- **Modelo:** Un modelo captura una vista de un sistema del mundo real. Es una abstracción de dicho sistema, considerando un cierto propósito. Así el modelo describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle.
- **Diagrama:** Un diagrama es una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos.

Un proceso de desarrollo de software debe ofrecer un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés.

El código fuente del sistema es el modelo más detallado del sistema (y además es ejecutable). Sin embargo, se requieren otros modelos.



Figura 2.5. Modelo Físico [WIKIPEDIA, 2005]

Cada modelo es completo desde su punto de vista del sistema, sin embargo, existen relaciones de trazabilidad entre los diferentes modelos.

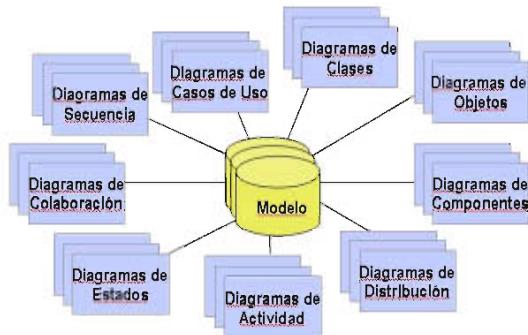


Figura 2.6. Diagramas que utiliza el modelo UML [WIKIPEDIA, 2005]

2.4.3. Diagramas que utiliza el UML

En UML cada diagrama usa la anotación pertinente y la suma de estos diagramas crean las diferentes vistas.

2.4.4. Elementos Comunes en todos los Diagramas

- Notas

Una nota sirve para añadir cualquier tipo de comentario a un diagrama o a un elemento de un diagrama. Es un modo de indicar información en un formato libre, cuando la notación del diagrama en cuestión no nos permite expresar dicha información de manera adecuada. Una nota se representa como un rectángulo con una esquina doblada con texto en su interior. Puede aparecer en un diagrama tanto solo como unido a un elemento por medio de una línea discontinua. Puede contener restricciones, comentarios, el cuerpo de un procedimiento, etc. Véase Figura 2.7.

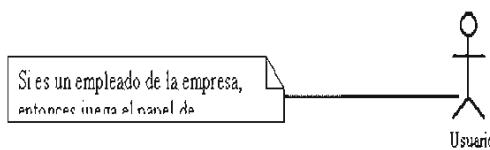


Figura 2.7. Ejemplo de Nota [WIKIPEDIA, 2005]

- Dependencia

La relación de dependencia entre dos elementos de un diagrama significa que un cambio en el elemento destino puede implicar un cambio en el elemento origen (por tanto, si cambia el elemento destino habría que revisar el elemento origen). Una dependencia se representa por medio de una línea de trazo discontinuo entre los dos elementos con una flecha en su extremo. El elemento dependiente es el origen de la flecha y el elemento del que depende es el destino (junto a él está la flecha). Véase Figura 2.8.

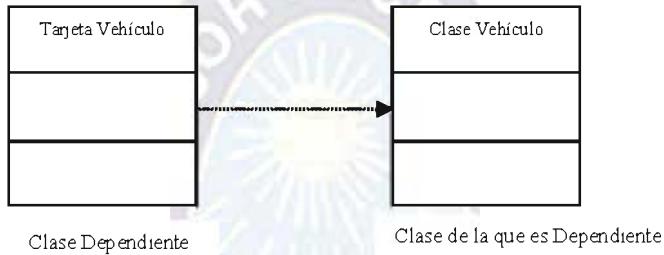


Figura 2.8. Ejemplo de Dependencia [Elaboración Propia]

Se dispone de dos tipos diferentes de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica. Los diagramas estáticos son:

- **Diagramas de Estructura Estática**

Con el nombre de Diagrama de Estructura Estática se engloba tanto al modelo conceptual de la fase de Análisis como al diagrama de Clases de la fase de Diseño, ambos son distintos conceptualmente, mientras el primero modela elementos del dominio el segundo presenta los elementos de la solución software.

Sin embargo, ambos comparten la misma notación para los elementos que los forman (clases y objetos) y las relaciones que existen entre los mismos (asociaciones).

- a) Clase

Es la unidad básica que encapsula toda la información de un objeto (un objeto es una instancia de una clase). A través de ella podemos modular el entorno en estudio (por ejemplo: un libro, un auto, una cuenta corriente, etc.) [SAL03]. Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones. Véase figura 2.9

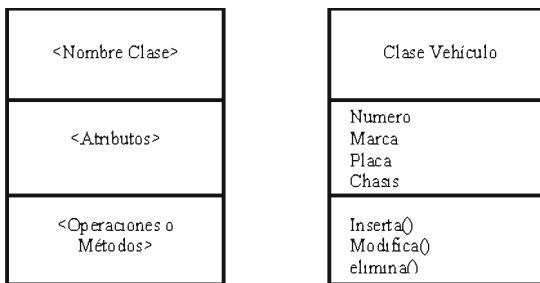


Figura 2.9. Representación de una Clase [Elaboración Propia]

Una clase puede representarse de forma esquemática, con los atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase.

b) Objeto

Un objeto se representa de la misma forma que una clase. En el compartimento superior aparecen el nombre del objeto junto con el nombre de la clase subrayados, según la siguiente sintaxis: nombre_del_objeto: nombre_de_la_clase Puede representarse un objeto sin un nombre específico, entonces sólo aparece el nombre de la clase. Véase Figura 2.10



Figura 2.10. Ejemplo de Objeto [Elaboración Propia]

c) Asociación

La asociación expresa una conexión bidireccional entre objetos, es una abstracción de la relación existente en los enlaces entre objetos, un ejemplo de asociación se lo puede ver en la Figura 2.11. Las asociaciones entre dos clases se representan mediante una línea que las une. La línea puede tener una serie de elementos gráficos que expresan características particulares de la asociación. A continuación se verán los más importantes de entre dichos elementos gráficos.

d) Multiplicidad

La multiplicidad es una restricción que se pone a una asociación, que limita el número de instancias de una clase que pueden tener esa asociación con una instancia de la otra clase.

[SAL03]

Puede expresarse de las siguientes formas:

- Con un número fijo: 1.
- Con un intervalo de valores: 2..5.
- Con un rango en el cual uno de los extremos es un asterisco. Significa que es un intervalo abierto. Por ejemplo, 2..* significa 2 o más.
- Con una combinación de elementos como los anteriores separados por comas: 1, 3..5, 7, 15..*.
- Con un asterisco: *. En este caso indica que puede tomar cualquier valor (cero o más).

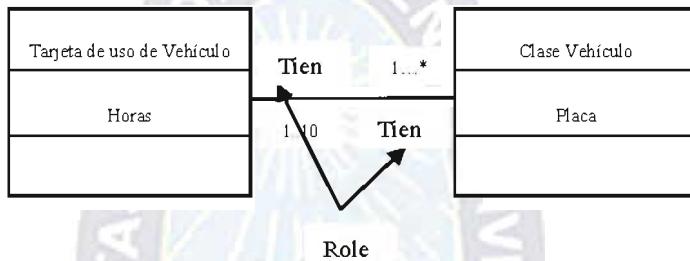


Figura 2.11. Ejemplo de Asociación [Elaboración Propia]

e) Roles

Para indicar el papel que juega una clase en una asociación se puede especificar un nombre de rol. Se representa en el extremo de la asociación junto a la clase que desempeña dicho rol.

f) Agregación

Es un tipo especial de asociación que representa una relación estructural entre el todo y sus partes. El símbolo de agregación es un diamante colocado en el extremo en el que está la clase que representa el “todo”.

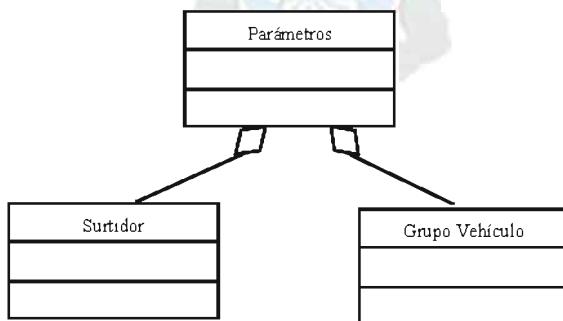


Figura 2.12. Ejemplo de Agregación [Elaboración Propia]

g) Clases Asociación

Cuando una asociación tiene propiedades propias se representa como una clase unida a la línea de la asociación por medio de una línea a trazos. Tanto la línea como el rectángulo de clase representan el mismo elemento conceptual: la asociación. Por tanto ambos tienen el mismo nombre, el de la asociación. Cuando la clase asociación sólo tiene atributos el nombre suele ponerse sobre la linea. Por el contrario, cuando la clase asociación tiene alguna operación o asociación propia, entonces se pone el nombre en la clase asociación y se puede quitar de la línea.

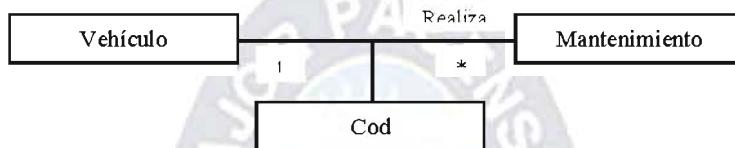


Figura 2.13. Ejemplo de Multiplicidad en asociación [Elaboración Propia]

h) Asociaciones N-Arias

En el caso de una asociación en la que participan más de dos clases, las clases se unen con una línea a un diamante central. Si se muestra multiplicidad en un rol, representa el número potencial de tuplas de instancias en la asociación cuando el resto de los N-1 valores están fijos.

i) Navegabilidad

En un extremo de una asociación se puede indicar la navegabilidad mediante una flecha. Significa que es posible "navegar" desde el objeto de la clase origen hasta el objeto de la clase destino. Se trata de un concepto de diseño, que indica que un objeto de la clase origen conoce al (los) objeto(s) de la clase destino, y por tanto puede llamar a alguna de sus operaciones.

j) Herencia

La relación de herencia se representa mediante un triángulo en el extremo de la relación que corresponde a la clase más general o clase "padre". Si se tiene una relación de herencia con varias clases subordinadas, pero en un diagrama concreto no se quieren poner todas, esto se representa mediante puntos suspensivos. Véase figura 2.14

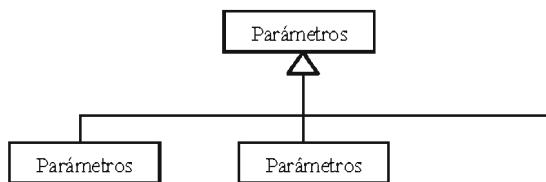


Figura 2.14. Ejemplo herencia [Elaboración Propia]

Diagrama de Casos de Uso

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. En la Figura 2.15 se muestra un ejemplo de Diagrama de Casos de Uso para el registro de los vehículos

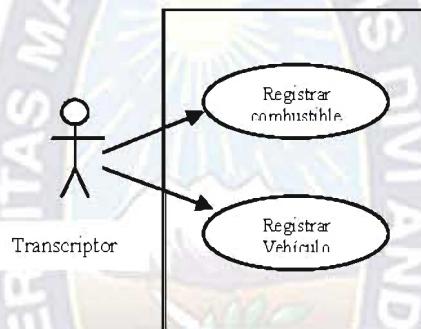


Figura 2.15. Ejemplo de Caso de Uso [Elaboración Propia]

Elementos:

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

- **Actores**

Un actor es algo con comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema.. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores.(otros sistemas, etc).

- **Casos de Uso**

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea

específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

- **Relaciones entre Casos de Uso**

Entre dos casos de uso puede haber las siguientes relaciones:

Extiende: cuando un caso de uso especializa a otro extendiendo su funcionalidad.

Usa: cuando un caso de uso utiliza a otro.

Diagrama de clases

Los diagramas de clases muestran las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.

Diagrama de objetos

El diagrama de objetos es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas. Se da una visión de casos reales.

➤ **Diagramas de estructura Dinámica**

Los diagramas de estructura dinámica nos permiten ver el funcionamiento del sistema, a través de eventos, transiciones, actividades que cada diagrama presenta.

Diagramas de Interacción

Modelan el comportamiento dinámico del sistema; el flujo de control en una operación describe la interacción entre objetos; los objetos interactúan a través de mensajes para cumplir ciertas tareas. En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información pero cada uno enfatizado un aspecto particular: diagramas de secuencia y diagramas de colaboración.

- **Diagrama de Secuencia**

Modela interacciones en un grafico de dos dimensiones. Un diagrama de secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. Véase figura 2.16

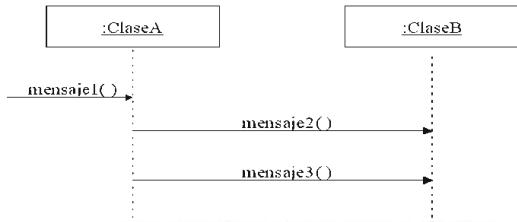


Figura 2.16. Ejemplo de Diagrama de secuencia [SCHMULLER, 2001]

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

- **Diagrama de Colaboración**

Modelan el comportamiento dinámico del sistema; el flujo de control en una operación.

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere).

A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

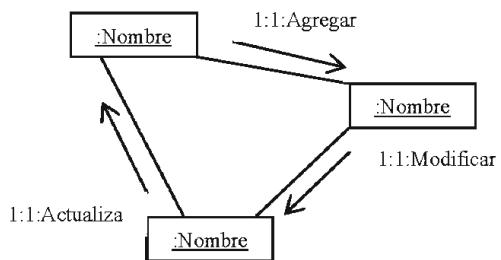


Figura 2.17. Ejemplo de Diagrama de colaboración [SCHMULLER, 2001]

Diagramas de comportamiento

- **Diagramas de estados**

Un diagrama de estados muestra la secuencia de estado por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando que eventos hacen que se pase de un estado a otro y cuales son las respuestas y acciones que genera.

En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos, un estado se representa como una caja redondeada con el nombre del estado en su interior.

Una transición se representa como una flecha desde el estado origen al estado destino.

La caja de un estado puede tener 1 o 2 compartimentos. En el primer compartimento aparece el nombre del estado. El segundo compartimento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas.

Una acción de entrada aparece en la forma entrada/acción asociada donde acción asociada es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta.

Una acción de salida aparece en la forma salida/acción asociada. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta.

Una acción interna es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma nombre de evento/acción asociada.



Figura 2.18. Ejemplo de Diagrama de estados [SCHMULLER, 2001]

- **Diagrama de Actividades**

El diagrama de actividades sirve para representar el sistema desde otra perspectiva, y de este modo complementa a los anteriores diagramas vistos. Gráficamente un diagrama de actividades será un conjunto de arcos y nodos. Desde un punto de vista conceptual, el diagrama de actividades muestra cómo fluye el control de unas clases a otras con la finalidad de culminar

con un flujo de control total que se corresponde con la consecución de un proceso más complejo. Por este motivo, en un diagrama de actividades aparecerán acciones y actividades correspondientes a distintas clases.

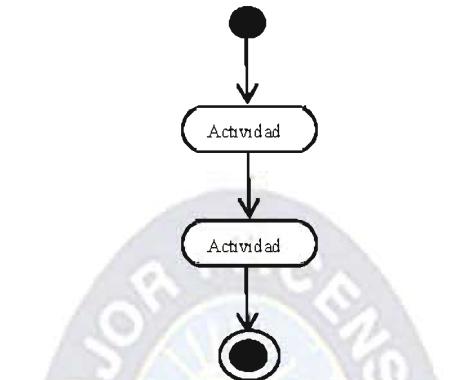


Figura 2.19. Ejemplo de Diagrama de actividades [SCHMULLER, 2001]

Diagramas de implementación

- **Diagrama de componentes:** Muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.

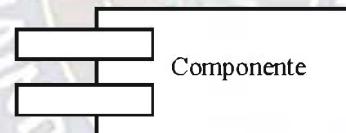


Figura 2.20. Ejemplo de Diagrama de componentes [SCHMULLER, 2001]

- **Diagrama de despliegue:** Muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice.

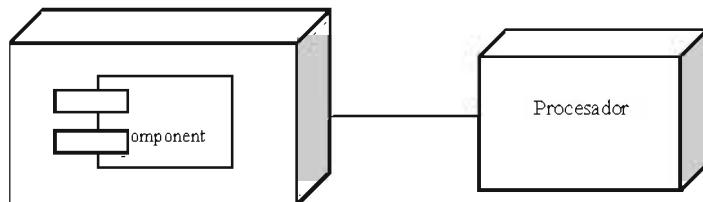


Figura 2.21. Ejemplo de Diagrama de despliegue [SCHMULLER, 2001]

2.5. MVC(Modelo-Vista-Controlador)

MVC es el patrón de diseño arquitectural recomendado para aplicaciones interactivas Java.

MVC separa los conceptos de diseño, y por lo tanto decrementa la duplicación de código, el centralizamiento del control hace que la aplicación sea más extensible.

MVC es el patrón de diseño arquitectural para la capa de presentación, este diseño de modelo permite separar el contenido web (html) de la lógica a usar, el uso de esta filosofía de diseño hace más fácil para que los diseñadores y los programadores se centren en sus campos de especialización respectivos.

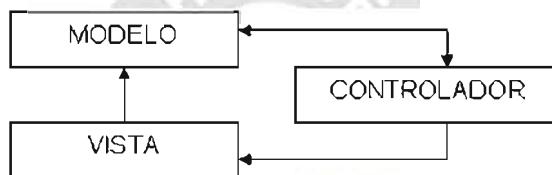


Figura 2.22. Arquitectura del MVC [Fowler, 1999]

El sistema de arquitectura MVC (Model-View-Controller) es un sistema destinado a separar una aplicación o una parte de la aplicación en 3 partes:

- ◆ **Modelo:** el modelo representa los datos de la empresa y las reglas de negocio a los cuales gobierne el acceso y las actualizaciones de estos datos. El modelo sirve a menudo como aproximación del software a un proceso del mundo real. Entonces modelo es el acceso a la base de datos.
- ◆ **Vista:** la vista rinde el contenido de un modelo. Tiene acceso a datos de la empresa a través del modelo y especifica como esos datos deben ser presentados es la responsabilidad de la vista mantener consistencia en su presentación cuando el modelo cambia. Esto puede lograrse usando un modelo de empuje, donde la visión se coloca con el modelo para las notificaciones del cambio, donde esta es responsable de llamar al modelo cuando necesita recuperar los datos más actuales. Entonces visión es la presentación visual de los datos.
- ◆ **Controlador:** el regulador traduce interacciones con la visión a las acciones que se realizaran por el modelo. De acuerdo con las interacciones del usuario y el resultado de las acciones del modelo, el regulador responde seleccionando una visión apropiada. Entonces el controlador se encarga de enlazar el acceso a datos con la representación de los mismos

2.6. Modelo de tres Capas

La idea del desarrollo en tres capas consiste en utilizar un método de desarrollo para nuestros sistemas que nos permita separar en distintas capas. Las capas recomendadas son: la interfaz de usuario, las reglas de negocio y la base de Datos, la siguiente figura [Ver Figura 2.23.] nos grafica el concepto del funcionamiento de esta arquitectura tres capas.

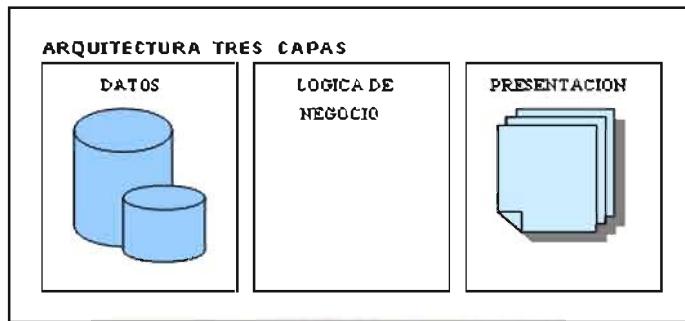


Figura. 2.23 Arquitectura tres capas [SUN, 2006]

Como se puede observar, la arquitectura de 3 capas permite hacer que tanto la interfaz de usuario, las reglas de negocios y el motor de datos se conviertan en entidades separadas unas de otras, lo importante es mantener bien definidas las interfaces que cada una de estas expongan para comunicarse con la otra.

Datos:

La capa de datos tiene como misión la administración de la información que maneja el sistema. Esto incluye el almacenamiento, la actualización y la consulta de todos los datos contenidos en el sistema. En la práctica, esta capa es esencialmente un servidor de bases de datos. Gracias a las herramientas propias y de terceras partes para la abstracción de base de datos, las aplicaciones pueden utilizar distintas bases de datos. También es posible agregar soporte para una nueva base de datos en un periodo de tiempo relativamente corto. La capa de datos puede estar en el mismo servidor que las de lógica de negocio y datos pueda estar en el mismo servidor que las de lógica de negocio y presentación, o en un servidor independiente.

Lógica de Negocios:

El comportamiento de la aplicación es definido por los componentes que modelan la lógica de negocio. Estos componentes reciben las acciones a realizar a través de la capa de presentación, y llevan a cabo las tareas necesarias utilizando la capa de datos para manipular la información del sistema. Tener la lógica de negocio separada del resto del sistema también permite una integración más sencilla y eficaz con sistemas externos, ya que la misma lógica utilizada por la capa de presentación puede ser accedida desde procesos automáticos que intercambian información con los mismos. En nuestros sistemas, la capa de lógica de negocios es construida utilizando tecnologías de componentes.

Presentación:

La capa de presentación representa la parte del sistema con la que interactúa el usuario. En una aplicación Web, generalmente la capa de presentación se divide en dos: el lado servidor y el lado cliente. En el lado servidor ocurre toda la interacción con la lógica de negocio, y es también donde se genera la interfaz del usuario. En el lado cliente se presenta la interfaz generada en el servidor al usuario, de forma tal que este pueda trabajar con ella. Los datos o acciones reunidas por el cliente son luego enviadas de vuelta al servidor para su procesamiento.

2.6.1. Ventajas del modelo tres capas

A lo largo del ciclo de vida de una aplicación, el uso del modelo de 3 capas brinda beneficios tales como:

- Reusabilidad
- Flexibilidad
- Simplicidad de administración y mantenimiento
- Escalabilidad

Los componentes y servicios creados según este modelo pueden compartirse y reutilizarse; se pueden colocar todos en el mismo equipo o, de ser necesario, distribuirse a través de una red. De esta manera, proyectos de gran envergadura pueden dividirse en pequeños proyectos más simples y manejables, que se pueden implementar en forma progresiva, agregando nuevos servicios según la medida de crecimiento de la organización.

2.7. Seguridad

El tema de seguridad de la información cobra valor en diferentes aplicaciones en la actualidad. Las organizaciones consideran que la información representa un bien más de sus activos y, en muchos casos, prioritario sobre los restantes.

Las aplicaciones Web permiten el acceso de usuarios a recursos centrales como puede ser el servidor Web, a través de este, a otros como servidores de base de datos.

Comprender e implementar las medidas de seguridad adecuadas permite:

- Proteger los recursos propios contra accesos no autorizados.
- Restringir los niveles de acceso por usuario o por rol.
- Establecer la integridad y confidencialidad de los datos, ofreciendo un entorno seguro en el que los usuarios se encuentren cómodos trabajando con la aplicación.
- Establecer control sobre como la aplicación obtiene acceso a recursos restringidos.
- Garantizar que el código de la aplicación se ejecuta de la forma esperada.

Para hacer frente a las amenazas a la seguridad del sistema se define una serie de servicios para proteger los sistemas de proceso de datos y de transferencia de información de una organización. Estos servicios hacen uso de uno o varios niveles de seguridad utilizando herramientas propias. Una clasificación útil de los servicios es:

Confidencialidad Se entiende por confidencialidad el servicio de seguridad, o condición, que asegura que la información no pueda estar disponible o ser descubierta por o para personas, entidades o procesos no autorizados.

Integridad Se entiende por integridad el servicio de seguridad que garantiza que la información es modificada, incluyendo su creación y borrado, solo por el personal autorizado. El concepto de integridad significa que el sistema no debe modificar o corromper la información que almacene, o permitir que alguien no autorizado lo haga.

Disponibilidad significa que el sistema, tanto hardware como software, se mantienen funcionando eficientemente y que es capaz de recuperarse rápidamente en caso de fallo.

2.8. Métricas de Calidad

El estándar ISO 9126 es un estándar internacional para la evaluación del Software. Este estándar permitirá medir el grado de calidad del sistema a construirse se empleará algunos puntos clave de calidad como ser:

a) Confiabilidad: La confiabilidad implica un comportamiento aceptable frente a situaciones inesperadas, en un caso ideal diremos que un software es confiable si cumple con los requerimientos especificados y no ocasiona graves problemas frente a situaciones imprevistas. La idea detrás de la confiabilidad es que quizás aunque en ciertas ocasiones el programa reaccione de forma inesperada, el usuario se encuentre cómodo usándolo ya que estos fallos no traen problemas graves.

En términos estadísticos la confiabilidad se define como: "la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico" [PRESSMAN, 2005].

La confiabilidad $R(t)$ de un componente (o subsistema) en determinado medio durante un periodo t se define como la probabilidad de que su tiempo para fallar excede a t (o sea para trabajar satisfactoriamente en el periodo t)", es decir:

$$R(t)=P[T>t]=1-Y(t)$$

Donde:

$R(t)$: Confiabilidad de un componente o subsistema en el tiempo t .

$Y(t)$: Probabilidad de falla del componente o subsistema en el tiempo t .

T : El tiempo para fallar o la duración del tiempo de trabajo sin falla.

Considerando que el tiempo T para fallar es una variable aleatoria exponencial entonces se tiene:

$$R(t)=1-(1-\lambda e^{-\lambda t})$$

$$R(t)=e^{-\lambda t}$$

Donde:

λ : es la tasa constante de fallo

t : periodo de operación en tiempo.

Una vez obtenido la función de confiabilidad, se irá analizando la confiabilidad de cada uno de los módulos definidos en el sistema.

El siguiente paso es analizar la confiabilidad del sistema. Para esto consideramos dos situaciones: en un caso se considera que el sistema entero falla, si falla cualquiera de sus componentes, en otro caso se considera que el sistema falla si y solo si falla todo. La primera situación modela una conexión de subsistemas en serie y la segunda modela una conexión en paralelo de los subsistemas o componentes.

b) Funcionalidad:

La funcionalidad se valora evaluando el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global.

La funcionalidad se valora mediante una medida llamada punto de función, donde primero se debe determinar cinco características de dominio de información: [PRESSMAN, 2005]

- Número de entradas de usuario.
- Número de salidas de usuario
- Número de peticiones de usuario
- Número de archivos
- Número de interfaces externas

Para calcular puntos de función(PF), se utiliza la siguiente relación:

$$PF = \text{Cuenta-total} * (X + Y * \sum F_i) \quad (1)$$

Donde:

PF: Medida de funcionalidad

Cuenta total: es la suma de las entradas obtenidas de la tabla.

X: Nivel de confiabilidad del sistema igual a 0.65.

Y: Nivel de significación de error igual a 0.01.

$\sum F_i$: Valores de ajuste a la complejidad ($i=1$ a 14).

Una vez detalladas cada una de las características, ahora estas cinco características la ponemos en una tabla de evaluación que se muestra a continuación, donde se toma un factor para multiplicar (alto, medio, bajo).

Tabla 2.7. Parámetros de Medición [Fuente: Elaboración Propia]

Parámetros de Medición	Cuenta	Factor de Ponderación (Medio)	Total
Número de entradas de usuario	10	4	40
Número de salidas de usuario	16	5	80
Número de peticiones de usuario	10	4	40
Número de archivos	16	10	160
Número de interfaces externas	4	7	28
Cuenta Total:			348

Los valore de ajuste de complejidad se obtiene respondiendo las preguntas destacables que se muestran en la Tabla 2.7

Se obtiene la suma del punto función y se reemplaza en la formula de punto función.

c) Mantenimiento de software

Existen varias definiciones sobre lo que el mantenimiento de software significa, a continuación se muestran algunas de ellas [SOM95].

- Modificar un programa después de que ha sido puesto en uso.
- El mantenimiento del software es la modificación de un producto software después de su entrega al cliente o usuario para corregir defectos, mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno.

Los sistemas necesitan y deben ser mantenidos si se quiere que sean útiles a través del tiempo, lo que lleva a la conclusión de que el mantenimiento es inevitable.

Existen cuatro tipos de mantenimiento, los cuales son [SOM95].

Mantenimiento correctivo: consiste en realizar cambios al sistema de forma que corrija deficiencias y así cumpla con sus requerimientos.

Mantenimiento Preventivo: este tipo de mantenimiento es realizado con el objetivo de evitar posibles fallas en el sistema y sobre todo para llevar a cabo una posible reutilización.

Mantenimiento Perfectivo: consiste en realizar cambios al sistema que permitan que los requerimientos sean mas efectivos o simplemente que amplie las funciones del mismo.

Mantenimiento adaptativo: este tipo de mantenimiento es frecuentemente mas utilizado, debido a que su objetivo es cambiar al sistema para que cumpla con nuevos requerimientos.

Según sommerville (1995), la administración del mantenimiento se refiere a la planeación y predicción del proceso de cambios, para lo cual es necesario:

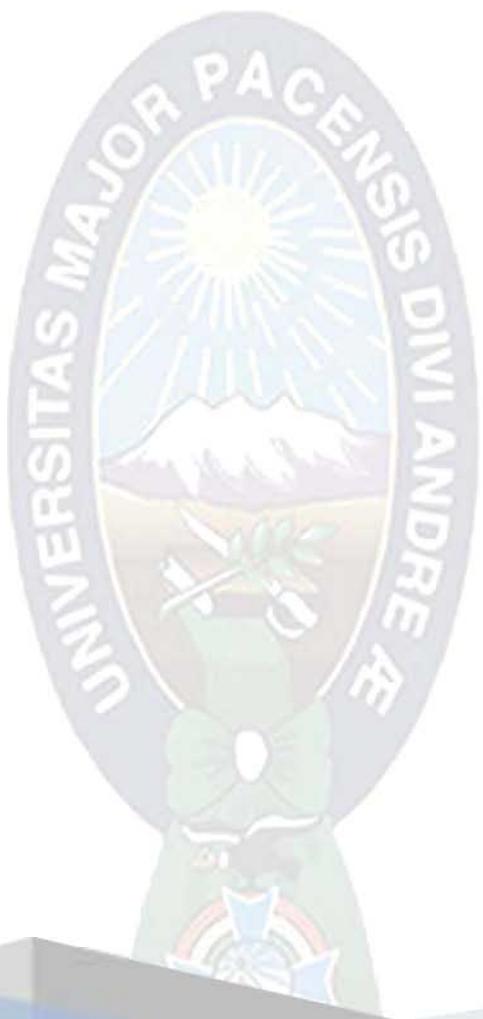
- Relacionar las compensaciones del mantenimiento al desempeño organizacional.
- Integrar el mantenimiento con el desarrollo.
- Crear un presupuesto para un mantenimiento preventivo.
- Elaborar planes para el mantenimiento en fases iniciales del proceso de desarrollo.
- Realizar esfuerzos y planes para llevar a cabo un programa de mantenimiento.

d) Portabilidad

La portabilidad es la capacidad de llevar exitosamente el sistema a otro entorno. Una unidad de software es portátil al grado de transferir la unidad del software de un entorno de sistema hardware y/o software a otro. Una unidad de software puede ser: un componente, un programa, un subsistema o el sistema completo.

El proceso de portabilidad tiene dos componentes que son:

- 1) Transportación.- se refiere a la fase física del programa, este quiere decir que es movido el programa físicamente a un sistema compatible.
- 2) Adaptación: es cualquier modificación que debe realizarse a la versión original, la misma puede ser realizado directamente por el sistema a través de algún proceso definido con anticipación o también a través del lenguaje en el que se están desarrollando.



Marco Aplicativo

CAPÍTULO 3

MARCO APlicativo

En este capítulo se podrá aplicar los fundamentos técnicos planteados en el capítulo anterior, de esta manera poder desarrollar el Sistema de Control y Seguimiento del Uso de vehículos vía web para la empresa CADEB. Aplicando la metodología RUP en el análisis y desarrollo del sistema.

3.1. Fase de Inicio

En esta fase se la dedicara a una planificación de las actividades que se realizaran dentro del marco del Proceso Unificado de Desarrollo identificando las funciones e hitos principales que involucran a cada actividad, considerando también los tiempos que fueron necesarios para su correspondiente realización.

3.1.1. Funciones Principales

Los Procesos que se realizan en CADE Talleres son, el de llevar el control del parque automotor en las diversas actividades de inversiones, mantenimiento y actividades desarrolladas por la empresa, de esta manera según su entorno de trabajo, realiza las siguientes tareas:

- Entrega en línea del presupuesto por el uso de cada vehículo a contabilidad
- Control de consumo de combustible
- Control y seguimiento del uso de Vehículos
- Distribución de gastos de consumo de combustible y mantenimiento de uso de vehículos a centros de costo y proyectos.
- Generación de asientos contables basado en la distribución de gastos por el uso de vehículos.

- **Entrega de presupuesto**

La entrega de presupuesto va de acuerdo a normas que se rigen en la empresa esto quiere decir que por parte de contabilidad se entrega un monto de dinero para la parte de el uso y mantenimiento de cada vehículo en CADE Talleres.

- **Uso de Vehículos**

Se registra el uso de los vehículos mediante las tarjetas que se usan para el control de los mismos, esto con la funcionalidad de saber, que persona realizó dicho trabajo y cuando lo realizó controlando así de ésta manera el uso que tuvo cada vehículo

- **Control del Combustible**

El control del consumo de combustible se realiza mediante tarjetas la cuales son llenadas por los conductores que manejan cada movilidad especificando en cada tarjeta el consumo de gasolina que se carga desde un surtidor de gasolina

- **Distribución de Gastos**

La distribución de gastos se basa en pasar un informe de todos los gastos de consumo de combustible, tanto del mantenimiento de los vehículos y también del uso que se le dió a cada uno de ellos en las distintas actividades de trabajo que se tuvo, enviando así un informe al área de contabilidad y los centros de costo (es decir subefaturas).

- **Generación de Asientos**

La Generación de asientos se encarga de enviar toda la información obtenida en base a la información generada de la distribución de gastos hacia la parte contable con el fin de tener un informe sobre los gastos de cada vehículo

De acuerdo a las tareas que se realiza en CADE Talleres se tiene la siguiente planificación de actividades dentro del marco de la fase de inicio la cual se presenta a continuación en la siguiente Tabla 3.1.

Tabla: 3.1 Planificación Fase de Inicio

Tareas	Semana1	Semana2	Semana3	Semana4
<ol style="list-style-type: none"> 1. Identificar necesidades y beneficios. Hito: declaración del producto definido. 2. Definir las salidas y entradas del sistema. Hito: Salidas y entradas definidas. 3. Definir el comportamiento Hito: definición de entradas y salidas completas. 4. Investigar el software existente. Hito: Componentes reutilizables identificados. 5. estimación del tamaño 6. Gestión de Riesgo Identificación de Riesgos Proyección de Riesgos. Superficies de Gestión de riesgos. Hito: Análisis y gestión de riesgos completo. 7. Planificación de la fase de Elaboración. 8. Evaluación de la fase de Inicio. 	[Green]	[Green]	[Green]	[Green]

3.1.2. Recopilación de Requisitos:

Los requerimientos identificados dentro de CADE Talleres están limitados en el marco del desarrollo del sistema a implementarse, se detallan a continuación:

1. Registrar Factores de distribución (costo/hora).
2. Registrar mantenimiento de los vehículos
3. Registrar parámetros de vehículos como ser : registro de empresas externas, registro de grupo de vehículos, registro de odómetro, registro de parámetros generales, registro de subgrupo de vehículos, registro de surtidores, registro de tipos de combustible.
4. Registrar y mantener la información actualizada de los vehículos.
5. Emitir y entregar reportes de distribución:
Por cargos por tipo, Cargos por tipo detalle, Costo por mes y vehículo, Promedio de costos por Vehículo, Detalle de distribución de uso de vehículos, Detalle de distribución de uso de vehículos por fechas, Detalle de distribución de uso de un vehículo, Resumen de distribución de uso de vehículos, Resumen de distribución de uso de vehículos por fechas, Resumen de distribución general por tipo de cargo.
6. Registrar el vale de consumo de combustible de los vehículos.

7. Registrar las tarjetas de uso de vehículos.
8. Distribución de los gastos por Combustible y Transportes en función de los registros de uso de vehículos, mantenimiento y cargos a los centros de costo que utilizaron las movilidades.
9. Generar de asientos contables en base a la distribución de los gastos de consumo de combustible y transporte.
10. Mejorar la parte de la interfaz para el usuario cambiando la estructura a un modo más accesible y fácil de manejar.

3.1.3. Análisis

3.1.3.1. Arquitectura Candidata:

Luego de haber realizado un estudio de la documentación y las funciones que el sistema realiza se pudo extractar una arquitectura funcional de CADE Talleres sobre la base de su funcionamiento actual. Véase Figura 3.1.

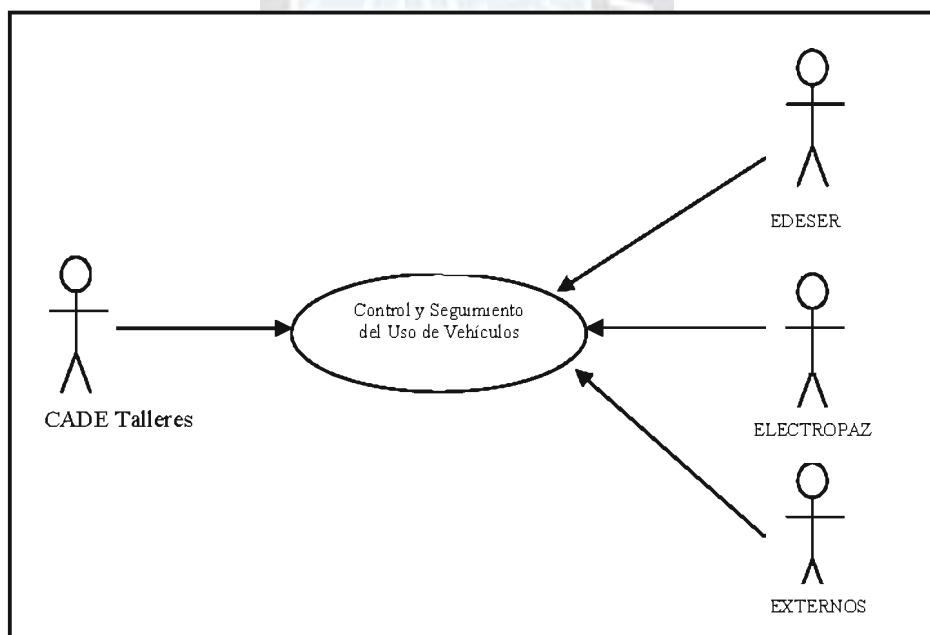


Figura 3.1 Arquitectura Candidata [Elaboración Propia]

3.1.4. Implementación

Por decisión del Jefe de Proyecto, el flujo de trabajo de la implementación no es llevada a cabo, puesto que se obtuvo una descripción de la arquitectura candidata como un avance significativo en la fase de inicio del proyecto.

3.1.5. Pruebas

Conforme a los requisitos obtenidos y el prototipo exploratorio realizado en la as400, el jefe de proyecto considero pertinente realizar algunas pruebas de los requisitos normales.

3.1.6. Análisis y Gestión de Riesgos

Un riesgo es un problema potencial que puede ocurrir o no, que afecta los futuros acontecimientos, es bueno evaluar su probabilidad su aparición, estimar su impacto y establecer un plan de contingencia por si ocurre el problema.

El riesgo implica las siguientes características:

- Incertidumbre (Acontecimiento que pueda o no ocurrir).
- Perdida (si el riesgo se convierte en realidad).
- Identificación de riesgos.

a) Identificación de Riesgos

El método aplicado para la identificación de los riesgos del presente trabajo, es la elaboración de la lista de comprobación de elementos de riesgos conocidos y predecibles en las siguientes categorías. Véase tabla 3.2.

b) Proyección del riesgo

La técnica usada para la estimación del riesgo se encuentra en la Tabla 3.2. donde la tercera columna asigna la probabilidad de ocurrencia de los riesgos identificados para el presente proyecto.

c) Refinamiento del riesgo

El refinamiento del riesgo fue llevado a cabo por los colaboradores del proyecto.

d) Reducción, supervisión y gestión de riesgo

La estrategia utilizada para reducción, supervisión y gestión de riesgo es la sugerida por Pressman, para lo cual se desarrollo un plan de reducción del riesgo, actividades de reducción y los informes relacionados a los riesgos posibles.

Tabla 3.2 Tabla de Riesgos [Elaboración Propia]

Riesgos	Categoría	Probabilidad	Impacto	RSGR
El presupuesto asignado para el desarrollo del proyecto es insuficiente.	IN	70%	2	
La fecha de entrega es sumamente ajustada.	TP	80%	2	
Cambios de requisitos por parte del cliente.	TP	60%	2	
Adición de requisitos por parte del cliente.	TP	50%	2	
Actividades y tareas retrasadas por fecha de entrega del sistema incumplida.	DP	60%	1	
Elevado costo en la compra del software requerido.	ED	80%	2	
Herramientas adquiridas no cumplen con las exigencias del sistema.	ED	40%	2	
Complejidad en el desarrollo del producto.	ED	80%	2	
Inexistencia de dispositivos para copias de seguridad	TC	60%	1	
El cambio de personal de la institución afecta al desarrollo y pruebas del sistema.	TE	40%	3	
Configuración inadecuada del software.	CC	60%	3	
Robo o eliminación de información por personas mal intencionadas.	TE	40%	2	
Software adquirido no compatible con el sistema.	TE	80%	2	
Perdida de información por conflictos sociales.	TE	50%	1	

<i>Subcategorías Genéricas de riesgo</i>	<i>Valores de impacto</i>
TP: Tamaño del producto	1: Catastrófico
IN: Impacto en el Negocio	2: Crítico
CC: Características del Cliente	3: Marginal
DP: Definición del proceso	4: Despreciable
ED: Entorno de Desarrollo	
TC: Tecnología a construir	
TE: Tamaño y experiencia en la plantilla	

3.1.7. Evaluación de la Fase de Inicio:

Para esta fase el hito está cumplido, consiguiendo los siguientes logros:

- Una arquitectura inicial, lo que identifica las partes difíciles y novedosas del sistema propuesto.
- Identificación de los riesgos hasta el punto de que sea factible construir el sistema.
- Obtención de la aprobación de todos los implicados en el proyecto. Y en particular de las personas de la institución.

3.2. Fase de Elaboración

3.2.1. Recopilación de Requisitos

Teniendo en cuenta la recopilación de requisitos que se realizó en la Fase de Inicio de manera esquematizada alcanzando al cincuenta por ciento de los requisitos recopilados, en esta fase se recopilan los requisitos con mayor detalle formulándolos en casos de uso y controlando los riesgos identificados.

Como resultado de la recopilación de los requisitos y después de un previo análisis con los encargados de CADE Talleres se obtuvo el siguiente documento de análisis del sistema propuesto, el que contiene información requerida por el cliente y las aportaciones realizadas por el analista de sistemas.

3.2.1.1. Documento de Análisis del “Sistema de Control y seguimiento del Uso de Vehículos”

Se debe realizar un sistema capaz de mantener una base de datos con información del uso de los vehículos, el mantenimiento de los vehículos, el uso de combustible que consume cada vehículo para luego así poder enviar esta información a el área de contabilidad y así poder predestinar un presupuesto para el Área de CADE Talleres de acuerdo a los gastos que este realiza para el Control y Seguimiento del uso de sus vehículos de manera que vaya en beneficio de la empresa.

Por lo tanto existen elementos importantes implicados en el sistema que son:

- Vehículos
 - Jefe de Talleres
 - Tarjetas de Uso de Vehículos
 - Vales de Consumo de combustible
 - Informes de la parte de mantenimiento del uso de vehículos.
- ✓ **Para Los vehículos es necesario:** el número de placa, empresa, nro. de vehículo, código odómetro, fecha de alta, recorrido inicial, horas de uso, grupo, marca, color, chasis, año vehículo, combustible subgrupo, motor, serie motor y cilindrada.
- ✓ **Para las Tarjetas de Vehículos:** Las tarjetas contiene la información sobre el uso de Vehículos, el conductor debe llenar cada vez que usando el vehículo concluye una actividad en la empresa, debe describir en forma breve pero completa la actividad desarrollada, el numero de horas empleadas, la cuenta a la que esa actividad se debe cargar, el kilometraje indicado en el odómetro y el nombre del

operador. Semanalmente debe entregar la tarjeta a personal de CADEB con el visto bueno del supervisor, quien revisa el tiempo y verifica la buena apropiación de cuentas.

Personal de Talleres transcribe al sistema el contenido completo de cada tarjeta.

- ✓ **Para los Parámetros Generales:** permite especificar los parámetros generales con los cuales el sistema funcionara, como ser registrar las cuentas contables, los tipos, conceptos presupuestarios a los que se harán cargo, etc.

- ✓ **Para las Estaciones de Servicio:** registra los datos de las estaciones de servicio que suministran combustible a los vehículos de la empresa.

- ✓ **Para los Odómetros:** Registra los tipos de odómetros con los cuales trabajan los vehículos de la empresa, con el fin de estandarizar las unidades de medida y llevar el control para su mantenimiento preventivo o correctivo.
También servirá para el cálculo de los factores de distribución de combustible y uso de vehículos.

- ✓ **Para las empresas Externas:** Registra a las empresas o sectores externos que por alguna razón necesiten usar una cuenta contable (cargos a terceros o cuentas por cobrar).

- ✓ **Para los Factores de los Vehículos:** Registra los factores de cada uno de los vehículos de forma mensual. Se denomina factores al costo por hora de alquiler de un vehículo y al rendimiento km./ litro de combustible.

- ✓ **Para el Cierre (Gestión de Vehículos):** permite registrar nuevas gestiones mensuales de vehículos, especificar el comportamiento de estas y generar los asientos contables de combustible y transportes.
Mediante la siguiente tabla se pretende comprender mejor el tipo de iteraciones que se pretende realizar.

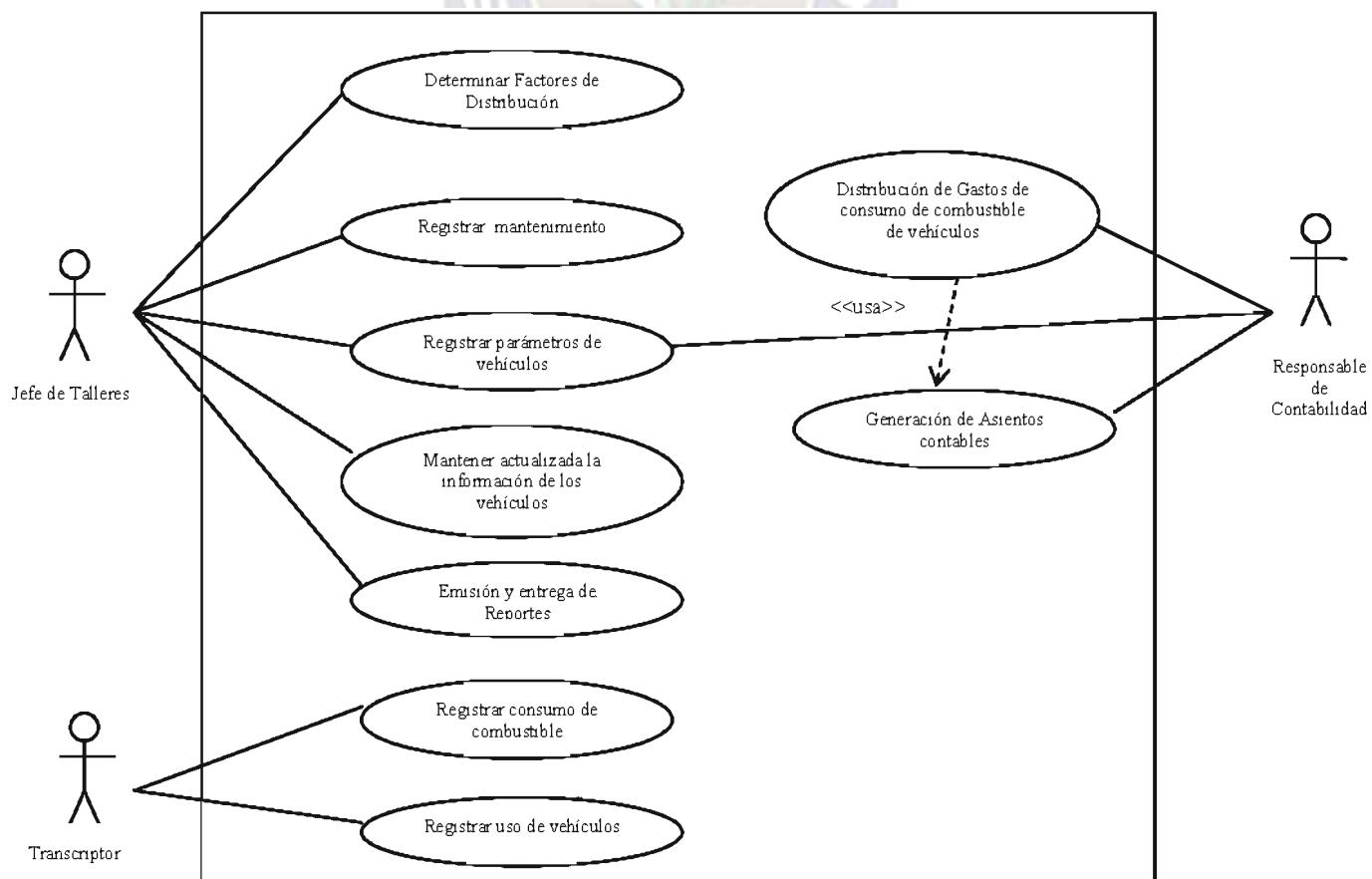
- ✓ **Para el Consumo de Combustible:** Cada vez que es necesario reponer el combustible de los vehículos, el conductor solicita un vale para determinada empresa expendedora de combustibles (se tiene contratos con un surtidor en La Paz, El Alto y otros), la empresa prepara cada 10 y 20 de mes un informe detallado de la atención de vehículos que es revisada en almacenes de ELECTROPAZ, luego se envía a CADE Talleres el cual transcribe al Sistema.

- ✓ **Para la parte de Mantenimiento:** CADE Talleres cada fin de Mes transcribe el monto que corresponde a cada vehículo por el mantenimiento efectuado en el sistema.
- ✓ **Distribución de Gastos:** El sistema procesa la base de datos y valora las horas del uso de vehículos con relación al mantenimiento y operación , proporciona un detalle por vehículo del costo de mantenimiento, CADE Talleres analiza la información obtenida y calcula en base a los datos del sistema y al tipo de vehículo un valor del mantenimiento por hora, este dato se introduce al sistema y sirve de base para el calculo contable y cargos a cuentas en las que el vehículo ha trabajado durante el periodo analizado.

3.2.2. Análisis

La recopilación de requisitos, fue realizada considerando la línea base de la arquitectura del sistema, y a la priorización de los riesgos percibidos, resultado de ello, fue el diagrama de casos de uso de primer de nivel, el cual es mostrado en la Figura 3.2.

Figura 3.2. Diagrama de Casos de Uso de Primer Nivel



3.2.2.1. Descripción de los Casos de Uso de Primer Nivel

Descripción de los casos de Uso de Primer nivel son los siguientes:

Tabla 3.3 Caso de uso de Distribución de Gastos

<i>Caso de Uso1:</i>	Distribución de Gastos
<i>Actor:</i>	Responsable de Contabilidad
<i>Tipo:</i>	Primario
<i>Descripción:</i>	Este caso de uso realiza la distribución de gastos del consumo de combustible y la distribución de gastos de transporte entre los diferentes Centros de Costos, Proyectos y Cuentas de Terceros dependiendo del uso de vehículos que hubieran tenido y del costo/hora.

Tabla 3.4 Caso de uso Mantener Información de Vehículos

<i>Caso de Uso2:</i>	Mantener Información de Vehículos
<i>Actor:</i>	Jefe de Talleres
<i>Tipo:</i>	Primario
<i>Descripción:</i>	Este caso de uso realiza el mantenimiento de los datos de los vehículos. El Jefe de Talleres mantiene actualizada información de vehículos.

Tabla 3.5 Caso de uso Registro de Vehículos

<i>Caso de Uso3:</i>	Registro del Uso de Vehículos
<i>Actor:</i>	Transcriptor
<i>Tipo:</i>	Primario
<i>Descripción:</i>	Este caso de uso realiza el registro de las tarjetas de uso de vehículos. Las cuales son llenadas por los conductores y luego son transcritas por el encargado de registro en este caso el transcriptor.

Tabla 3.6 Caso de uso Determinar Factores de Distribución

<i>Caso de Uso4:</i>	Determinar Factores de Distribución
<i>Actor:</i>	Jefe de Talleres
<i>Tipo:</i>	Primario
<i>Descripción:</i>	Este caso de uso determina los factores de distribución (Costo/Hora) de vehículos a través de cálculos.

Tabla 3.7 Caso de uso Registrar consumo de Combustible

<i>Caso de Uso5:</i>	Registrar consumo de Combustible
<i>Actor:</i>	Transcriptor
<i>Tipo:</i>	Primario
<i>Descripción:</i>	Este caso de uso realiza el registro de los vales del consumo de combustible los cuales son llenados por los conductores y luego son transcritas por el encargado de registro en este caso el transcriptor.

Tabla 3.8 Caso de uso Registrar mantenimiento

<i>Caso de Uso6:</i>	Registrar mantenimiento
<i>Actor:</i>	Jefe de Talleres
<i>Tipo:</i>	Primario
<i>Descripción:</i>	Este caso de uso importa la información de gastos de mantenimiento de vehículos realizada por otra empresa.

Tabla 3.9 Caso de uso Registrar parámetros

<i>Caso de Uso7:</i>	Registrar parámetros
<i>Actor:</i>	Jefe de Talleres
<i>Tipo:</i>	Secundario
<i>Descripción:</i>	Este caso de uso realiza el registro de todos los parámetros involucrados como ser: los odómetros (unidad de medida de recorrido), surtidores, tipos de combustible, grupos y subgrupo de vehículos.

Tabla 3.10 Caso de uso Generar Asientos contables

<i>Caso de Uso7:</i>	Generar Asientos Contables
<i>Actor:</i>	Responsable de Contabilidad
<i>Tipo:</i>	Primario
<i>Descripción:</i>	Este caso de uso realiza el calculo contable de costo de consumo de combustible es directo por el sistema y se apropiá también a las cuentas en forma automática.

- Los actores que intervienen en el sistema son los siguientes:

Tabla 3.11 Tabla que representa al actor Jefe de Talleres

Nombre del actor:	Jefe de Talleres
Definición:	Es la persona que esta en contacto directo con los clientes, tiene acceso a casi toda la información del sistema.
Notas:	Es el responsable de la información del sistema.

Tabla 3.12 Tabla que representa al actor Responsable de contabilidad

Nombre del actor:	Responsable de Contabilidad
Definición:	Es la persona que esta a cargo de ejecutar las diversas tareas como ser ejecutar la parte contable y la distribución de gastos del consumo del combustible.
Notas:	Es la persona que se encarga de realizar la generación de asientos contables y realizar la distribución de gastos.

Tabla 3.13 Tabla que representa al actor Transcriptor

Nombre del actor:	Responsable de Contabilidad
Definición:	Es la persona que registra el uso de los vehículos y el registro de consumo del combustible.
Notas:	Es la persona responsable de registrar el uso que tuvo cada vehículo y en que cada actividad fue usado además de registrar el consumo de gasolina que utiliza cada vehículo.

- Las interfaces con los actores del sistema son descritas a continuación:

Tabla 3.14 Tabla para representar la interfaz del Jefe de Talleres

Nombre de la interfaz:	Interfaz del jefe de Talleres
Definición:	El jefe de Talleres tendrá acceso al 80% de la información que el sistema soporta donde su capacidad de maniobra no es limitada

Tabla 3.15. Tabla para representar la interfaz del responsable de contabilidad

Nombre de la interfaz:	Interfaz del responsable de contabilidad
Definición:	El responsable de contabilidad solo tendrá acceso a la parte contable en este caso la generación de asientos contables el cierre contable, la distribución de gastos, parámetros generales y empresas externas.

Tabla 3.16. Tabla para representar la interfaz del Transcriptor

Nombre de la interfaz:	Interfaz del Transcriptor
Definición:	El transcriptor solo tendrá acceso a el registro de consumo de combustible y el registrar el uso de los vehículos.

3.2.3. Diseño

Se implementara el diseño de los casos de uso arquitectónicamente significativos, como resultado de las tareas asignadas a esta fase, consideraremos la clase vehículo representada en la Figura 3.7 que es un elemento significativo del cual se muestra sus características principales, más adelante se observara el esquema de casos de uso para el registro de vehículos.



Figura 3.3. Representación de la clase vehículo

A continuación se muestran los atributos de la clase vehículo los cuales muestran la información más importante.

Atributos de la tabla vehículos:

- Placa: captura el número de placa del vehículo para su correspondiente identificación.
- Marca: captura la marca del vehículo para su identificación
- Modelo: describe el tipo de modelo que es el vehículo
- Chasis: captura el número de chasis del vehículo para su identificación
- Plazas: captura el número de asientos que tiene cada vehículo
- Motor: captura el número de motor
- Km. Inicial: captura el kilometraje inicial con el que empezó el vehículo.
- Lim. Odómetro: captura el número inicial de odómetro.
- Horas: horas que se uso ese vehículo.
- Fecha alta: captura la fecha de alta que se le dio a ese vehículo.
- Fecha baja: captura la fecha de baja que se le dio a ese vehículo.
- Estado: captura el estado del vehículo el cual puede ser habilitado o inhabilitado.
- Usuario: captura el código de usuario que está dando de alta a un vehículo.
- Código: captura el código que es asignado a cada vehículo.

Flujo Básico de la clase Vehículo:

1. El usuario selecciona la opción “Vehículo”.
2. El sistema despliega el listado de los vehículos existentes con las siguientes opciones: Insertar, Modificar y Dar de Baja.
3. El usuario selecciona la opción “Insertar”
4. El sistema despliega el formulario para la alta del vehículo.
5. El usuario ingresa los datos del vehículo.
6. El sistema valida los datos ingresados.
7. El usuario selecciona la opción “Confirmar”.
8. El sistema graba la información del vehículo.
9. El sistema despliega otro formulario en blanco hasta que el encargado seleccione la opción “Atrás”.

Descripción del Caso de Uso Mantener Información Vehículos

CASO DE USO: Mantener Información Vehículos

ACTORES: Jefe de Talleres

PROPOSITO: Mantener actualizada la información de los vehículos.

RESUMEN: Este caso de uso realiza el mantenimiento de los datos de los vehículos, el jefe de Talleres es el encargado de mantener actualizada la información de los vehículos.

TIPO: Primario

CURSO NORMAL DE ACCIONES

ACCION DEL ACTOR:

1. El Jefe de Talleres selecciona la opción “mantenimiento de vehículos”.
3. El Jefe de Talleres selecciona la opción “insertar”
5. El Jefe de Talleres ingresa los datos del vehículo.
7. El Jefe de Talleres selecciona la opción Grabar.

RESPUESTA DEL SISTEMA:

2. El Sistema despliega el listado de los vehículos existentes con las siguientes opciones: insertar, modificar y dar de baja.
4. El Sistema despliega el formulario de alta de vehículos.
6. El Sistema valida los datos ingresados.
8. El Sistema graba la información del vehículo.
9. El Sistema despliega otro formulario en blanco hasta que el encargado seleccione la opción “Salir”

3.2.4. Implementación

Habiendo construido parcialmente los casos de uso paso a paso, ahora se podrá ver más a detalle construyendo un diagrama de Colaboración para así comprender más a fondo el proceso que se debe de realizar en cada uno de los casos de uso a continuación veremos un ejemplo del caso de uso registro de vehículo plasmado en los siguientes diagramas de colaboración del registro grupo de Vehículos y registro de Combustible

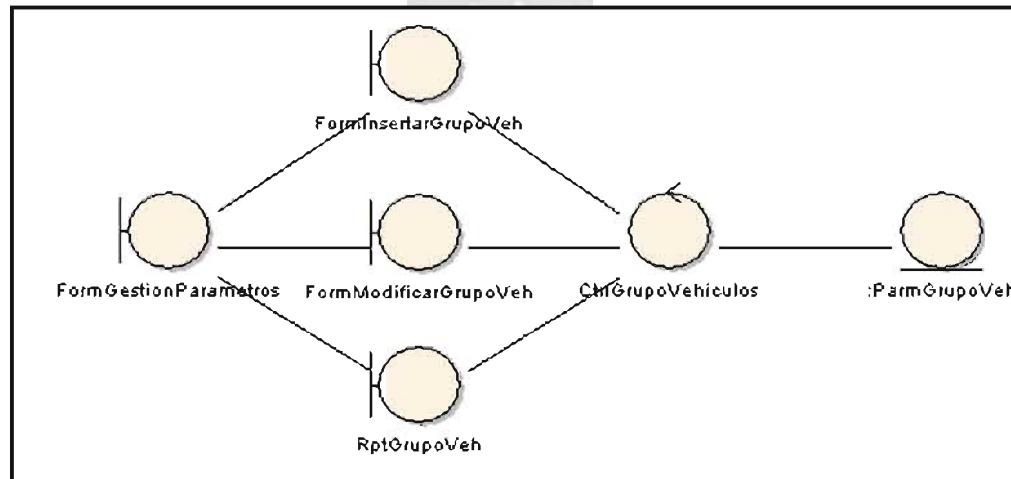


Figura 3.4. Diagrama de colaboración Registro grupo de Vehículos

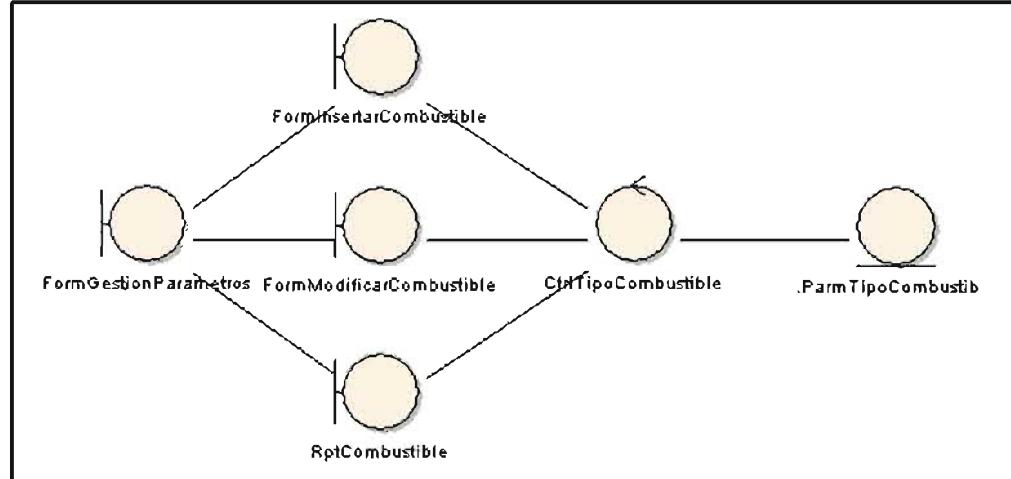


Figura 3.5. Diagrama de Colaboración Registro de Combustible

3.2.5. Pruebas

La programación y planificación de las pruebas de la gestión de vehículos se realizaron basados en el siguiente cronograma presentado en la siguiente tabla.

Tabla 3.17. Programación y Planificación de pruebas

Clasificación	Caso de Uso	Justificación
Alto	<ul style="list-style-type: none"> – Distribución de gastos. – Registro de uso de vehículos – Generación de asientos contables – Registrar consumo de combustible. 	Corresponde a criterios en los que se vio un nivel de complejidad alto ya que se encuentran relacionados con los subsistemas de contabilidad, recursos humanos, proyectos y presupuesto financiero.
Mediano	<ul style="list-style-type: none"> – Mantener información de vehículos – Revertir cierre de vehículos – Revertir cierre contable. 	Se los considera como casos de uso medianos por no ser de muy alto nivel realizando procesos importantes para enviar a los otros subsistemas.
Bajo	<ul style="list-style-type: none"> – Registro de odómetro – Registro de tipos de combustibles – Registro de surtidores – Registro de grupos de vehículos – Registro de subgrupos de vehículos – Registro mantenimiento – Determinación de factores de distribución – Registro de parámetros generales – Registro de empresas externas. 	Tareas principales para el proceso de la información que es significativa para el control y seguimiento del vehículo. No afectan de manera significativa a otros subsistemas

3.2.6. Criterios de Evaluación

Los hitos logrados en esta fase de elaboración fueron los siguientes:

- Identificación de los casos de uso y actores del sistema.
- Una arquitectura ampliada, lo que identifica los subsistemas del sistema propuesto.
- Revisión y control de los riesgos considerados en la fase de inicio hasta el punto que es factible continuar con la construcción del sistema.
- Planificación y programación de los subsistemas, previa a la construcción.
- Aprobación de los miembros implicados en el proyecto y usuarios de la institución.

3.3. Fase de Construcción

En la presente fase se detallan los casos de uso y escenarios correspondientes identificados en la fase anterior (fase de elaboración), los que se exponen con el necesario nivel en detalle, sencillez y rapidez para su inmediata comprensión, haciendo que el usuario pueda darse cuenta de las acciones que debe realizar y cuáles serán las operaciones y respuestas del sistema.

3.3.1. Modelo del Negocio

En este punto el modelo del negocio se basa en el modelo de casos de uso del negocio y el modelo del dominio como en la anterior fase nos habíamos planteado un esquema introductorio del diagrama de casos de uso en la fase de elaboración ahora podemos complementar mas el modelo de casos de uso del negocio.

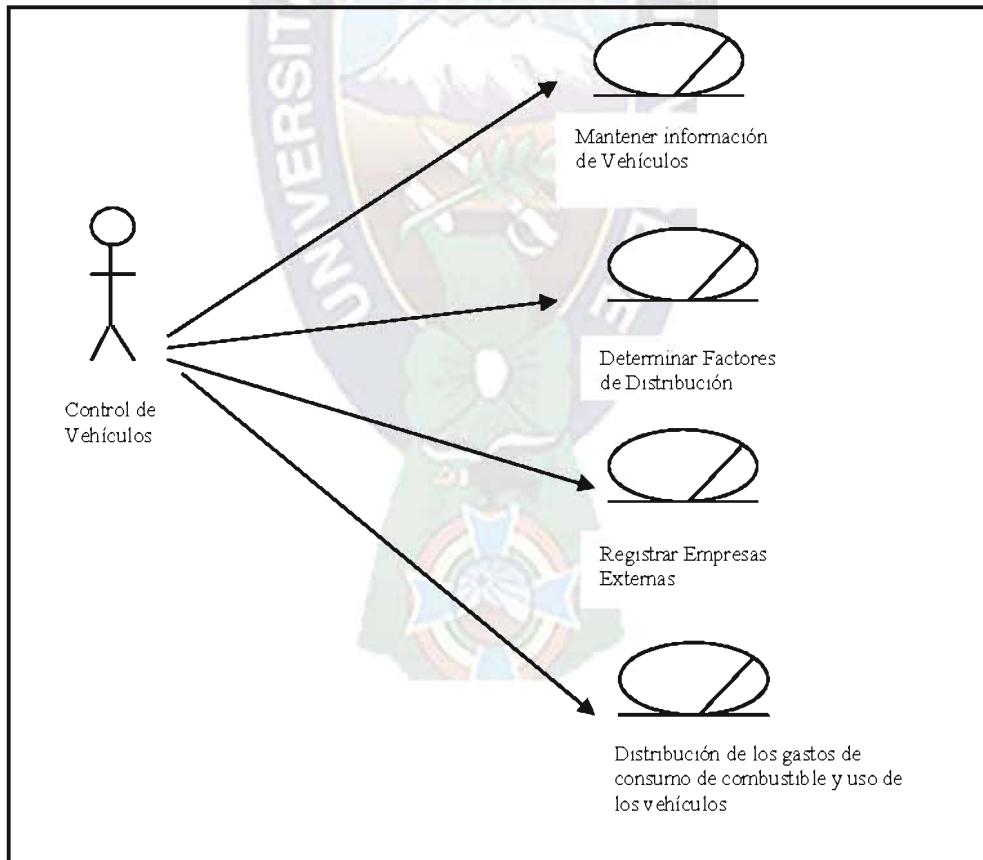
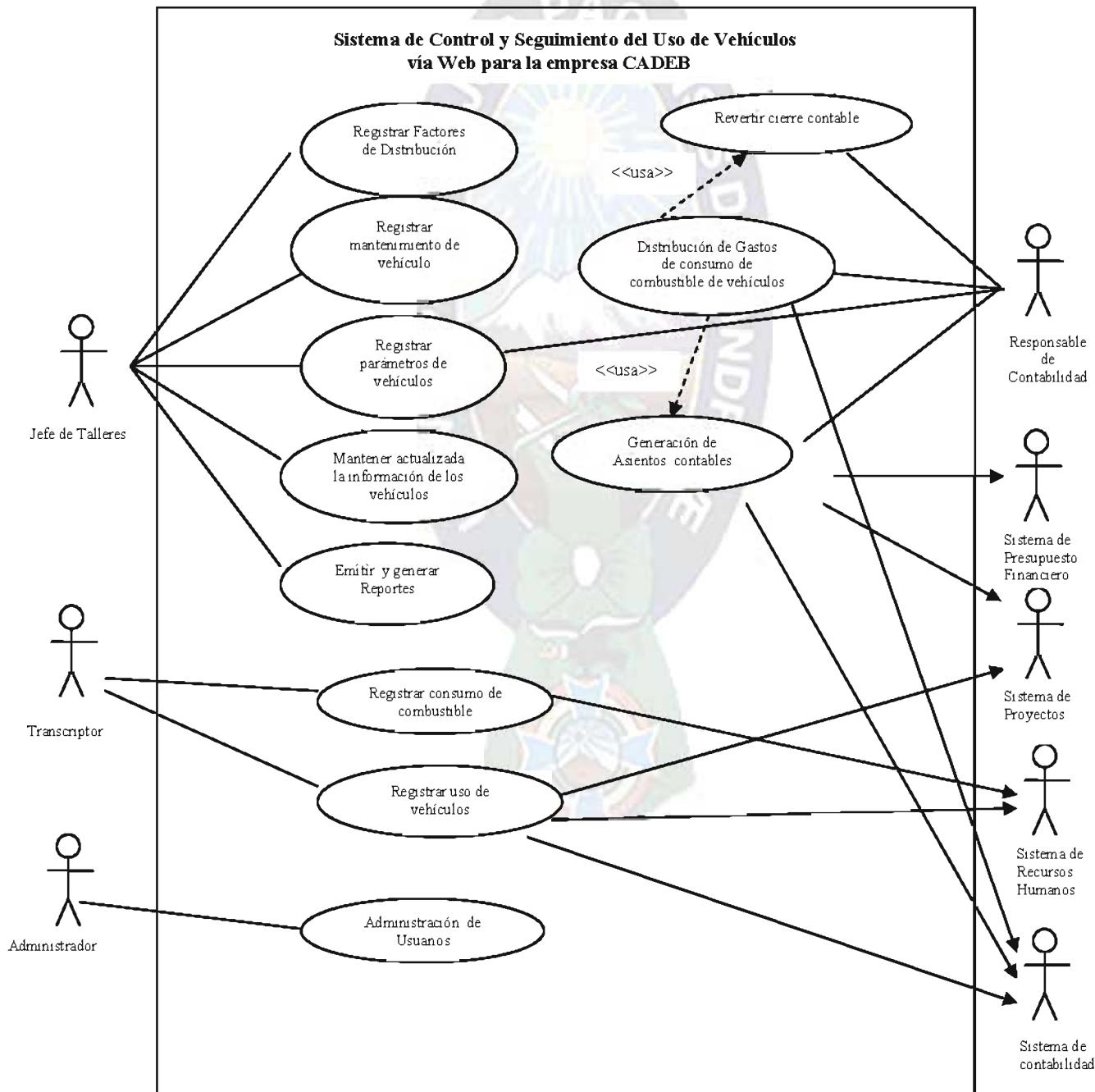


Figura 3.6. Modelo de casos de Uso del Negocio

3.3.2. Recopilación de requisitos

Los requisitos para la construcción se muestran a través del diagrama de casos de uso de segundo nivel mostrado a continuación en la Figura 3.10. Después de haber hecho énfasis en la etapa de Elaboración teniendo parcialmente parte del análisis y diseño del Sistema en esta etapa se pretende profundizar el análisis y diseño del sistema propuesto para que así se determine en su totalidad todos los casos de uso a realizar.

Figura 3.7. Diagrama de Casos de Uso de Segundo Nivel



3.3.3 Análisis

A partir del diagrama de casos de uso realizado anteriormente se describirá profundamente cada uno de los casos de uso para ello se utilizara el modelo de plantilla recomendado por LARMAN.

A continuación vemos la descripción de algunos casos de Uso en formato extendido el resto se encuentra en el Anexo D

Tabla 3.18. Caso de Uso extendido Mantener Información Vehículos

CASO DE USO:	Mantener Información Vehículos
ACTORES:	Jefe de Talleres
PROPOSITO:	Mantener actualizada la información de los vehículos.
RESUMEN:	Este caso de uso realiza el mantenimiento de los datos de los vehículos, el jefe de talleres es el encargado de mantener actualizada la información de los vehículos.
TIPO:	Primario
REFERENCIA CRUZADA:	
CURSO NORMAL DE ACCIONES	
ACCION DEL ACTOR:	RESPUESTA DEL SISTEMA:
1. El Jefe de Talleres selecciona la opción “mantenimiento de vehículos”. 3. El Jefe de Talleres selecciona la opción “insertar” 5. El Jefe de Talleres ingresa los datos del vehículo. 7. El Jefe de Talleres selecciona la opción grabar	2. El Sistema despliega el listado de los vehículos existentes con las siguientes opciones: insertar, modificar y dar de baja 4. El Sistema despliega el formulario de alta de vehículos. 6. El Sistema valida los datos ingresados. 8. El Sistema graba la información del vehículo. 9. El Sistema despliega otro formulario en blanco hasta que el encargado seleccione la opción “salir”.
CURSOS ALTERNATIVOS	
<ul style="list-style-type: none"> - Se realiza la modificación de los vehículos. - Se da de baja a un vehículo. 	

Tabla 3.19. Caso de Uso extendido Registrar Uso de Vehículos

CASO DE USO:	Registrar uso de Vehículos
ACTORES:	Encargado de Registro
PROPOSITO:	Registrar las tarjetas del uso de vehículos.
RESUMEN:	Este caso de uso realiza el registro de las tarjetas de uso de los vehículos, las cuales son llenadas por los conductores y son transcritas por el encargado de registro.
TIPO:	Primario
REFERENCIA CRUZADA:	
CURSO NORMAL DE ACCIONES	
ACCION DEL ACTOR:	RESPUESTA DEL SISTEMA:
1. El encargado de registro selecciona la opción “Insertar Tarjeta”. 3. El encargado de registro ingresa los datos de la tarjeta 5. El encargado de registro selecciona la opción “Grabar Tarjeta”	2. El sistema despliega un formulario en blanco de uso de vehículos. 4. El sistema valida los datos ingresados relacionados con otros módulos del sistema como ser: Recursos Humanos, Contable y Proyectos. 6. El sistema graba la tarjeta. 7. El sistema despliega otro formulario en blanco hasta que el encargado seleccione la opción “Salir”.
CURSOS ALTERNATIVOS	
<ul style="list-style-type: none"> – Se realiza la modificación de las tarjetas – Se realiza la eliminación de tarjeta. – Cierre de transcripción 	

3.3.3.1. Construcción del Modelo conceptual

El modelo conceptual es un paso esencial de la fase de análisis es una representación de conceptos en un dominio del problema [MO95, FOWLER].

El modelo conceptual se comenzará a construir partiendo de la lista de informaciones que hemos obtenido en el modelado del dominio. A continuación se mencionan una lista de conceptos adecuados para el Control y Seguimiento del uso de vehículos sujeto a los requerimientos.

Tabla 3.20 Categoría de Conceptos Sistema de Control y Seguimiento del uso de Vehículos

Categoría del concepto	Ejemplos
Objeto físicos o tangibles	Vehículos Vale de uso de Vehículos Vale de consumo de combustible
Eventos realizados	Parámetros Grupos Parámetros Subgrupos Parámetros Externas Parámetros Surtidores Parámetros Odómetros Parámetros Tipo combustible Parámetros generales
Papel de las personas	Jefe de talleres Transcriptor Responsable de contabilidad
Transacción	Factores costo/hora mantenimiento
Procesos	Distribución de gastos de consumo de combustible Generación asientos de distribución Reversiones

Con la lista anterior de conceptos importantes podemos representar gráficamente una estructura estática de UML. Esto con el fin de mostrar un modelo conceptual inicial del dominio del sistema de Control y Seguimiento del uso de Vehículos, este esquema nos ayudara a tener mas clara la construcción de un diagrama de clases en la siguiente fase que es la de construcción.

Sistema de Control y Seguimiento del Uso de Vehículos Vía Web

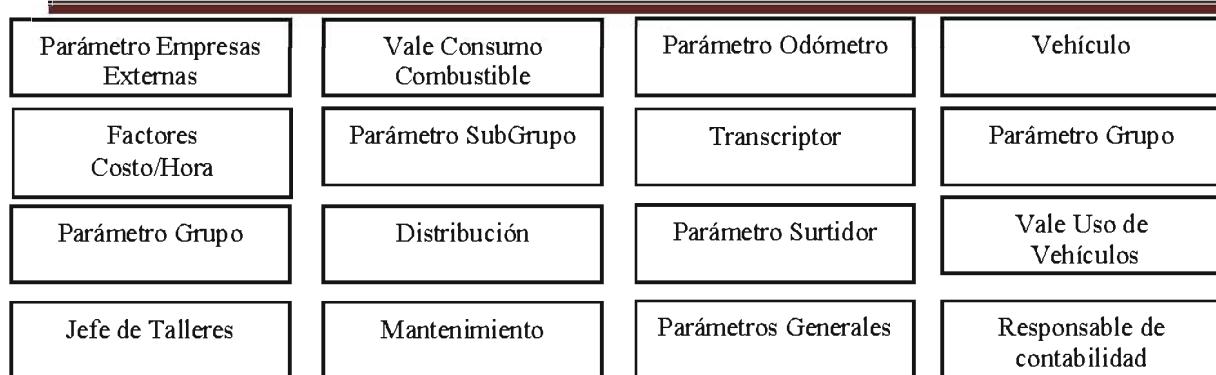


Figura 3.8 Modelo Conceptual Inicial del Dominio

i) Identificación de atributos del sistema

Grupo Vehículo	Odómetro	Empresas Externas	Generales
Código Descripción Estado Fecha Estado Cód Subgrupo	Código Parámetro Odómetro Descripción Estado Fecha Estado	Código Sigla Descripción Estado Fecha Estado	Código Descripción Estado Fecha Estado Tipo Asiento Tarjeta día Hasta Vale Día Hasta Código de Usuario Fecha modificación
Tipo Combustible	Surtidor	SubGrupo	Distribución
Código Descripción Estado Fecha Estado	Código Impuesto Descripción Estado Fecha Estado	Código Descripción Estado Fecha Estado	Total Kilometraje Total Horas Importe combustible Importe transporte Gestión Tipo Cargo Cargo Folio Código Usuario
Factor Costo/ Hora	Mantenimiento	Giro Cambio Odómetro	
Factor Costo/Hora Código Vehículo Gestión Kilómetros por litro Usuario	Mantenimiento Correctivo Mantenimiento Preventivo Código Vehículo Gestión Usuario	Kilometraje inicio Motivo registro Fecha cambio Número correlativo Código Usuario	
Vehículo	Vale Uso Vehículo	Vale Consumo Combustible	
Año Modelo Marca Chasis Cilindrada Motor Placa Km Inicial Lím Odómetro Horas Fecha alta Fecha baja Estado Usuario Código Código odómetro	Número Registro Código Vehículo Gestión Tipo Cargo Cargo Folio Descripción Fecha Uso Horas Uso Minutos uso Registro de Lectura Importe combustible Importe transporte Fecha transcripción Registro de recorrido Factor odómetro	Cantidad despachada Fecha atención Kilometraje Litros autorizados Persona que despacha Neto combustible Tipo combustible Gestión Número de vale Empleado que autoriza Empleado que recoge Empleado que solicita Código Usuario Código Surtidor Código Vehículo	

Figura 3.9 Atributos en el Modelo conceptual Inicial

ii) Especificación de operaciones de algunas tablas:

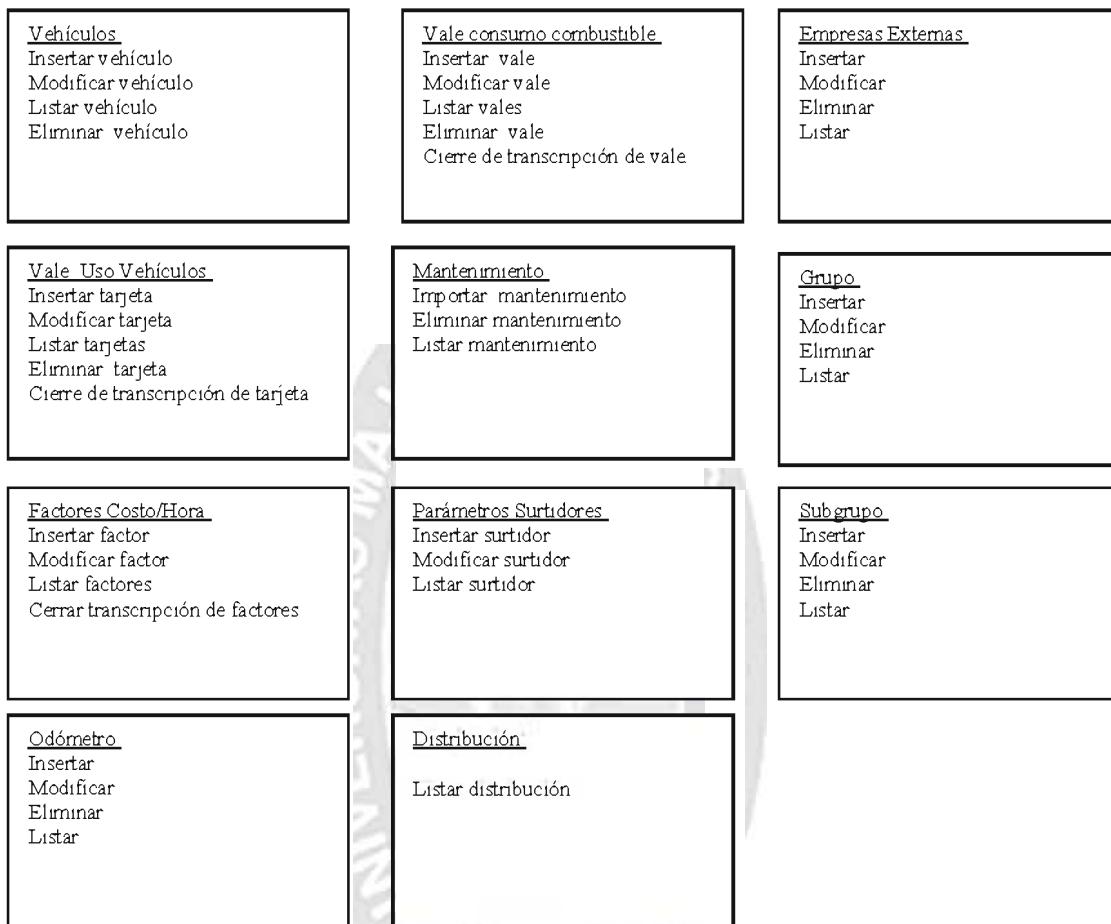


Figura 3.10 Operaciones de las clases en el Modelo Conceptual Inicial

Habiendo construido un modelo conceptual inicial el siguiente paso es plantear un modelo conceptual completo para que en esta parte del análisis se identifique claramente todos los conceptos del dominio significativos e importantes relacionados con los casos de uso.

Una vez definido el modelo conceptual podemos ver que la lógica de negocios que se definió anteriormente realizando un análisis del sistema facilita el desarrollo para un mejor diseño. Mediante el patrón MVC se maneja por separado la parte del modelo para poder acceder a la capa de almacenamiento de Datos de manera que se vuelve independiente.

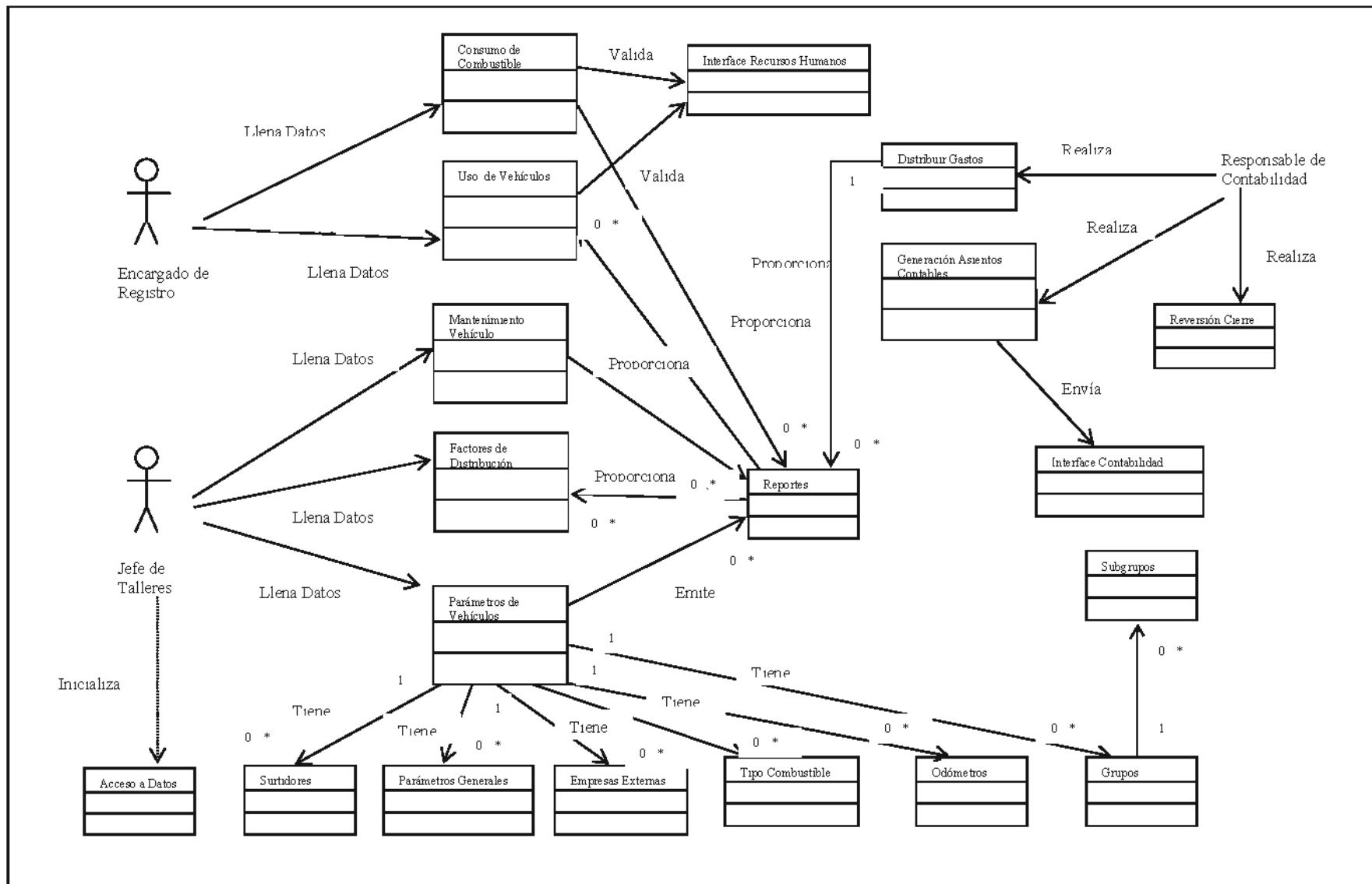


Figura 3.11. Modelo Conceptual

3.3.3.2. Glosario

En el Glosario se registrara los términos que requieren información, para mejorar la comunicación [LARMAN, 1999]. A continuación se muestra el glosario de términos correspondiente al sistema.

Tabla 3.21. Glosario de Términos

Termino	Categoría	Comentario
Distribución de Gastos de consumo de combustible de vehículos	Casos de Uso	Se encarga de generar información mediante el uso de los vehículos, mantenimiento de los mismos y el consumo de combustible
Mantenimiento de Vehículos	Caso de Uso	uso importa la información de gastos de mantenimiento de vehículos realizada por otra empresa.
Emitir y Generar Reportes	Caso de Uso	Emite datos referente al uso de vehículos , mantenimiento y control de los mismos.
Registrar parámetros de vehículos	Caso de Uso	Se encarga de registrar los Grupos, Subgrupos, Empresas externas , Surtidores, Odómetro, Parámetro Generales.
Registra consumo de combustible	Casos de uso	Se encarga de realizar el control del consumo de combustible mediante los vales.
Generación de Asientos	Casos de uso	Se encarga de enviar información a la parte contable empleando la información de la distribución

3.3.3.3. Diagramas de Secuencia

Como se desarrollaron los casos de Uso en formato expandido ahora se muestra los diagramas de secuencia a continuación se tiene algunos diagramas de secuencia de algunos casos de Uso el resto se encuentra en el Anexo H.

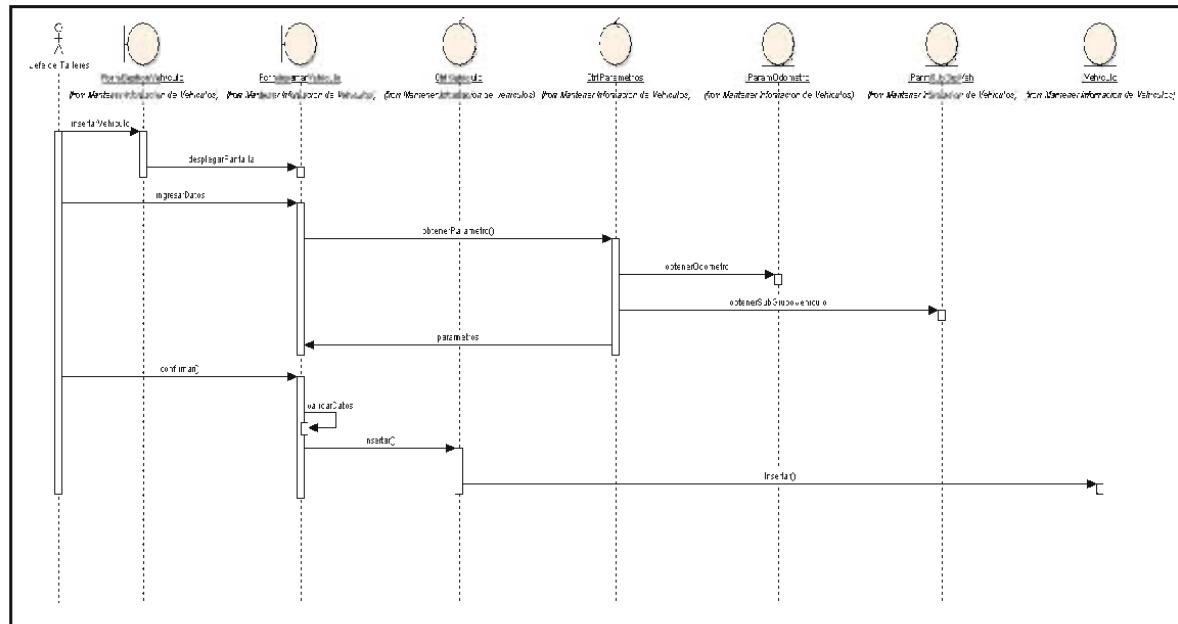


Figura 3.12 Diagrama de secuencia registro del uso de Vehículos

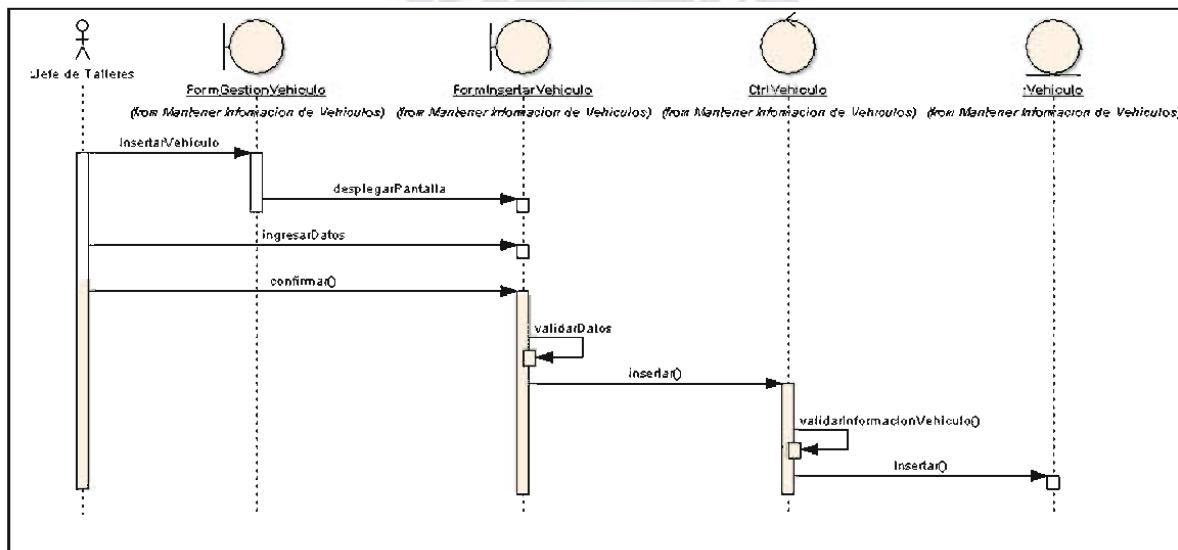


Figura 3.13. Diagrama de secuencia mantener información Vehículos

3.3.3.4. Diagramas de contrato

Para ver el comportamiento del sistema a través de los cambios que existen se utilizaran contratos véase a continuación la tabla 3.22 el resto se encuentra en el anexo E.

Tabla 3.22. Contrato Registrar Uso de Vehículos

CONTRATO	
Nombre:	Registro vehículos (pVehCod, pVehProHrs, pVehPlaca, pVehAno, pVehMarca, pVehModelo, pVehCilin, pVehChasis, pVehMotor, pVehOdomCod, pVehSgrpCod, pVehCctCod, pVehKmIni, pVehLimOdom, pVehFecAlt, pVehUsrCod)
Responsabilidades:	Capturar (registrar) la información del registro de los vehículos.
Tipo:	Sistema
Referencias cruzadas:	Funciones del sistema. Caso de uso registrar uso de vehículos
Notas:	Utilizar el acceso súper rápido a la base de datos
Excepciones:	De existir error en los procesos de Modificar e Insertar, el sistema despliega un mensaje de error.
Salida:	
Precondiciones:	Antes de que el caso de uso empiece el encargado ya tuvo que hacer un login en el sistema y tener la autorización correspondiente.
Poscondiciones:	

3.3.4. Diseño

3.3.4.1. Diagrama de Colaboración

A continuación se muestra el diseño de algunos casos de uso que se abordaron para la realización del sistema los cuales se muestran a continuación en los siguientes diagramas de colaboración, el resto se encuentra en el anexo F.

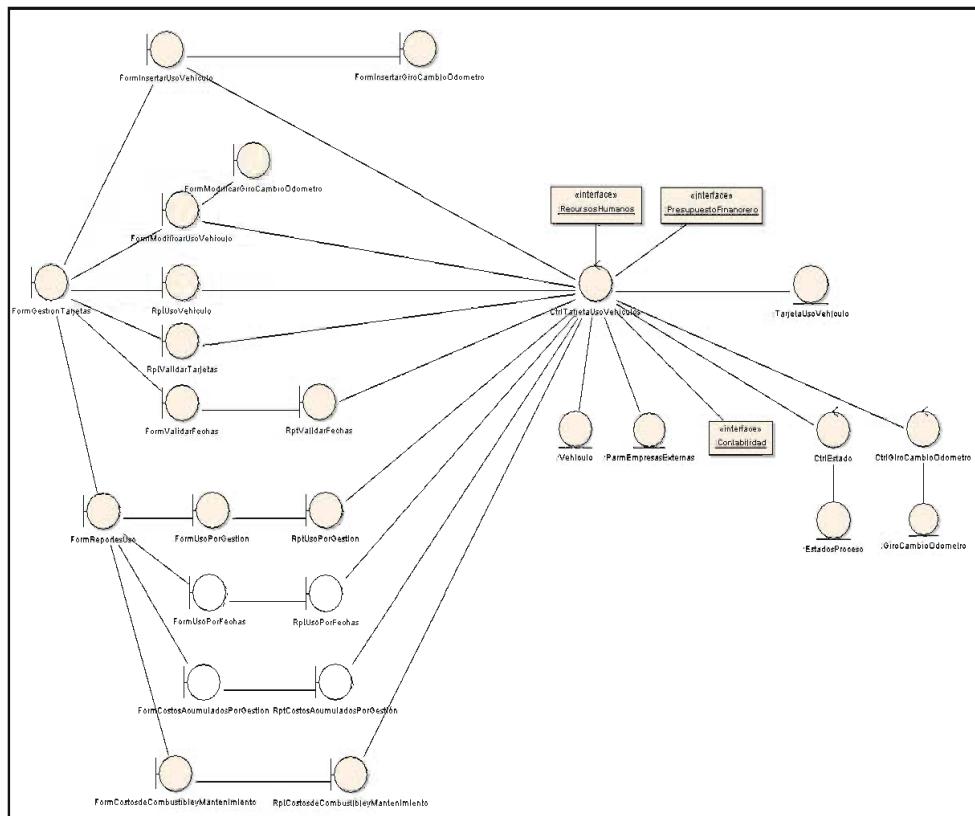


Figura 3.14. Diagrama de colaboración Registro del Uso de Vehículos

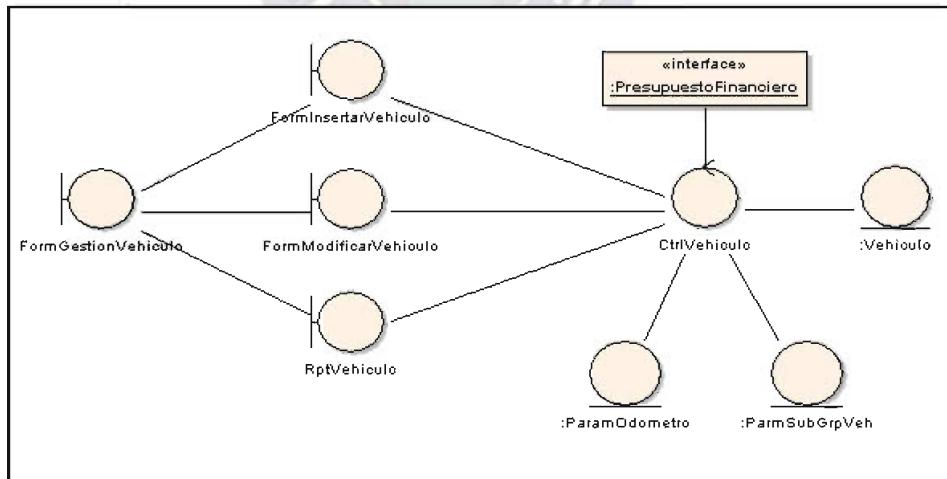


Figura 3.15. Diagrama de colaboración mantener información vehículos

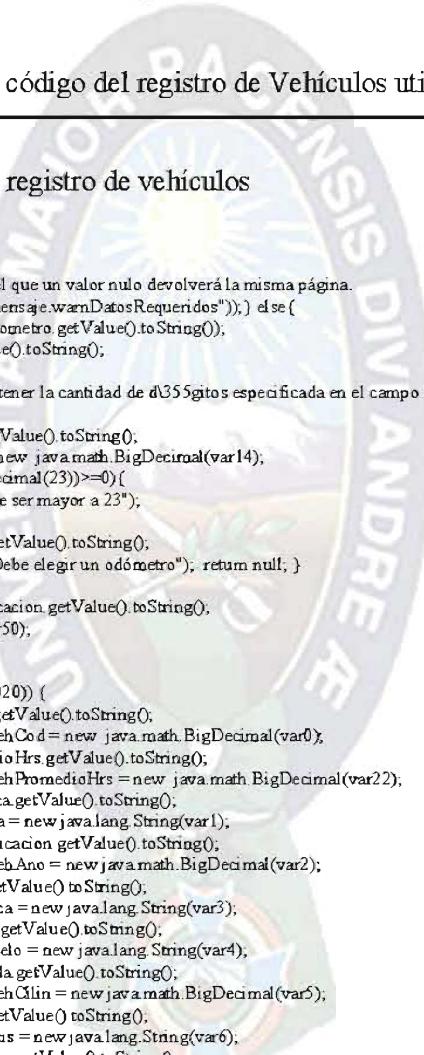
A continuación se tiene el código del diagrama de colaboración del registro del uso de vehículos donde se puede ver el modo de implementación del patrón MVC con el controlador que se encarga de procesar las peticiones del usuario realizando los cambios por separado si es que se trata de la vista o del modelo, en este punto podemos ver también que se aplica el modelo de tres capas que al igual que el MVC podemos manejar de manera independiente de la capa de lógica de negocios, siendo que con el MVC podemos realizar un mantenimiento sencillo en la parte de la lógica de negocios, lo cual facilita y garantiza un mejor diseño haciendo así reutilizable el código para un mejor mantenimiento.

Implementación de código del registro de Vehículos utilizando el patrón MVC



Formulario para el registro de vehículos

Vehiculo.jsp



```

public String btnConfirmar_action() {
    // nombre de caso de navegación en el que un valor nulo devolverá la misma página.
    warn(mensaje.getURL("vehiculos.mensaje.warnDatosRequeridos")); else {
        int val = Integer.parseInt(textLimOdometro.getValue().toString());
        String val2 = textKmInicial.getValue().toString();
        if(val < val2.length()) {
            warn("El Kilometraje inicial debe tener la cantidad de dígitos especificada en el campo Dígitos de Odómetro");
            return null;
        }
        String var14 = textPromedioHrs.getValue().toString();
        java.math.BigDecimal PromHrs = new java.math.BigDecimal(var14);
        if(PromHrs.compareTo(new BigDecimal(23))>=0) {
            warn("El promedio horas no debe ser mayor a 23");
            return null;
        }
        String odometro = dropOdometro.getValue().toString();
        if(odometro.equals("0")) { warn("Debe elegir un odómetro"); return null; }
        try {
            String var50 = textAñoFabricacion.getValue().toString();
            int año = Integer.parseInt(var50);
            int ini = 1970;
            int fin = 2020;
            if((1970 < año) && (año<2020)) {
                String var0 = textoCodigo.getValue().toString();
                java.math.BigDecimal pVehCod = new java.math.BigDecimal(var0);
                String var22 = textPromedioHrs.getValue().toString();
                java.math.BigDecimal pVehPromedioHrs = new java.math.BigDecimal(var22);
                String var1 = textNumPlaca.getValue().toString();
                java.lang.String pVehPlaca = new java.lang.String(var1);
                String var2 = textAñoFabricacion.getValue().toString();
                java.math.BigDecimal pVehAño = new java.math.BigDecimal(var2);
                String var3 = textMarca.getValue().toString();
                java.lang.String pVehMarca = new java.lang.String(var3);
                String var4 = textModelo.getValue().toString();
                java.lang.String pVehModelo = new java.lang.String(var4);
                String var5 = textCilindrada.getValue().toString();
                java.math.BigDecimal pVehCilin = new java.math.BigDecimal(var5);
                String var6 = textChasis.getValue().toString();
                java.lang.String pVehChasis = new java.lang.String(var6);
                String var7= textNumMotor.getValue().toString();
                java.lang.String pVehMotor = new java.lang.String(var7);

                vehiculo.insertarVehiculo(pVehCod,pVehPromedioHrs,pVehPlaca,pVehAño,pVehMarca
                ,pVehModelo,pVehCilin,pVehChasis,pVehMotor,pVehOdomCod,pVehSgpCod,
                pVehCicCod,pVehKmIni,pVehLimOdorn,pVehFecAlt,usuario);
                info(mensaje.getURL("vehiculos.mensaje.infoInsertar"));
                getSessionBean1().setBusquedaVehiculo(null);
                getSessionBean1().setFilaSubGrupo(null);
                getSessionBean1().setDatosVehiculo(null);
            } else {
                warn("El año debe estar entre 1970 y el 2020!!");
            }
        } else{
            warn("La fecha de alta, no cumple el formato");
        }
    return null;
}

```

→

Llamada al método insertar del controlador vehículo

En el anterior código se puede ver la construcción de un formulario jsp para el registro de vehículos en el cual se aprecia la llamada al controlador vehículo, de manera que al llamar al método insertar vehículo a través del controlador este llamará al método para poder ser utilizado. En caso de que hubiesen cambios en la parte del controlador solo tendrá que modificarse el método del controlador siendo así más fácil el cambio sin afectar la parte del diseño de la interfaz, el patrón MVC nos separa independientemente de una manera sencilla la parte de la lógica de negocios, la vista y el controlador con la finalidad de conseguir un mantenimiento más sencillo en las aplicaciones y nos ahorra el tiempo y hacer más sencilla la programación.



llamada al Controlador vehículo dentro del formulario de registro de vehículos

```
private Cadet vehiculos controladores CtrlVehiculoRemote lookupCtrlVehiculoBean() {
    try {
        javax.naming.Context c = new javax.naming.InitialContext();
        Object remote = c.lookup("ejb/CtrlVehiculoBean");
        Cadet.vehiculos controladores CtrlVehiculoRemote rv = (Cadet.vehiculos controladores CtrlVehiculoRemote) javax.rmi.PortableRemoteObject.narrow(remote, Cadet.vehiculos.controladores.CtrlVehiculoRemoteHome.class);
        return rv.create();
    } catch(javax.naming.NamingException ne) {
        java.util.logging.Logger.getLogger(getClass().getName()).log(java.util.logging.Level.SEVERE, "exception caught", ne);
        throw new RuntimeException(ne);
    } catch(javax.ejb.CreateException ce) {
        java.util.logging.Logger.getLogger(getClass().getName()).log(java.util.logging.Level.SEVERE, "exception caught", ce);
        throw new RuntimeException(ce);
    } catch(java.rmi.RemoteException re) {
        java.util.logging.Logger.getLogger(getClass().getName()).log(java.util.logging.Level.SEVERE, "exception caught", re);
        throw new RuntimeException(re);
    }
}
```



Controlador Vehículo

método insertar

```
public void insertarVehiculo(BigDecimal pVehCod, BigDecimal pVehProHrs, String pVehPlaca, BigDecimal pVehAno, String pVehMarca, String pVehModelo, BigDecimal pVehCilin, String pVehChasis, String pVehMotor, BigDecimal pVehOdomCod, BigDecimal pVehSgrpCod, BigDecimal pVehCctCod, BigDecimal pVehKmln, BigDecimal pVehLmOdom, java.sql.Date pVehFecAlt, String pVehUsrCod) {
    Date fechaSistema = new Date();
    java.sql.Date fechaModificacion = new java.sql.Date(fechaSistema.getTime());
    BigDecimal estado = new BigDecimal(!);
    VehprmodomLocal odometro,
    VehprmsubgLocal subGrupo;
    try{
        Vehiculo findByPrimaryKey(pVehCod),
        throw new EJBException("El vehículo ya existe");
    }catch(FinderException fecep){
        try{ odometro = Odometro findByPrimaryKey(pVehOdomCod);
        }catch(FinderException fe){
            throw new EJBException("El odómetro no existe");
        }
    }try{
        subGrupo = SubGrupo findByPrimaryKey(pVehSgrpCod);
    }catch(FinderException fe){
        throw new EJBException("El subgrupo no existe");
    }
    try{ Collection c = Vehiculo findByNumeroPlaca(pVehPlaca+"%");
    if(c.size()==0){ Vehiculo create(pVehCod, pVehAno, pVehModelo, pVehMarca, pVehChasis, pVehCilin, pVehMotor, pVehPlaca, pVehCctCod,
        pVehKmln, pVehLmOdom, pVehProHrs, pVehFecAlt, null, pVehUsrCod, fechaModificacion, estado, odometro, subGrupo);
    }else{ throw new EJBException("Ya existe un vehículo registrado con la placa introducida"); }
    }catch(Exception e){
        System.out.println("Error al insertar el vehículo: "+e.getMessage());
        throw new EJBException(e.getMessage());
    }
}
```

3.3.4.2. Diagrama de Clases

A partir del modelo conceptual y los diagramas de colaboración se elaborará el diagrama de clases. solamente mostraremos las clases que han ido apareciendo en los diagramas de colaboración y se mantendrán las asociaciones y atributos existentes en el modelo conceptual incluyendo además todos los métodos que hayan ido apareciendo en los diagramas de colaboración.

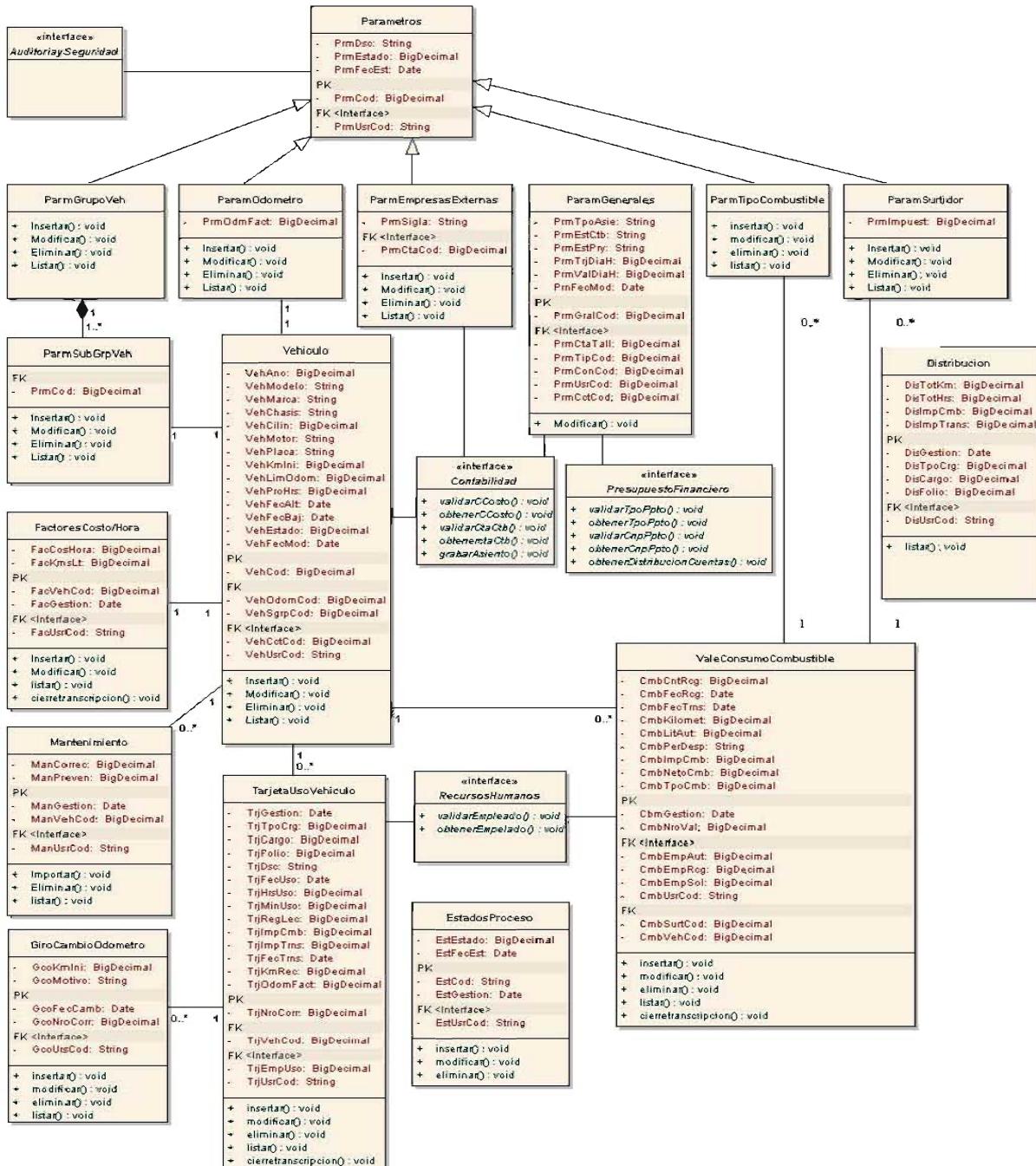


Figura 3.16 Diagrama de Clases

Sistema de Control y Seguimiento del Uso de Vehículos Vía Web

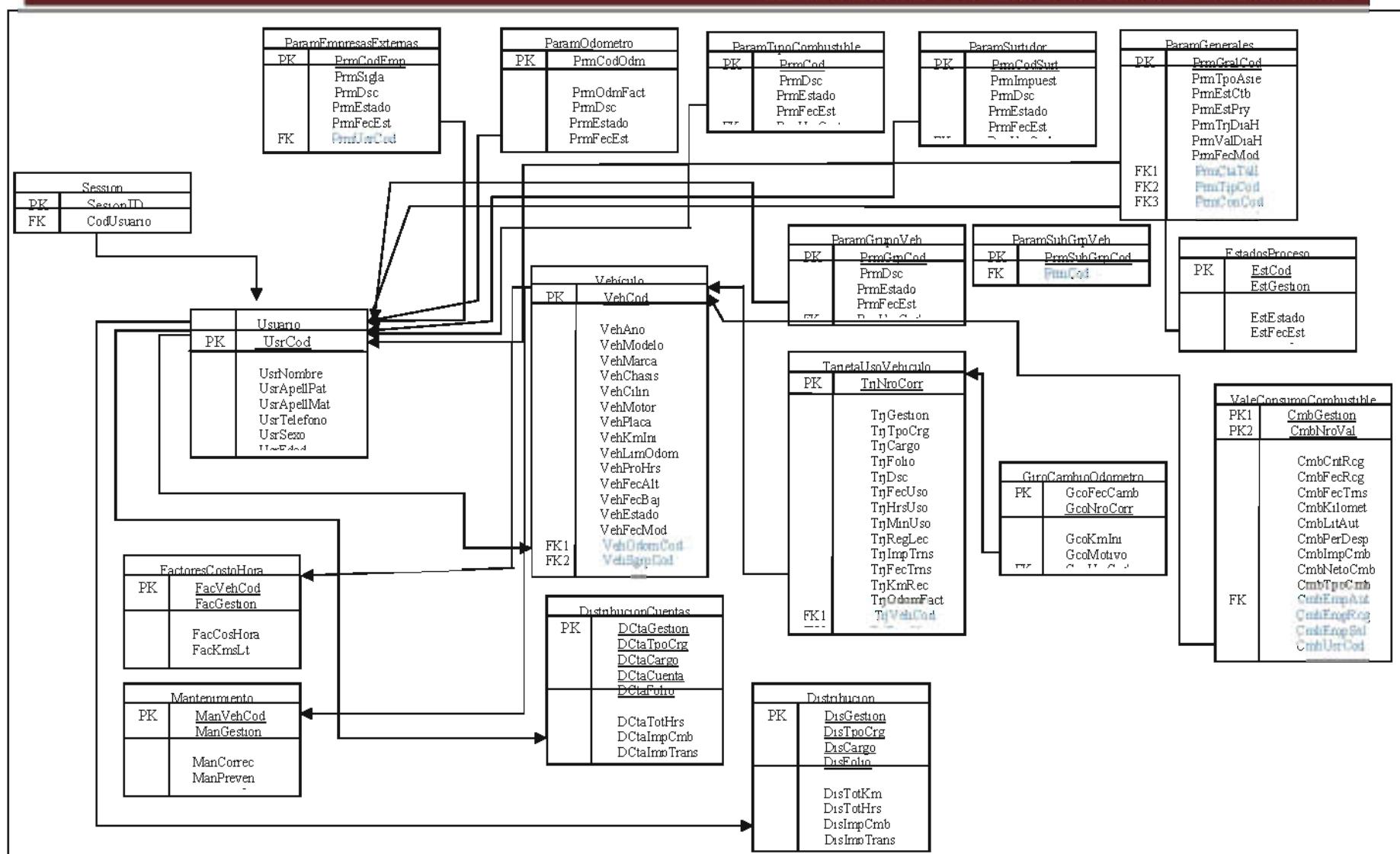


Figura 3.17 Modelo Objeto Relacional

3.3.5. Implementación

3.3.5.1. Diagrama de Componentes

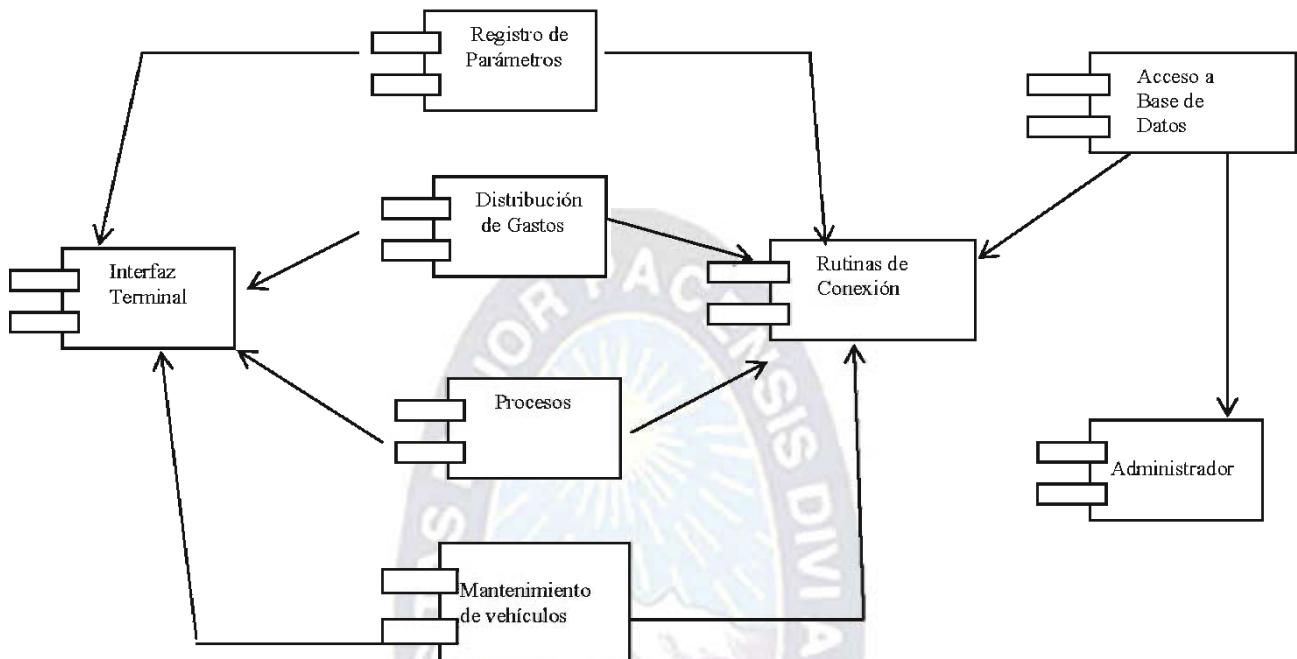


Figura 3.18. Diagrama de Componentes

3.3.5.2. Diagrama de Despliegue

Mediante el diagrama de despliegue podemos ver la arquitectura en el modelo de tres capas inmerso con el patrón MVC el cual maneja vista, controlador, modelo (base de datos).

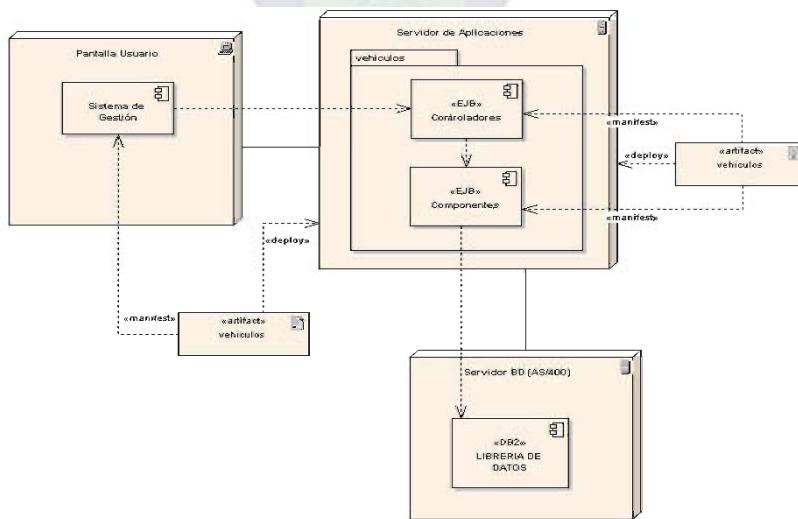


Figura 3.19. Diagrama de Despliegue

3.3.6. Pruebas

3.3.6.1. Caja Blanca

La prueba de camino básico es una técnica de prueba de caja blanca que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental.

a) Complejidad ciclomática

Es una métrica de software que proporciona una medición cuantitativa de la complejidad lógica del programa.

La complejidad ciclomática está basada en la teoría de grafos y nos da una métrica de software extremadamente útil.

La complejidad se puede calcular de tres formas:

- El numero de regiones del grafo.
- La complejidad ciclomática $V(G)$, de un grafo de flujo G se define como.

$$V(G) = A - N + 2$$

Donde:

A es el numero de aristas del grafo de flujo.

N es el número de nodos del mismo.

- La complejidad ciclomática, $V(G)$ de un grafo de flujo G también se define como.

$$V(G) = P + 1$$

Donde:

P es el número de nodos predicable contenido en el grafo de flujo G.

A continuación vemos la complejidad ciclomática del registro del uso de vehículos véase la figura 3.19

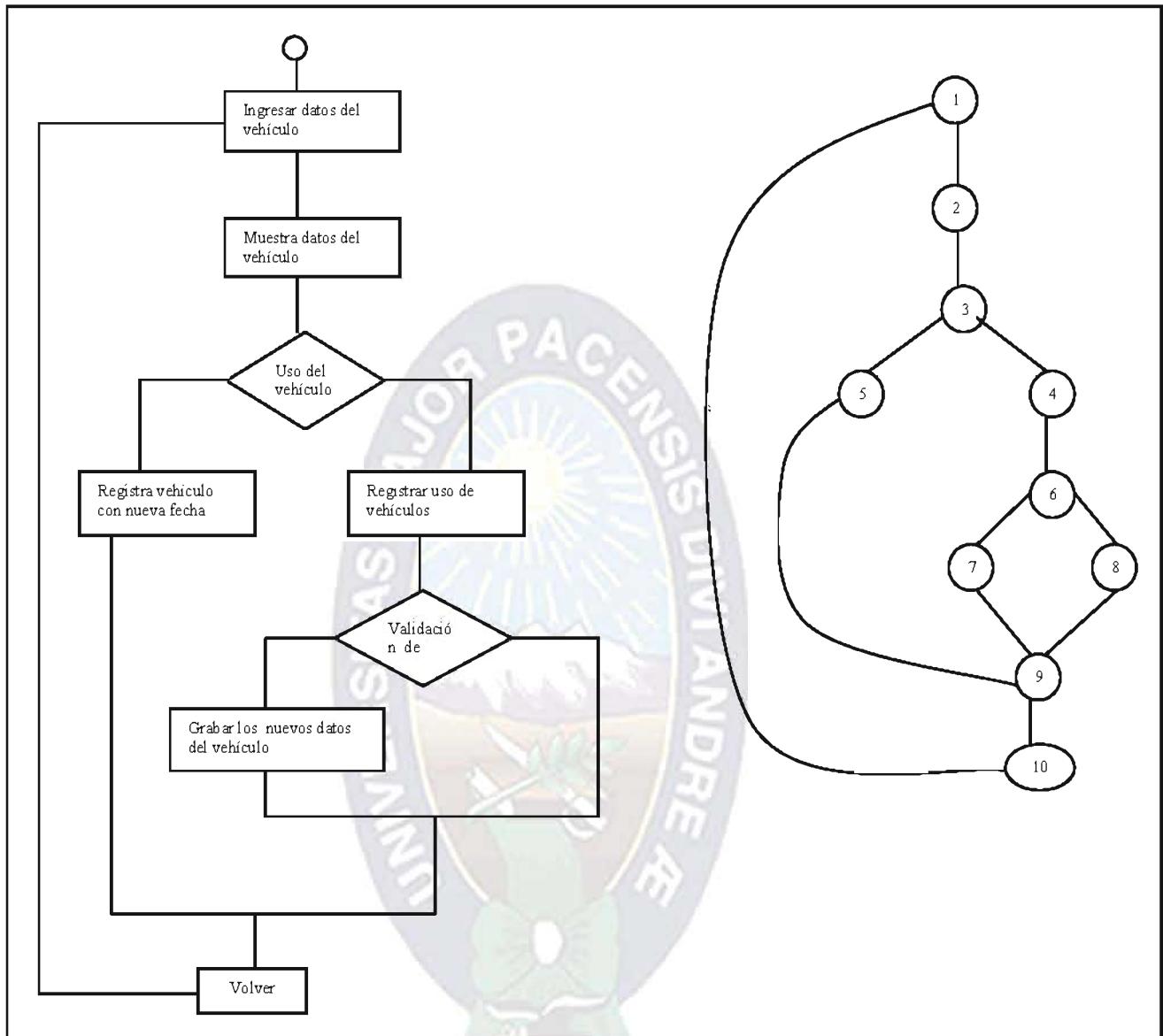


Figura 3.20 Grafo complejidad Ciclomática

Remplazando el valor del numero en la segunda formula se obtiene.

$$V(G) = 12 - 10 + 2 = 4$$

Este valor determina cuatro caminos independientes y cuatro casos de prueba para el modulo de Uso de Vehículos. Por tanto se realiza las pruebas necesarias para cada uno de los caminos encontrados como se muestra en la tabla.

Tabla 3.23. Caminos de la complejidad ciclomática

Camino 1	Camino2	Camino3	Camino4
1.Ingresa datos del vehículo 2. Mostrar datos del vehículo 3. Se usa el vehículo 4.Registra vehículo con nueva fecha 10.Volver a ingresar datos	1.Ingresa datos del vehículo 2.Mostrar datos del vehículo 3.Se usa el vehículo 4.Registra uso de vehículo 6. Valida datos con otros módulos. 7. registra vehículo 10.Volver a ingresar datos	1.Ingresa datos del vehículo 2.Mostrar datos del vehículo 3.Se usa el vehículo 4.Registra uso de vehículo 6. Valida datos con otros módulos. 8. Existe validación con otros módulos. 9.Volver a ingresar dato	1.Ingresa datos del vehículo 2.Mostrar datos del vehículo 3.Se usa el vehículo 4.Registra uso de vehículo 6. Valida datos con otros módulos. 8. Existe validación con otros módulos. 10.Volver a ingresar dato

3.3.6.2. Caja Negra

También denominada prueba de comportamiento, se centra en los requisitos funcionales del sistema y el probarlo se limitara a suministrarle datos de entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el modulo por dentro.

a) Análisis de valores de Límites

Los errores tienden a darse con mayor frecuencia en los límites del campo de entrada por ello se utilizara esta técnica de prueba que nos llevara a una elección de casos de prueba que ejerciten los valores límite.



Figura 3.21 Modelo de Prueba de Caja Negra

Las pruebas que se pueden realizar sobre el caso de uso “Registro del uso de Vehículos” para comprobar si la funcionalidad es la correcta. A continuación veamos el caso de prueba en la siguiente tabla 3.24 caso de prueba registro del uso de vehículos.

Tabla 3.24. Caso de Prueba Registro del uso de Vehículos

Descripción	El Jefe de Talleres hace una consulta sobre el uso de los vehículos que fueron utilizados en algún trabajo. El sistema muestra el detalle de los vehículos que estuvieron en uso.
Condición de Ejecución	Las condición de ejecución del caso de prueba es que el jefe de talleres este dado de alta en la base de datos de usuario para poder ingresar al sistema.
Entrada	<ul style="list-style-type: none">▪ Se introduce en el campo de usuario, al usuario ‘Mario’.▪ Se introduce en el campo de contraseña ‘mario321’.▪ Se pulsa el botón ‘inicio de sesión’▪ Aparece la interfaz de Usuario.▪ Se despliega en forma de árbol las opciones en las cuales trabajara el jefe de talleres.▪ Se seleccionara el uso de vehículos del árbol.▪ El sistema mostrara el detalle de todos los vehículos que estuvieron en uso.▪ Se ingresara los datos correspondientes.▪ La información se guardada.
Resultado Esperado	El sistema muestra los vehículos que han sido utilizados para algún trabajo de campo.
Evaluación de la Prueba	Prueba superada con éxito.

3.3.6.3. Análisis de Resultados

El análisis de resultados sobre el rendimiento en el tiempo de respuesta del Sistema de Control y Seguimiento del uso de Vehículos vía Web para la empresa CADEB se lo detalla en la tabla 3.25.

Tabla 3.25. Análisis de Resultados

Proceso	Antes	Ahora
Distribución	Realiza la distribución de gastos de consumo de combustible de 30 a 40 minutos.	La distribución de gastos se lo realiza de en 5 minutos
Registro	El registro de los parámetros se realiza de 25 a 30 segundos.	El registro de los parámetros se lo realiza por la internet lo que disminuye el tiempo de grabación, aproximadamente de 5 a 6 segundos.
Carga de tarjetas	Realiza la carga de datos de tarjetas de 20 a 40 minutos	La carga de datos de tarjetas se lo realiza en 5 minutos
Gastos de mantenimiento	Realiza la distribución de gastos de mantenimiento de vehículos de 30 a 40 minutos.	La distribución de gastos de mantenimiento se lo realiza en 5 minutos.
Mantenimiento	Realiza la carga de datos de mantenimiento de manera manual lo cual emplea el tiempo de 30 min.	Se realiza la carga de datos de mantenimiento desde el sistema en 1 min.
Reportes	La generación de reportes tarda 5 minutos.	La generación de reportes se lo realiza en 5 segundos.
Consultas	El jefe de talleres no podía consultar directamente sobre el mantenimiento de los vehículos se recibe la información por separado.	El jefe de talleres puede consultar sobre los vehículos que están en mantenimiento desde el sistema.
No existía	No existía los siguientes reportes como ser: Reporte de consumo de combustible por rango de fechas Reportes de consumo de combustible por gestión Reporte costo combustible y mantenimiento Reportes consumo combustible por vale Reportes consumo combustible por vehículos Reportes tarjetas del uso de los vehículos Reportes costo de repuestos Reportes costo de mano de obra Reportes costo de mantenimiento	se tienen los nuevos reportes los cuales se generan en 10 segundos.

3.3.6.4. Evaluación:

La construcción de esta primera versión del software fue considerada como satisfactoria, pues se llegó a implementar los requisitos deseados por el cliente, además de haber logrado desarrollar con éxito el modulo de vehículos realizando pruebas y verificando resultados finales, los cuales fueron óptimos.

3.3.7. Diagrama Navegacional

El diagrama Navegacional del Sistema de Control y Seguimiento del uso de Vehículos Vía Web para la empresa CABED al estar orientada a un diseño Web presenta el diagrama navegacional de los usuarios que utilizaran el sistema.

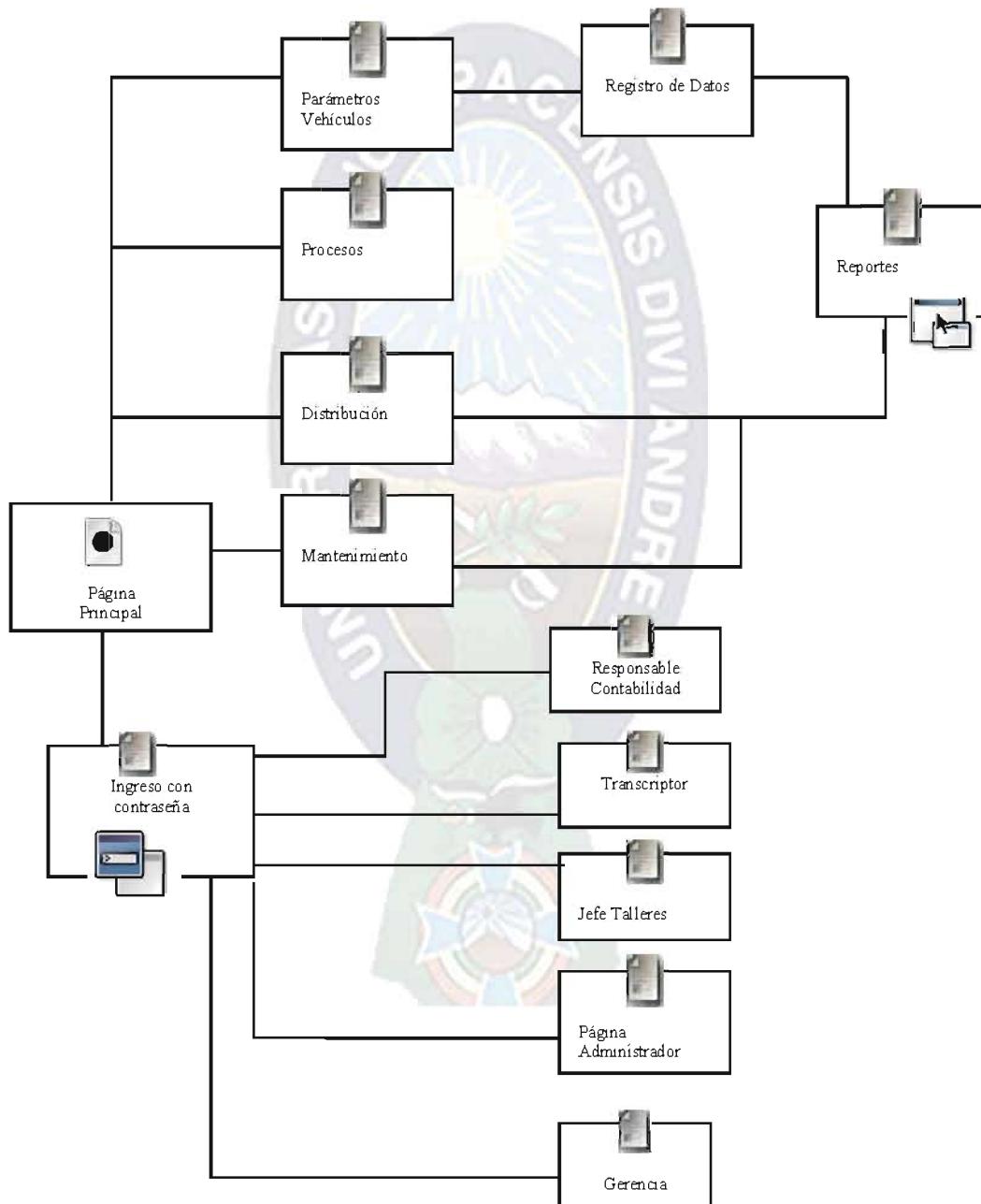


Figura 3.22. Diagrama Navegacional del Sistema

3.3.8. Descripción de Pantallas

El diseño de interfaz de usuario, crea un medio de comunicación entre el hombre - máquina o denominado usuario, se debe tomar en cuenta que la interfaz debe de ser amigable, que permita al usuario realizar las diferentes acciones.

A continuación se mostrara algunas pantallas del sistema de control y seguimiento del uso de vehículos vía web.

La Figura 3.23 es la pantalla de ingreso al sistema en la cual el usuario ingresa con su nombre de usuario y su correspondiente contraseña.

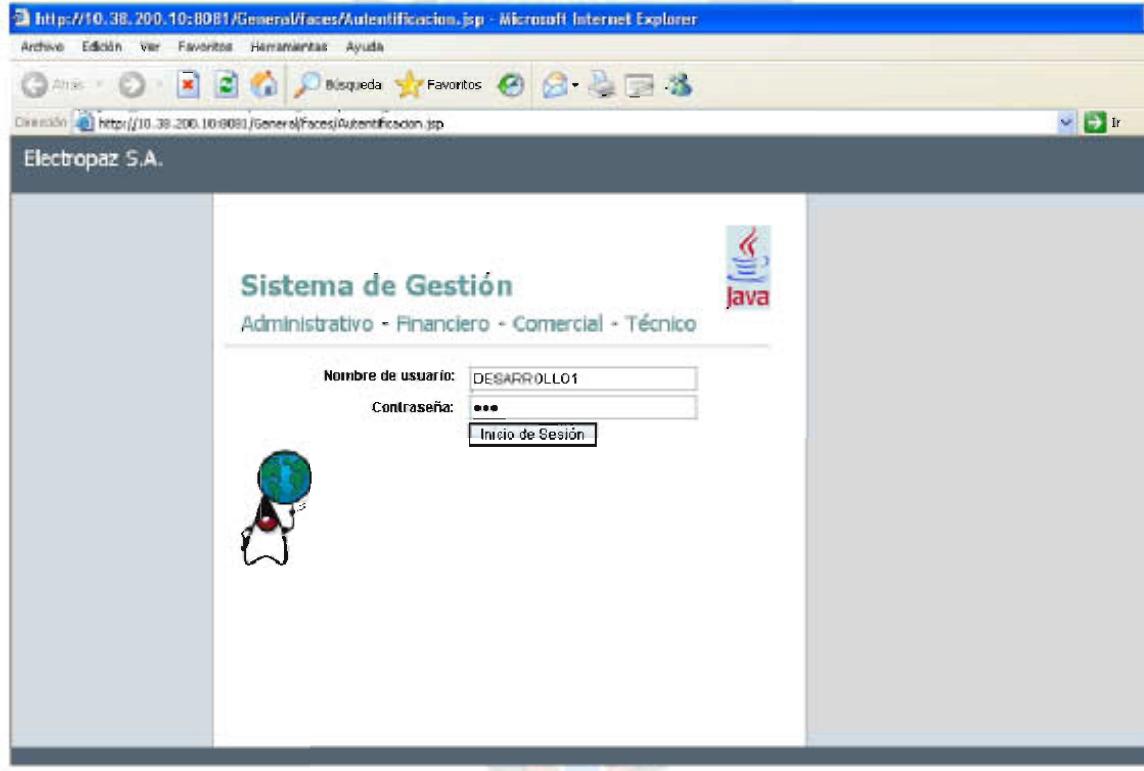


Figura 3.23 Pantalla de Ingreso al sistema – registro Login y Password

Sistema de Control y Seguimiento del Uso de Vehículos Vía Web

La figura 3.24 muestra la pantalla donde se carga el árbol en el cual se encuentra el módulo de vehículos, se puede apreciar el listado principal de todos los vehículos con las siguientes opciones: insertar, modificar, eliminar, imprimir, actualizar y la búsqueda que se la realizará por código, placa o estado.

Vehículos

Nro.	Número placa	Fecha alta	Fecha baja	Grupo - Sub grupo	Centro costo	Ot.
24	1093FCL	25/11/1997		CAMIONETA 4 X 4 ESTANDAR	411	IQL
25	1093FDR	25/12/1997		CAMIONETA FORD F-150	411	IQL
26	1093FEU	25/12/1997		CAMIONETA 4 X 4 ESTANDAR	411	IQL
27	1093FBH	13/04/1998		CAMIONETA FORD E-150	411	IQL
28	1093UBI	25/08/1998		CAMIONETA FORD F-150	411	IQL
29	1093HSA	25/08/1998		CAMIONETA FORD F-150	411	IQL
31	666 KSRAS#	25/11/2002		CAMIONETA FORD F-350 4 X 4	411	IQL
32	666 LYH	25/11/1998		CAMIONETA FORD F-350 4 X 4	411	IQL
33	1097ATT	25/01/2001		CAMIONETA FORD F-350 4 X 4	411	IQL
34	923 PCI	25/02/1998		CAMIONETA 4 X 2 ESTANDAR	411	IQL
35	1002AFN	25/05/1998		CAMIONETA FORD RANGER	411	IQL
36	1091BXI	25/08/1998		CAMIONETA FORD F-350 4 X 4	411	IQL

Buttons at the bottom: Insertar, Modificar, Eliminar, Imprimir, Actualizar.

Figura 3.24 Pantalla del Listado general de los vehículos

La Figura 3.25 muestra la pantalla principal en donde se registrara a los vehículos nuevos los cuales serán luego dados de alta.

Insertar Vehículo

Fields include:

- * Código: [Input]
- * Número de placa: [Input]
- * Año de fabricación: [Input]
- * Fecha de alta: [Input] (with calendar icon)
- * Marca: [Input]
- * Modelo: [Input]
- * Cilindrada: [Input]
- * Odómetro: [Input] (set to 'nuevos odómetros')
- * Sub - grupo: [Input]
- * Centro de costo: [Input]
- * Clasif.: [Input]
- * Número de motor: [Input]

Buttons at the bottom: Confirmar, ATRÁS.

Figura 3.25 Pantalla de registro de Vehículos

Sistema de Control y Seguimiento del Uso de Vehículos Vía Web

La Figura 3.26 muestra la pantalla de procesos en el cual tenemos la opción de factores de distribución la cual realizará y verifica el costo por hora del uso de los vehículos teniendo en cuenta el kilometraje de cada vehículos que luego se procesará.

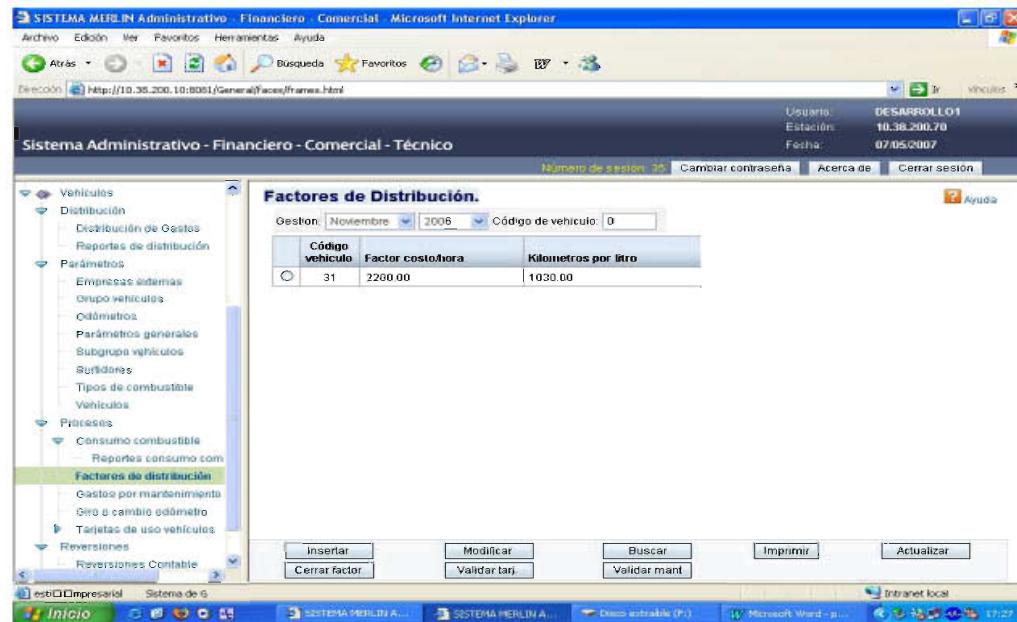


Figura 3.26 Pantalla de Factores de Distribución

La Figura 3.27 muestra la pantalla de reportes de los vehículos donde se observa a todos los vehículos que fueron dados de alta y que pueden ser utilizados.

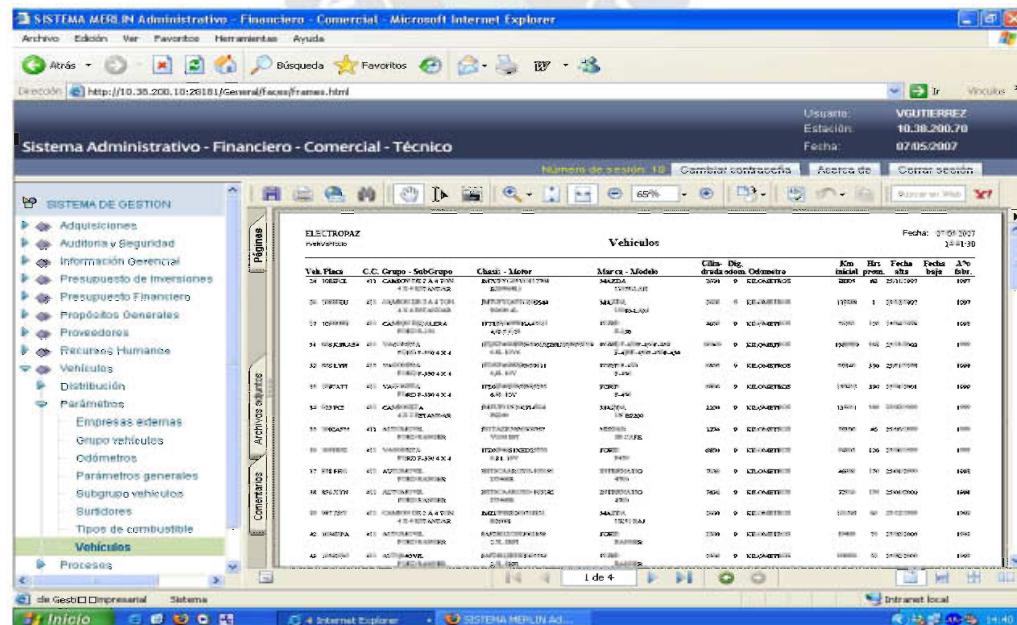


Figura 3.27 Pantalla de visualización de reportes de Vehículos

3.4. FASE DE TRANSICIÓN

Durante esta fase, se hace una descripción de la evolución del sistema las actividades realizadas para la puesta en marcha, así como la capacitación y operación del mismo.

Durante el análisis se estableció que el software debe apoyar las funciones del jefe de Talleres del área de CADE Talleres, mostrando la información necesaria que este requiera, una vez refinado y aprobado el sistema, el mismo que será instalado en el área de CADE Talleres.

3.4.1. Implementación del Sistema:

La implementación del sistema se la realizará una vez liberado el producto, se espera que surjan detalles que requieran el desarrollo de nuevas versiones, las correcciones de ciertos problemas que sean no tanto de fondo si no de forma.

3.4.2. Instalación del Sistema:

El nuevo sistema no necesita ser instalado, se ingresara mediante el internet explorer, mozilla u otros navegadores, teniendo previamente la dirección de la pagina web en la intranet de la empresa, el Jefe de Talleres podrá ingresar para realizar las pruebas finales de funcionamiento, antes de ser entregado como producto final a los usuarios del mismo tal como sea había planificado en las primeras fases en donde se han recogido los requisitos para el desarrollo del proyecto.



**Métricas de
Calidad**

CAPÍTULO 4

MÉTRICAS DE CALIDAD

Las métricas de calidad que se utilizan para la medición de un software, son importantes ya que mediante ellas se puede testear la calidad de un software [PRESSMAN,2003]. Se puede mencionar que existen diferentes factores de calidad de software el factor a utilizar en el presente proyecto es el estándar ISO 9126 que identifica los atributos claves de calidad para el software los cuales son:

- Funcionalidad
- Confabilidad
- Eficiencia
- Facilidad de mantenimiento
- Portabilidad

4.1. Funcionalidad

Se entiende por funcionalidad el grado que el software satisface las necesidades funcionales.

4.1.1. Métricas basadas en el Punto Función

La métrica punto función se usa como medio para predecir el tamaño del sistema que se va obtener por medio del análisis.

➤ Número de Entradas de Usuarios

Es el número de entradas que proporciona datos al sistema con el fin de poder realizar las distintas operaciones (Altas, Bajas, Modificaciones).

➤ Número de Salidas de Usuario

Una Salida de usuario es aquella que proporciona información orientada al sistema. Las salidas de usuario se refieren a informes pantallas, mensajes de error.

➤ Número de peticiones de Usuario

Es una entrada interactiva que produce la generación de alguna respuesta de software en forma de salida interactiva.

➤ **Numero de Archivos.**

Es un grupo lógico de datos que puede ser parte de una gran base de datos o de un archivo independiente.

➤ **Numero de Interfaces Externas**

Es el numero de interfaces legibles por la maquina que se utilizan para trasmisir información a otra máquina.

Tamaño del Sistema

Entradas de Usuario	<ul style="list-style-type: none">✓ Registro de Empresas Externas✓ Registro de Grupo Vehículos✓ Registro de Odómetro✓ Registro de Parámetros Generales✓ Registro de Subgrupo Vehículos✓ Registro de surtidores✓ Registro de Tipos de Combustible✓ Registro de Vehículos✓ Registro de Factores Costo/hora✓ Registro de mantenimiento
Salidas de Usuario	<ul style="list-style-type: none">✓ Reportes de factores de distribución✓ Reporte de Tarjetas de uso de vehículos✓ Reporte de detalle de Uso de Vehículos✓ Reporte de detalle de Uso de Vehículos por gestión✓ Reporte de detalle de Uso de Vehículos por fechas✓ Reporte de mantenimiento de Vehículos✓ Reporte de Costos de Combustible✓ Reporte de consumo de Combustible✓ Detalle de consumo de combustible por vehículo✓ Reporte Detalle de pedido de combustible por empleado✓ Reporte de vehículos✓ Reporte de resumen de distribución del uso de vehículos✓ Detalle de distribución del uso de vehículos✓ Reporte de Tipos de combustible✓ Reportes Surtidores, Grupo, Subgrupo✓ Resumen de Distribución General de Uso de Vehículos por Tipo de Cargo

Peticiones de Usuario	<ul style="list-style-type: none"> ✓ Listado de vehículos ✓ Listado de odómetros ✓ Visualización de los factores costo/hora ✓ Listado del mantenimiento vehículos ✓ Listado de Vehículos por Gestión ✓ Listado de consumo de combustible ✓ Listado de Grupos de Vehículos ✓ Listado de subgrupo de vehículos ✓ Visualización de la distribución ✓ Listado de gastos por mantenimiento
Número de Archivos	Tablas de la base de datos 16
Interfaces Externas	Disco impresoras Backups Conexión de red.

Una vez recopilados los datos anteriores, a continuación se realiza el cálculo de punto función, asignando un factor de ponderación a los parámetros de medición ya descritos.

Tabla 4.1. Parámetros de Medición

Parámetros de Medición	Cuenta	Factor de Ponderación (Medio)	Total
Número de entradas de usuario	10	4	40
Número de salidas de usuario	16	5	80
Número de peticiones de usuario	10	4	40
Número de archivos	16	10	160
Número de interfaces externas	4	7	28
Cuenta Total:			348

Para calcular el punto función (PF), se utiliza la siguiente relación:

$$PF = \text{Cuenta Total} * [0.65 + 0.01 * \sum F_i] \quad (1)$$

Donde:

Cuenta Total: es la suma de todas las entradas PF obtenidas en la tabla 4.1.

F_i ($i = 1$ a 14): son valores de ajuste de complejidad.

0.65: Confiabilidad del sistema.

0.01: Error de confiabilidad del sistema.

Los valores de ajuste de complejidad, se evalúan en base a una escala de ajuste vista en la tabla 4.2.

Tabla 4.2. Escala de Ajuste

Descripción	Escala
No influencia	0
Incidencia	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Los valores de ajuste de complejidad se obtienen respondiendo a las preguntas destacables que se muestra en la tabla 4.3. Aplicando los valores de la tabla de escala de ajuste tabla 4.2

Tabla 4.3. Valores de Ajuste de complejidad

Valores de Ajuste(Preguntas)	Valor
1.- ¿Requiere el sistema copias de seguridad y de recuperación fiable?	5
2.- ¿Se requiere comunicación de datos?	5
3.- ¿Existen funciones procesos distribuidos?	3
4.- ¿Es crítico el rendimiento?	2
5.- ¿Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?	5
6.- ¿Requiere el sistema la entrada de datos interactiva?	4
7.- ¿Requiere la entrada de datos interactiva que las transacciones se lleven a cabo sobre múltiples pantallas u operaciones?	2
8.- ¿Se actualiza los archivos maestros de forma interactiva?	4
9.- ¿Son complejas las entradas, las salidas, los archivos o las peticiones?	2
10.- ¿Es complejo el procesamiento interno?	3
11.- ¿Se ha diseñado el código para ser reutilizable?	4
12.- ¿Están incluidos en el diseño la conversión y la instalación?	4
13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	5
14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?	5
$\sum F_i = \text{Total}$	53

Una vez obtenido el valor de ajuste y reemplazando en la ecuación (1) se tiene:

$$PF = \text{Cuenta Total} * [0.65 + 0.01 * \sum F_i]$$

$$PF = 348 * [0.65 + 0.01 * 53] = 410.64$$

Ajustando a la curva normal:

$$\text{Funcionalidad} = PF / PF \text{ M\'aximo}$$

Donde:

$$PF \text{ M\'aximo} = \text{Cuenta Total} * [0.65 + 0.01 * \sum F_i] ; \text{ se toma el valor m\'aximo de } \sum F_i$$

$$[i = 1 \text{ a } 14], [0 \leq \sum F_i] \leq 70$$

Reemplazando se tiene:

$$PF \text{ M\'aximo} = \text{Cuenta total} * [0.65 + 0.01 * \sum F_i]$$

$$PF \text{ M\'aximo} = 348 * [0.65 + 0.01 * 70] = 469.80$$

Por Tanto:

$$\begin{aligned} \text{Funcionalidad} &= (PF / PF \text{ M\'aximo}) * 100 = (410.64 / 469.80) * 100 \\ &= 87.40 \end{aligned}$$

La funcionalidad del sistema es de: 87%

4.2. Confidabilidad

La confidabilidad es la cantidad tiempo que el software est\'a disponible para su uso, es decir, la cantidad de tiempo que el sistema se encuentra en funcionamiento dentro de la instituci\'on y est\'e libre de fallas ; se puede considerar que a mayor n\'umero de fallas, menor confidabilidad, pero a menor n\'umero de fallas mayor ser\'a la confidabilidad.

Para poder medir la confidabilidad del sistema, se aplico la teor\'ia estad\'istica, el cual nos permitir\'a calcular el porcentaje de confidabilidad, para esto se tomaron las siguientes consideraciones:

- Supongamos que el sistema comienza a trabajar en el instante $t = 0$ y se observa hasta que falle, lo cual ocurre en el instante t .
- Designamos una variable aleatoria T , que representa la duraci\'on del tiempo de trabajo del sistema sin fallas, entonces la probabilidad de que el sistema falle en un tiempo t ser\'a: $P[T \leq t]$ y la probabilidad de funcionamiento del sistema ser\'a: $P[T > t] = 1 - F(t)$.

- Para calcular la confiabilidad $R(t)$ del sistema, se tomara en cuenta un periodo t que será el periodo de prueba hasta que se produzca la falla, para esto aplicaremos la función exponencial y considerando el valor del punto función calculado en la anterior sección.

$$F(t) = PF * e^{-\lambda t}; \text{ con } x > 0$$

- El margen de error de Λ es de un décimo, significa que calculando el error durante 10 ejecuciones en un mes obtendremos un valor probable de fallas que puede tener el sistema.
- Si calculamos para una gestión de un año es decir 12 meses aplicando el valor Punto Función $PF = 0.874$ obtenido en la sección anterior.

A continuación se realizara el cálculo de la probabilidad de que el sistema tenga fallas:

$$P[T \leq t] = F(t)$$

$$F(t) = PF * e^{-\lambda t}; \text{ con } x > 0$$

$$F(t) = 0.874 * (\text{Exp}(-1/10 * 12))$$

Para $t = 12$

$$F(12) = 0.29 = 29\%$$

Calculo de probabilidad del sistema sin fallas:

$$P[T \leq t] = 1 - F(t)$$

$$F(t) = PF * e^{-\lambda t}; \text{ con } x > 0$$

$$F(t) = 1 - [0.874 * (\text{Exp}(-1/10 * 12))]$$

Para $t = 12$

$$F(12) = 1 - 0.29 = 0.71 = 71\%$$

Por tanto se puede decir que el sistema tiene un grado de confiabilidad del 71%, y que seguirá funcionando en un año (12 meses).

4.3. Facilidad de Mantenimiento

La facilidad de mantenimiento, es la facilidad con que una modificación puede ser realizada en el sistema. Las modificaciones pueden incluir correcciones, mejoras o adaptación del sistema a cambios en el entorno, y especificaciones de requerimientos funcionales [PRESSMAN, 2003].

Capacidad de ser analizado: la capacidad del producto de software para atenerse a diagnósticos de deficiencias o causas de fallas en el sistema o la identificación de las partes a ser

modificadas , por lo cual se fue analizando e identificando en el proceso de construcción del sistema las posibles fallas luego identificadas para poder ser modificadas.

Cambiabilidad: la capacidad del software para permitir que una determinada modificación sea implementada. En el proceso de desarrollo del sistema se fue realizando diferentes modificaciones, hasta que el sistema tome la madurez adecuada en cada etapa

Estabilidad: la capacidad del producto de software para evitar efectos inesperados debido a modificaciones del software. Es necesario que el sistema tenga estabilidad por las modificaciones que se hicieron, por esto fue necesario tomar en cuenta la estabilidad del sistema para que pueda adecuarse a los cambios evitando efectos inesperados.

Tomando en cuenta los anteriores conceptos, se observa que el software fue tomando la madurez adecuada para tener un buen producto. En este sentido se tomo en cuenta el estándar IEEE 9824-1998[IEE94] sugiere un índice de madurez de software (IMS) que proporciona un indicación de estabilidad de un producto software (basado en los cambios que ocurren con cada versión del producto) [PRESSMAN, 2003].

La ecuación que nos ayudara a calcular el índice de madurez es la siguiente:

$$IMS = [M_T - (F_a + F_c + F_d)] / M_T$$

Donde:

M_T = Número de módulos de la versión actual.

F_a = Número de módulos en la versión actual que sean cambiado.

F_c = Número de módulos en la versión actual que sean añadido.

F_d = Número de módulos de la versión anterior que se han borrado en la versión actual.

Tabla 4.4. Indicador de estabilidad del Software

Versión del Sistema	M_T	F_a	F_c	F_d	IMS
Ver 1.0	3	3	0	0	0
Ver 1.1	4	2	1	0	0.25
Ver 1.3	5	2	1	0	0.4
Ver 1.4	7	0	1	0	0.85

4.4. Portabilidad

El esfuerzo de transportar o migrar un producto de una configuración hardware y/o software a otro está referido a las siguientes características.

Adaptabilidad:

la capacidad del producto de software para ser adaptado a diferentes entornos especificados sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerado. En cuanto a la plataforma el sistema está desarrollado para el sistema operativo Microsoft Windows, como ser Windows 98, Windows 2000 , Windows XP Profesional con el Servi Pack 1.Tener Instalado el navegador Explorer o mozilla para poder ver el sistema a través de Internet. También es posible la adaptabilidad en el sistema operativo Linux.

Facilidad de Instalación:

La capacidad del producto de software para ser instalado en un ambiente especificado. En cuanto al sistema este se realizó en Java, el cual permite que se tenga un acceso rápido a la aplicación, por cuanto no necesita de instalación pues solo se accede mediante Internet Explorer a una dirección (Servidor de Aplicaciones). En cuanto a las restricciones, se cuenta con Internet Explorer 6.0 o versiones superiores para tener acceso a la red del área de la empresa CADEB.

Coexistencia:

La capacidad del producto de software para coexistir con otros productos de software independientes dentro de un mismo entorno, compartiendo recursos comunes. El sistema coexistente con otros sistemas dentro de un mismo entorno y comparte ciertos recursos como base de datos de otro sistema.



Conclusiones y Recomendaciones

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Al terminar el presente proyecto, se observó que se realizaron todas las actividades propuestas al inicio y las conclusiones a las que se llegaron son las siguientes:

- A partir de los requerimientos que se obtuvieron se pudieron identificar los problemas del Área de CADE Talleres de los cuales se proponieron soluciones factibles que permitieron resolver los problemas planteados.
- La generación del Proceso de distribución de gastos de consumo de combustible y mantenimiento de los vehículos es más rápida y efectiva.
- La información recabada de los vehículos, el consumo de combustible y las tarjetas de uso de vehículos es realizada de manera automática.
- La información que se maneja es más detallada y rápida al realizar las diferentes consultas que se generan en los reportes de los vehículos.
- Se integro el área de mantenimiento al sistema de manera que se obtiene la información más a detalle de los vehículos que estuvieron en mantenimiento.
- El sistema del Control y Seguimiento del Uso de Vehículos es una herramienta con una interfaz de fácil manejo para el usuario, y permite consultar información con relación al Control y Seguimiento del uso de vehículos a partir de la información que se encuentra en la base de Datos.
- El sistema desarrollado es un aporte más para el área de CADE Talleres el cual logró llenar con buenas expectativas a la empresa CADEB y al área de CADE Talleres.

5.2. Recomendaciones

Habiendo concluido el proyecto es posible plantear las siguientes recomendaciones:

- Realizar el mantenimiento del software del sistema de manera periódica, para su buen funcionamiento.
- Para futuros trabajos desarrollar un modulo que integre el sistema actual con el área de mantenimiento de vehículos lo cual beneficiará de manera positiva en la empresa.
- Ampliar más el uso de la tecnología (J2EE) en aplicaciones web para que así se pueda realizar aplicaciones más complejas, empleando las nuevas técnicas que esta tecnología que va editando como ser nuevas versiones, mejorando también de esta manera los otros sistemas relacionados con el sistema actual.





REFERENCIAS BIBLIOGRÁFICAS

- [SCHUMULLER, 2001]** Josehp Schumuller.,Aprendiendo UML en 24 Horas, Pág. 230,Primera Edición, Prentice Hall, 2001
- [LARMAN, 1999]** Larman, C., UML y patrones, introducción al análisis de diseño orientada a objetos, Pág. 506, Prentice-Hall, Hispanoamerica 1999.
- [JACOB, 1999]** Jacobson/Booch/Rumbaugh, The Unified Software Development Process Pág. 210, Addison-Wesley,2000
- [PRESSMAN, 2005]** Pressman Rogers,2003,Ingenieria de Software un Enfoque Práctico Pág,17, quinta edición MGRAW-HILL Interamericana de España, Madrid España
- [FOWLER, 1999]** Martin Fowler,1999,UML gota a gota Pág. 22, tercera edición, Addison Wesley , Longman, de México S.A, DE C.V.
- [KEN, 1998]** Kendal &Kendal, 1998, Análisis y Diseño de Sistemas , Pág. 65, segunda Edición, México
- [LEYTOLD, 2003]** Programación en Java, quinta edición, Pág. 487, Editorial MGRAW-HILL, Madrid España
- Referencias Web:**
- [JUA03]** Juan Manuel Barrios Tutorial de J2EE(2003). Extraído en Septiembre 24.2007 Disponible en: <http://www.dcc.uchile.cl/~jbarrios/J2EE/MemoriaJ2EE.pdf>
- [MET03]** Ejemplos de Metodología RUP(2003). Extraído en Octubre 13.2007 Disponible en: <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/> - 29k

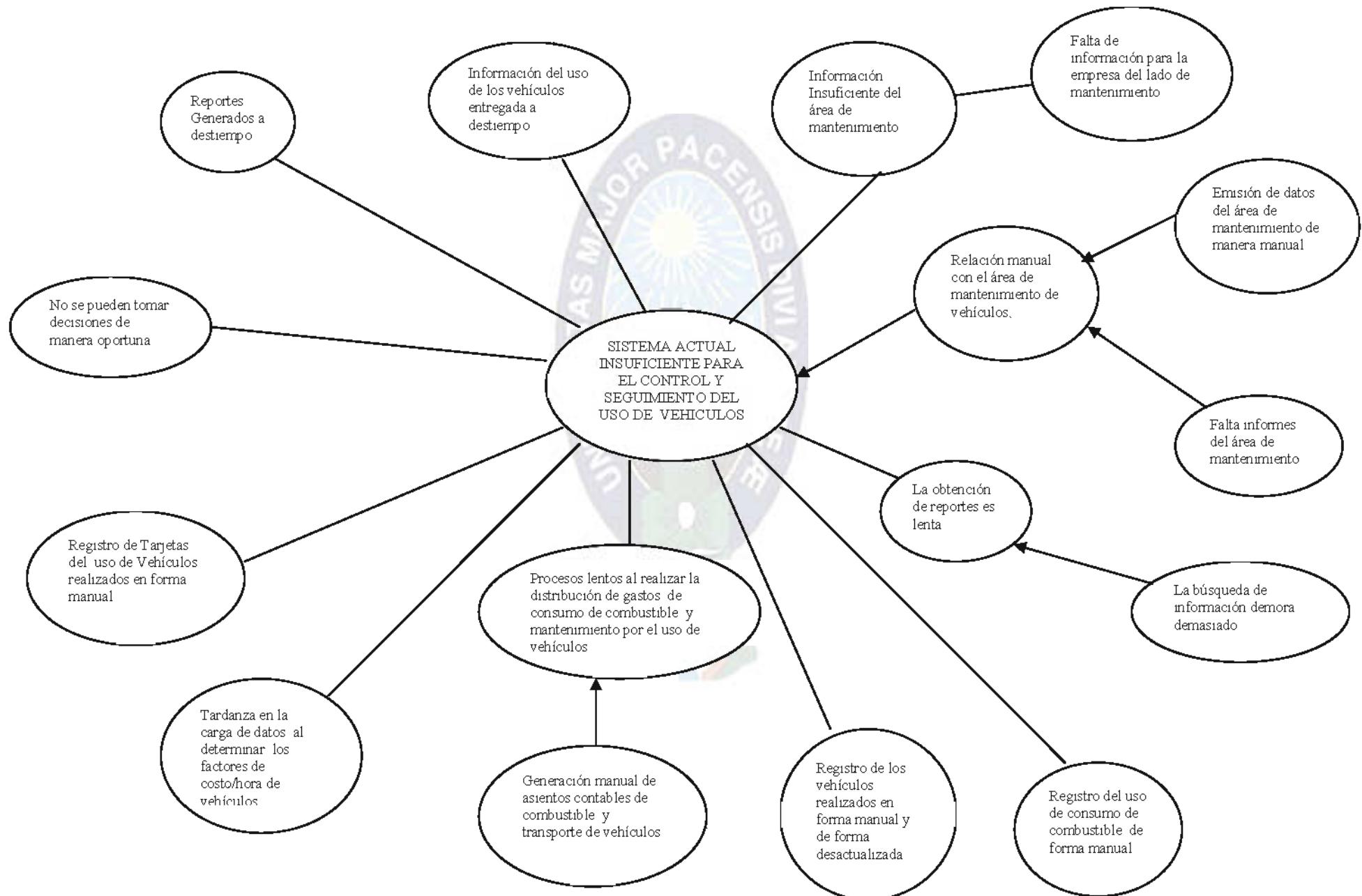
- [METD04]** Metodologías de Desarrollo de Software(2004). Extraído en Octubre 27.2007
Disponible en: www.informatizate.net/metodologias_de_desarrollo_de_software
- [WIK04]** Introducción al lenguaje Modelado de Unificado ejemplos de UML (2005).
Extraído en Septiembre 15.2007 Disponible en:
http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado
- [DEF04]** Java server faces Tecnology, introducción a la tecnología Java server faces(2005).extraído en Agosto 28.2007 Disponible en:
<http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htmldinternet>
- [PROG03]** Tutorial de Programación en java (2003).extraído en Septiembre 25 .2007
Disponible en: <http://www.programacion.com/tutorial/uml/2/>
- [EVANS, 2006]** Simplified Approach to RUP, Tutorial del RUP, extraído en Enero del 2008
http://www.therationaledge.com/content/jan_01/t_rup_ge.html
- [SUN, 2009]** Java Technology Overview, manual de programación en java, extraído 01.2007
disponible en: <http://www.sun.com/download/manuals>
- [SOM95]** Sommerville,1995 Ingeniería de software disponible en
<http://delta.cs.cinvestav.mx/~pmelia/softeng/cap32.ppt>
- [NETBEANS, 2008]** NetBeans Organization, manual of programming java. Extraído 2007
disponible en: <http://www.netbeans.org.com/download/manuals>



Anexos

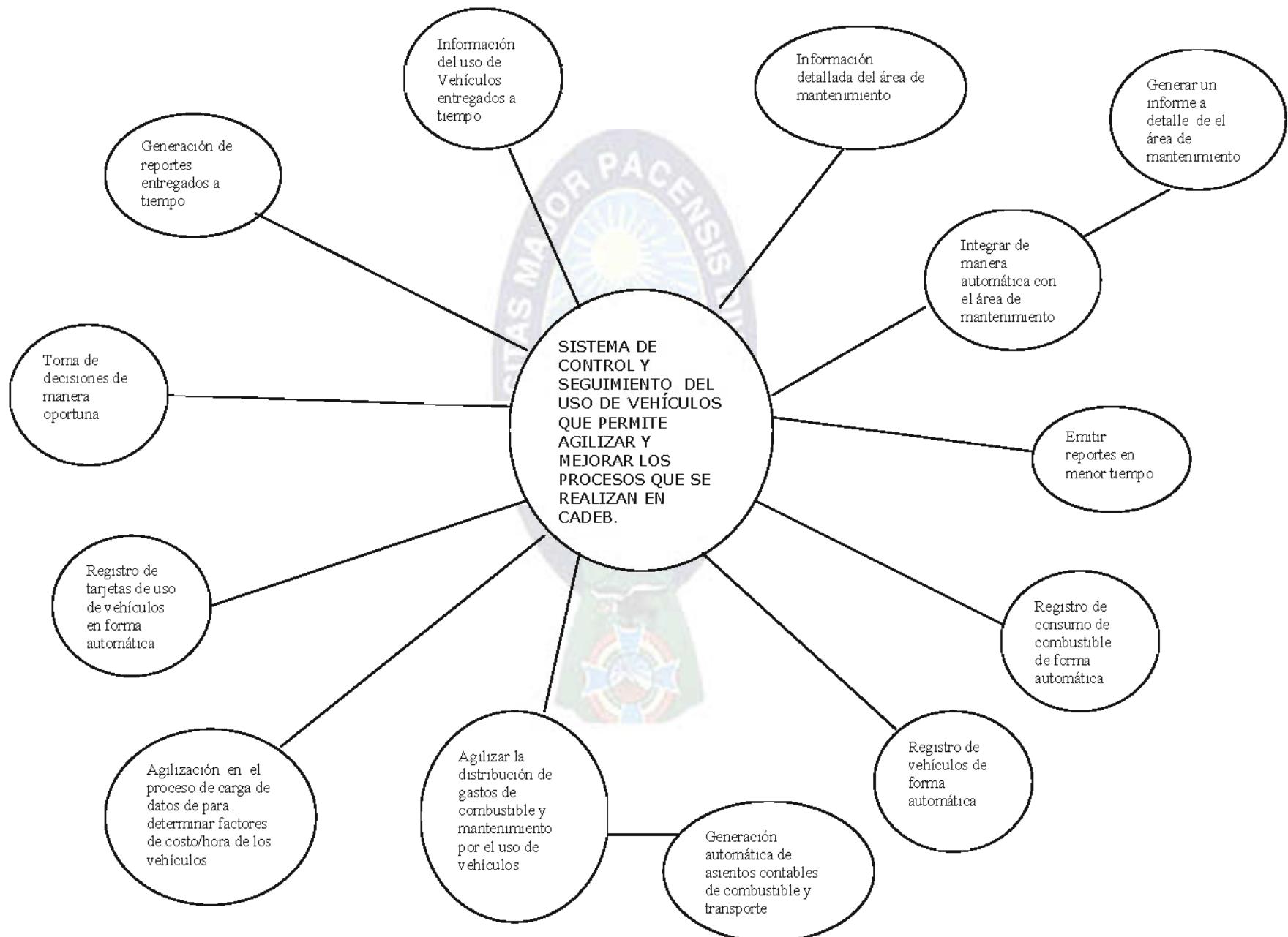
ANEXO A

ARBOL DE PROBLEMAS



ANEXO B

ARBOL DE OBJETIVOS



ANEXO C

MARCO LÓGICO

Resumen Narrativo	Indicadores Objetivamente Verificables	Medios de Verificación	Supuestos
<u>Fin Del Proyecto</u> Mejorar el acceso a la información generada por Procesos del Sistema de Control y Seguimiento del Uso de Vehículos brindando la información de manera disponible, confiable y actual.	<ul style="list-style-type: none"> ◆ Reducción de un 80% del tiempo para la Realización de los procesos del Sistema. ◆ Aumento de Accesibilidad en el Sistema de un 95%. ◆ Aumento de rendimiento en el Sistema de un 90%. 	<ul style="list-style-type: none"> ◆ Observación del funcionamiento del Sistema. ◆ Entrevistas al personal de el área de Informática. ◆ Recopilación de la Información de documentos 	Apoyo de la Institución para el desarrollo e implantación de el proyecto.
<u>Propósito del Proyecto</u> Implantar el sistema de Control y Seguimiento del Uso de Vehículos Vía Web para la Empresa CADEB S.A.	<ul style="list-style-type: none"> ◆ Desarrollo del Proyecto desde enero hasta mayo del 2009. ◆ Sistema de Control y Seguimiento de el Uso de Vehículos almacenado e instalado en el servidor SUN Application Server 8.1 en el Área de Informática de la Empresa CADEB S.A. junio del 2009 	<ul style="list-style-type: none"> ◆ Aval de el Tutor. ◆ Aval de el Revisor. ◆ Aval de la Empresa. ◆ Documentación de el Proyecto <p>Copias de Seguridad de el Sistema en disco e incluyendo el código fuente del sistema.</p>	Conclusión y Pruebas del Sistema.
<u>Producto</u> El Sistema de Control y Seguimiento del Uso de Vehículos comprende los siguientes módulos: <ul style="list-style-type: none">◆ Distribución◆ Parámetros procesos◆ Procesos◆ Reversiones◆ Mantenimiento	<p>Modulo de distribución realización en el mes de enero de 2009.</p> <p>Modulo de parámetros realización en el mes de febrero de 2009.</p> <p>Modulo de Procesos realización en el mes de Marzo de 2009.</p> <p>Modulo de Reversiones realización en el mes de Abril de 2009.</p> <p>Modulo de Mantenimiento realización en el mes de Mayo de 2009</p> <p>Sistema culminado y entregado el 30 de mayo de 2009.</p>	<p>Presentación de módulos probados y terminados</p> <p>Presentación de documentación concluida.</p>	<p>Seguimiento del cronograma de actividades.</p> <p>Disponibilidad de hardware y software y otros medios.</p>

Continuación del Marco Lógico

Resumen Narrativo	Indicadores Objetivamente Verificables	Medios de Verificación	Supuestos
<p>Actividades del Proyecto</p> <p>Elaboración de Cronograma Fase de Inicio Requerimientos Casos de Uso Fase de Elaboración Requerimientos Casos de Uso Análisis y Diseño Plan de pruebas Fase de Construcción Iteración 1 Iteración 2 Iteración 3 Fase de Transición Capacitación del personal para un buen uso del sistema.</p>	<ul style="list-style-type: none"> ◆ Material de Escritorio 600Bs. ◆ Compra de Libros 300Bs. ◆ Total de Carga Horaria estimado de 8000Bs 1400 Horas de Trabajo. 	<p>Verificación y seguimiento bajo cronograma establecido por la metodología RUP.</p> <p>Informes de Avances y pruebas durante el ciclo del desarrollo del Proyecto.</p>	<ul style="list-style-type: none"> ◆ Existencia de recursos financieros. ◆ Existe apoyo por parte de la gerencia del área de Informática y por parte de las autoridades de la empresa. ◆ Existen los medios para el desarrollo del proyecto.

ANEXO D

Tabla D1. Caso de Uso extendido Registrar Mantenimiento

CASO DE USO:	Registrar Mantenimiento
ACTORES:	Responsable de contabilidad
PROPOSITO:	Importación de datos de gastos de mantenimiento.
RESUMEN:	Este caso de uso importa la información de gastos de mantenimiento de vehículos realizada por otra empresa.
TIPO:	Secundario
REFERENCIA CRUZADA:	
CURSO NORMAL DE ACCIONES	
ACCION DEL ACTOR:	RESPUESTA DEL SISTEMA:
1. El Responsable de contabilidad selecciona la opción “Importar Registro de Mantenimiento”. 3. El Responsable de contabilidad selecciona el archivo a importar.	2. El sistema permite buscar la ruta física del archivo a importar. 4. El sistema procesa y valida la carga. 5. El sistema despliega el mensaje “Proceso concluyo satisfactoriamente”. 6. El sistema despliega la pantalla principal del modulo de vehículos hasta que el encargado seleccione la opción “Salir”.
CURSOS ALTERNATIVOS	

Tabla D2. Caso de Uso extendido Determinar Factores de Distribución (Costo/Hora)

CASO DE USO:	Determinar Factores de Distribución
ACTORES:	Jefe de Talleres
PROPOSITO:	Determinar los factores Costo/Hora.
RESUMEN:	Este caso de uso determina los factores Costo/Hora de vehículos a través de cálculos el cual es determinado por el Jefe de talleres.
TIPO:	Primario
REFERENCIA CRUZADA:	
CURSO NORMAL DE ACCIONES	
ACCION DEL ACTOR:	RESPUESTA DEL SISTEMA:
1. El Jefe de Talleres selecciona la opción “Factores Costo/Hora”. 3. El Jefe de Talleres verifica la información presentada de los factores.	2. El sistema despliega un formulario con los factores calculados del mes en curso. 4. El sistema se queda en la pantalla del despliegue hasta que el encargado seleccione la opción “salir”.
CURSOS ALTERNATIVOS	
<ul style="list-style-type: none"> - Se realiza la modificación de los factores costo/hora 	

Tabla D3. Caso de Uso extendido Registrar Consumo de Combustible

CASO DE USO:	Registrar Consumo de Combustible
ACTORES:	Encargado de Registro
PROPOSITO:	Registrar las tarjetas del consumo de combustible.
RESUMEN:	Este caso de uso realiza el registro de las tarjetas del uso de consumo de combustible los cuales son llenados por los conductores y luego transcritas por el encargado de registro.
TIPO:	Primario
REFERENCIA CRUZADA:	
CURSO NORMAL DE ACCIONES	
ACCION DEL ACTOR:	RESPUESTA DEL SISTEMA:
1. El encargado de registro selecciona la opción “Insertar vale”. 3. El encargado de registro ingresa los datos del vale. 5. El encargado selecciona la opción “Grabar vale”.	2. El sistema despliega un formulario en blanco de consumo de combustible. 4. El sistema valida los datos ingresados relacionados con otros módulos del sistema como ser Recursos Humanos. 6. El sistema graba el vale. 7. El sistema despliega otro formulario en blanco hasta que el encargado de registro selecciona la opción “Salir”.
CURSOS ALTERNATIVOS	
<ul style="list-style-type: none"> – Se realiza la modificación del vale. – Se realiza la eliminación del vale. – Se realiza el cierre de transcripción. 	

Tabla D4. Caso de Uso extendido Registrar Parámetros

CASO DE USO:	Registrar Parámetros
ACTORES:	Responsable de contabilidad, Jefe de Talleres
PROPOSITO:	Registrar los parámetros
RESUMEN:	Este caso de uso realiza el registro de todos los parámetros involucrados con el vehículo como ser : los odómetros, estaciones de servicio de combustible, externos y los parámetros de vehículos.
TIPO:	Primario
REFERENCIA CRUZADA:	
CURSO NORMAL DE ACCIONES	
ACCION DEL ACTOR:	RESPUESTA DEL SISTEMA:
1. El encargado selecciona la opción “Registrar Parámetros” 3. El encargado elige una opción. 5. El encargado elige “Insertar” e ingresa los datos. 7. El encargado selecciona la opción “Grabar”.	2. El sistema despliega una pantalla con las opciones de parametrizar: Odómetros, Estaciones de Servicio de Combustible, Externos y los parámetros de Vehículos. 4. El sistema despliega el formulario con las opciones de insertar, modificar y eliminar. 6. El sistema valida los datos ingresados. 8. El sistema graba la información. 9. El sistema despliega otro formulario en blanco hasta que el encargado seleccione la opción “Salir”.
CURSOS ALTERNATIVOS	
<ul style="list-style-type: none"> – Se realiza la modificación de los parámetros – Se realiza la eliminación de los parámetros 	

Tabla D5. Caso de Uso extendido Generación de Asientos Contables

CASO DE USO:	Generación de Asientos Contables
ACTORES:	Responsable de contabilidad
PROPOSITO:	Generación de Asientos Contables
RESUMEN:	Este caso de uso realiza la generación de asientos contables de combustible y transporte en base a la información generada en el proceso de distribución de gastos. Además de realizar cargos a proyectos. para la realización de los asientos contables con cargos a los centros de costo se realiza en función a la distribución de bienes y los cargos a terceros son directos a las cuentas por cobrar
TIPO:	Secundario
REFERENCIA CRUZADA:	
CURSO NORMAL DE ACCIONES	
ACCION DEL ACTOR:	RESPUESTA DEL SISTEMA:
1. El responsable selecciona la opción “Cierre de Mes” 3. El responsable selecciona “generar asientos contables”. 5. El responsable confirma la generación.	2. El sistema despliega el mes pendiente de cierre y las opciones de “Distribuir gastos”, “Generación de Asientos Contables”, “Cierre de Distribución”, “Cierre de transcripción” y “reversión de la generación de asientos”. 4. El sistema solicita la confirmación para generar asientos contables. 6. El sistema procesa y graba información de los asientos contables y los cargos a proyectos. 7. El sistema despliega un mensaje una vez finalizado el proceso. 8. El sistema despliega la pantalla “cierre de mes” hasta que el responsable seleccione la opción “salir”.
CURSOS ALTERNATIVOS	
– Se realiza la reversión de la generación de asientos.	

Tabla D6. Caso de Uso extendido Distribuir Gastos

CASO DE USO:	Distribuir Gastos
ACTORES:	Responsable de Contabilidad
PROPOSITO:	Realizar la distribución de Gastos del consumo de combustible y mantenimiento por el uso de vehículos.
RESUMEN:	Este caso de uso realiza la distribución de Gastos del consumo de combustible y mantenimiento por el uso de vehículos entre los diferentes Centros de Costos, Proyectos y Cuentas de Terceros dependiendo del uso que hubieran tenido y del costo/hora.
TIPO:	Primario
REFERENCIA CRUZADA:	
CURSO NORMAL DE ACCIONES	
ACCION DEL ACTOR:	RESPUESTA DEL SISTEMA:
1. El responsable selecciona la opción “Cierre de Mes” 3. El responsable selecciona “Distribuir Gastos”	2. El sistema despliega el mes pendiente de cierre y las opciones de “Distribuir Gastos”, “Generación de Asientos Contables”, “Cierre de Distribución” y “Reabrir Transcripción” 4. El sistema graba la información de la distribución de gastos. 5. El sistema presenta un mensaje una vez finalizado el proceso 6. El sistema despliega la pantalla “Cierre de Mes” hasta que el responsable seleccione la opción “Salir”.
CURSOS ALTERNATIVOS	
Se realiza el “Cierre de Distribución” y “Reabrir Transcripción”.	

ANEXO E

Tabla E1. Contrato Mantener Información Vehículos

CONTRATO	
Nombre:	Información de vehículos ()
Responsabilidades:	Capturar (registrar) la información de los vehículos.
Tipo:	Sistema
Referencias cruzadas:	Funciones del sistema. Caso de Uso Mantener Información Vehículos
Notas:	Utilizar el acceso súper rápido a la base de datos
Excepciones:	De existir error en los procesos de Modificar e Insertar, el sistema despliega un mensaje de error.
Salida:	
Precondiciones:	<ul style="list-style-type: none"> • Antes de que el caso de uso empiece el encargado ya tuvo que hacer un login en el sistema y tener la autorización correspondiente. • Antes de que el caso de uso empiece se debe haber realizado el cierre de Transcripción e importación de la información de mantenimiento.
Poscondiciones:	

Tabla E2. Contrato Distribuir Gastos

CONTRATO	
Nombre:	Distribuir gastos ()
Responsabilidades:	Distribuir los datos sobre el consumo del combustible y del mantenimiento del uso de vehículos a los centros de costos, proyectos y cuentas a terceros.
Tipo:	Sistema
Referencias cruzadas:	Funciones del sistema. Caso de Uso Distribuir Gasto.
Notas:	Utilizar el acceso súper rápido a la base de datos
Excepciones:	Si los datos no son validos indicar que se cometió un error
Salida:	
Precondiciones:	<ul style="list-style-type: none"> • Antes de que el caso de uso empiece el encargado ya tuvo que hacer un login en el sistema y tener la autorización correspondiente. • Antes de que el caso de uso empiece el estado de transcripción de uso de vehículos y consumo de combustible debe ser “cerrado” • Antes de que el caso de uso empiece la distribución debe estar en estado “abierto”
Poscondiciones:	Después de que el caso de uso de distribución de gastos termina el estado del uso de vehículos y consumo de combustible debe de estar cerrado.

Tabla E3. Contrato Determinar Factores Costo/Hora

CONTRATO	
Nombre:	Factores Costo/Hora()
Responsabilidades:	Se Realiza la carga de los factores costo /hora desde un archivo xls.
Tipo:	Sistema
Referencias cruzadas:	Funciones del sistema
Notas:	Realizar la carga de datos hacia la base de datos de manera rápida.
Excepciones:	De existir error en el proceso de Modificar factores costo/hora el sistema despliega un mensaje de error.
Salida:	
Precondiciones:	Antes de que el caso de uso empiece el encargado ya tuvo que hacer un login en el sistema y tener la autorización correspondiente.
Poscondiciones:	

Tabla E4. Contrato Registrar Consumo de Combustible

CONTRATO	
Nombre:	Registro consumo combustible ()
Responsabilidades:	Se registra (captura) el consumo de combustible mediante los vales.
Tipo:	Sistema
Referencias cruzadas:	Funciones del sistema
Notas:	Utilizar el acceso súper rápido a la base de datos
Excepciones:	De existir error en el proceso de Modificar e insertar el sistema despliega un mensaje de error.
Salida:	
Precondiciones:	Antes de que el caso de uso empiece el encargado ya tuvo que hacer un login en el sistema y tener la autorización correspondiente.
Poscondiciones:	

Tabla E5. Contrato Registrar Mantenimiento

CONTRATO	
Nombre:	Registrar mantenimiento ()
Responsabilidades:	Se realiza la importación de los datos del registro del mantenimiento.
Tipo:	Sistema
Referencias cruzadas:	Funciones del sistema
Notas:	Utilizar el acceso súper rápido a la base de datos
Excepciones:	De existir error en el proceso de importar el sistema despliega un mensaje de error.
Salida:	
Precondiciones:	Antes de que el caso de uso empiece el encargado ya tuvo que hacer un login en el sistema y tener la autorización correspondiente.
Poscondiciones:	

Tabla E6. Contrato Registrar Parámetros

CONTRATO	
Nombre:	Registro Parámetros ()
Responsabilidades:	Se registra (captura) los parámetros con relación a los vehículos.
Tipo:	Sistema
Referencias cruzadas:	Funciones del sistema
Notas:	Utilizar el acceso súper rápido a la base de datos
Excepciones:	De existir error en los procesos de modificar e insertar, el sistema despliega un mensaje de error.
Salida:	
Precondiciones:	<ul style="list-style-type: none"> • Antes de que el caso de uso empiece el encargado ya tuvo que hacer un login en el sistema y tener la autorización correspondiente. • Cuando el caso de uso empiece el sistema valida las opciones a las cuales tiene acceso el encargado correspondiente. • Antes de realizar la opción de eliminar, verifica que el registro no haya sido usado.
Poscondiciones:	

Tabla E7. Contrato Generación de Asientos Contables

CONTRATO	
Nombre:	Genera Asiento Contable ()
Responsabilidades:	Se realiza la generación de asientos contables.
Tipo:	Sistema
Referencias cruzadas:	Funciones del sistema
Notas:	Utilizar el acceso súper rápido a la base de datos
Excepciones:	De existir error en el proceso de generación, el sistema despliega un mensaje de error.
Salida:	
Precondiciones:	<ul style="list-style-type: none"> • Antes de que el caso de uso empiece el encargado ya tuvo que hacer un login en el sistema y tener la autorización correspondiente. • Antes de que el caso de uso empiece debe estar habilitado para la generación de asientos contables en los parámetros de modulo y el estado de la distribución de gastos debe ser “cerrado”. Además no deberá existir ya los asientos para el periodo actual.
Poscondiciones:	

ANEXO F

Figura F1. Diagrama de Colaboración Parámetros odómetro

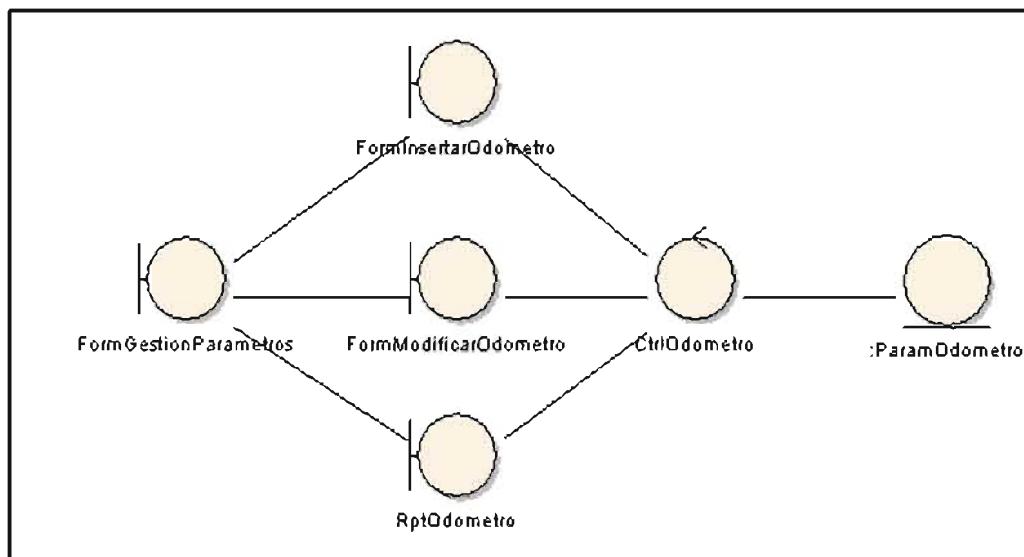


Figura F2. Diagrama de Colaboración Parámetros surtidores

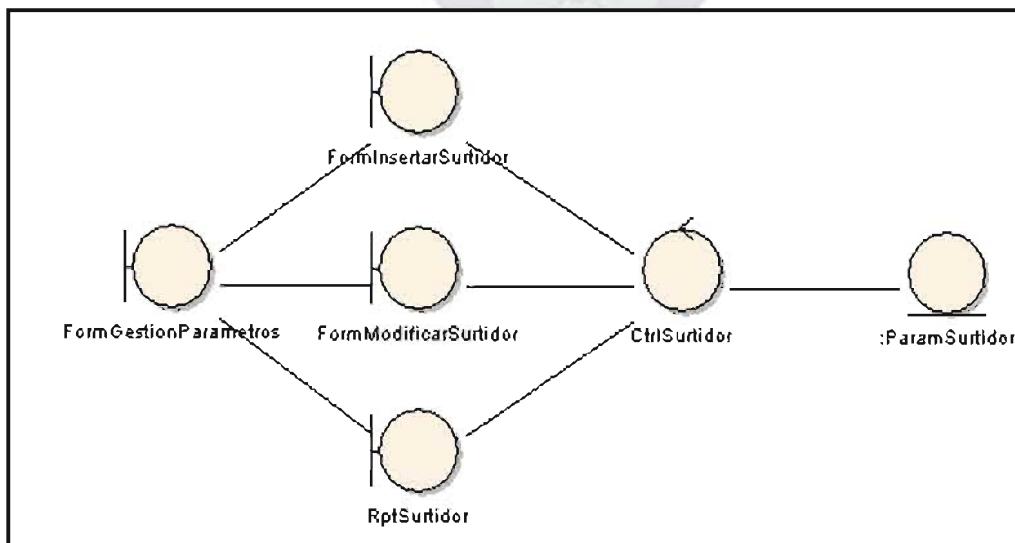


Figura F3. Diagrama de Colaboración Parámetros subgrupo vehículos

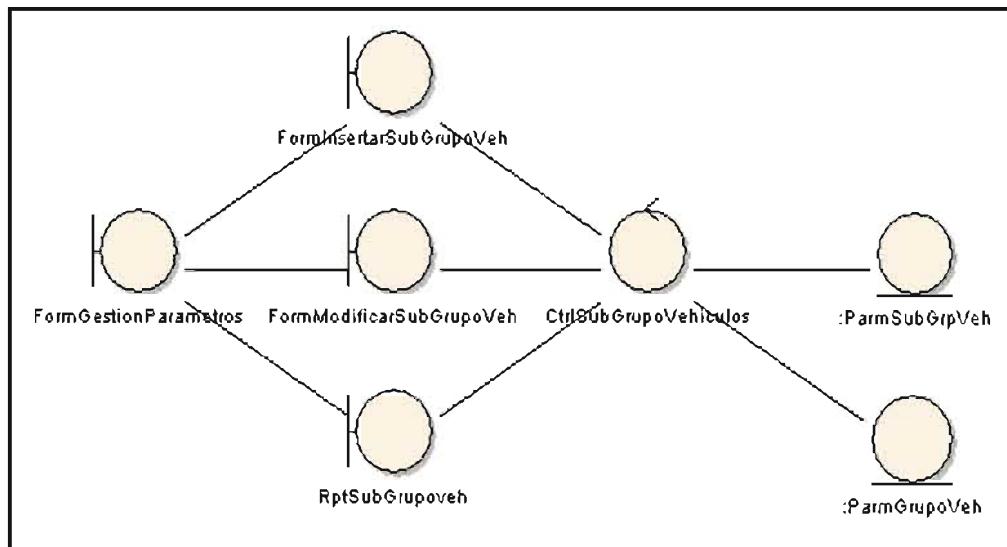


Figura F4. Diagrama de Colaboración. Parámetros generales

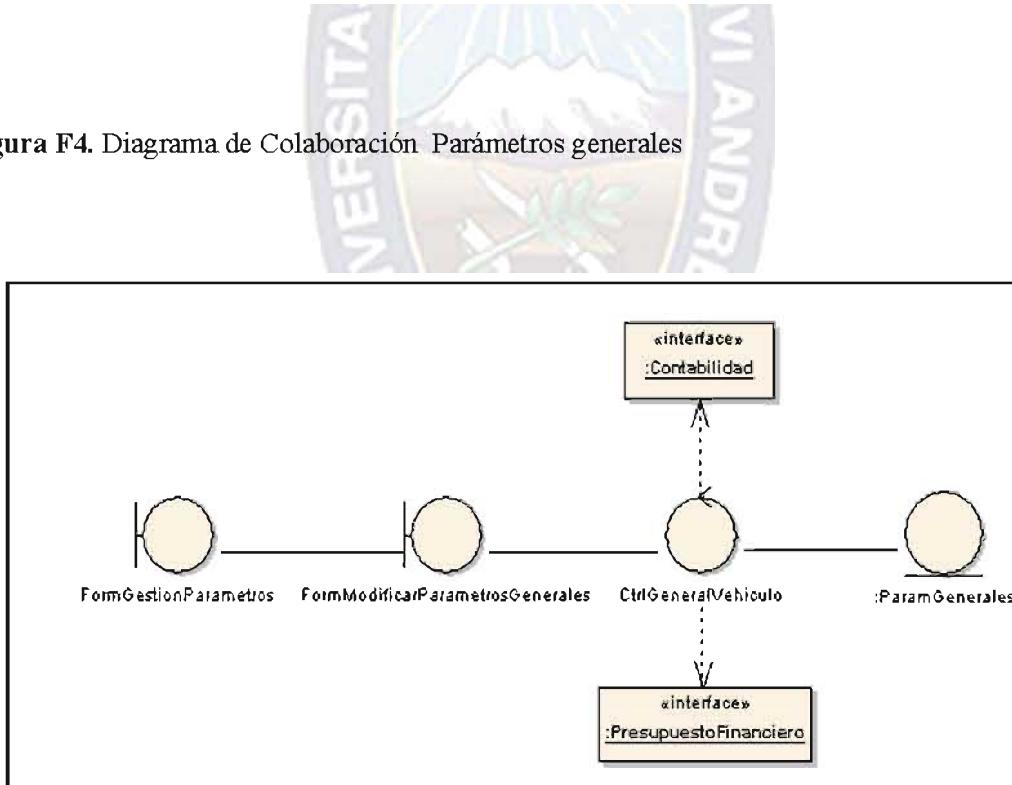


Figura F5 Diagrama de Colaboración Registrar consumo de combustible

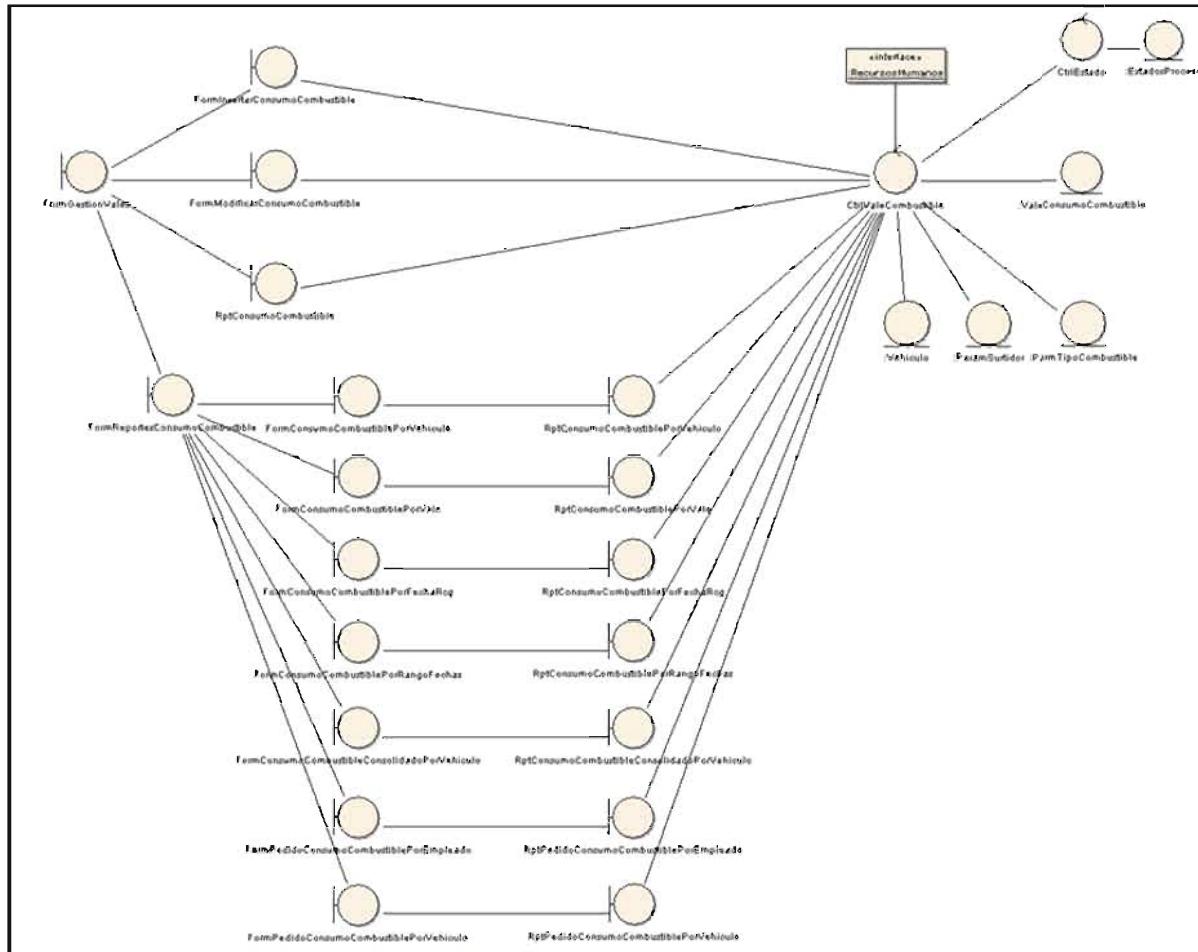


Figura F6. Diagrama de Colaboración Parámetros empresas externas

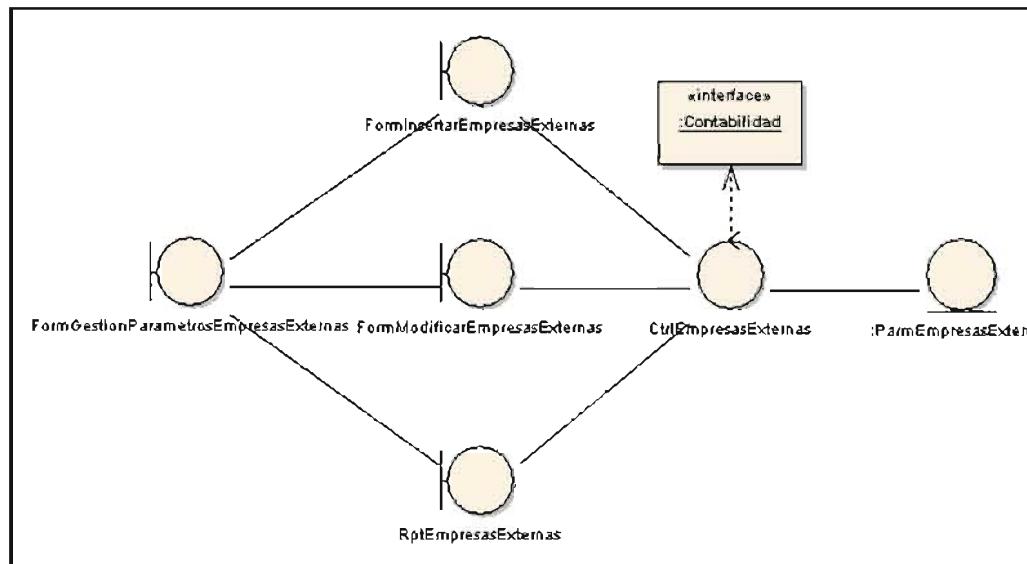


Figura F7. Diagrama de Colaboración Registrar mantenimiento

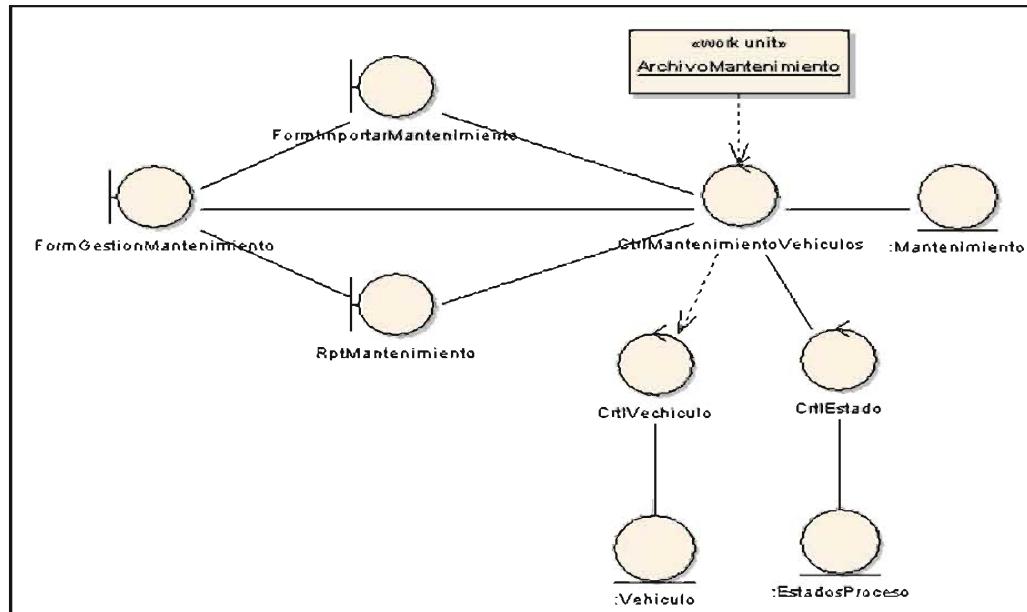


Figura F8. Diagrama de Colaboración Determinar factores costo hora

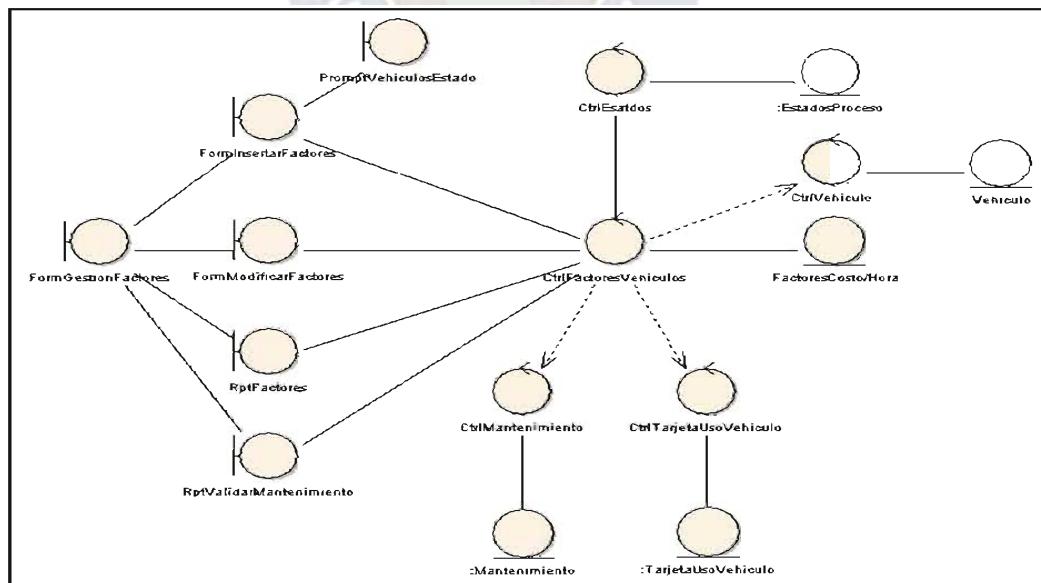


Figura F9. Diagrama de Colaboración Distribuir gastos

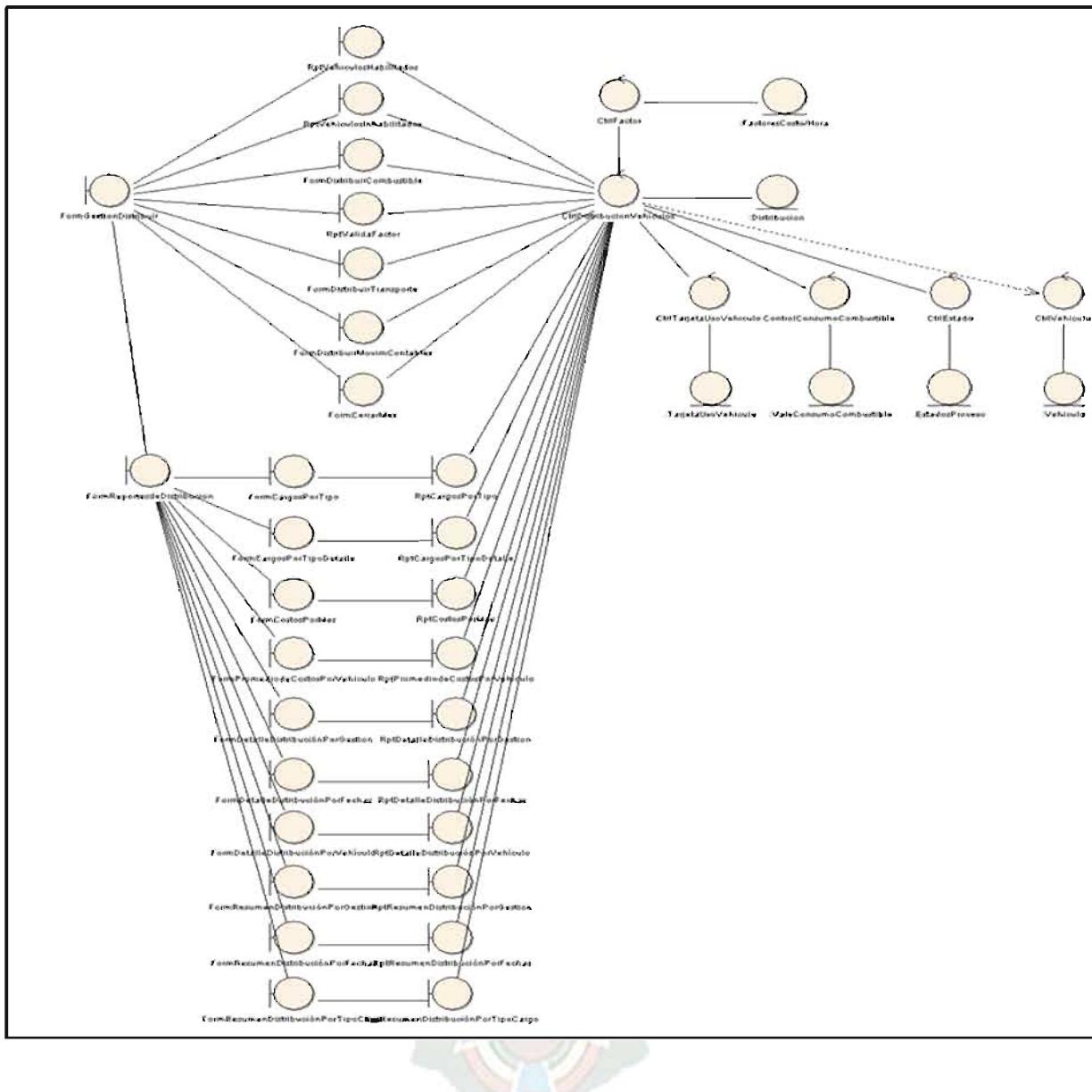


Figura F10. Diagrama de Colaboración Generar asientos contables

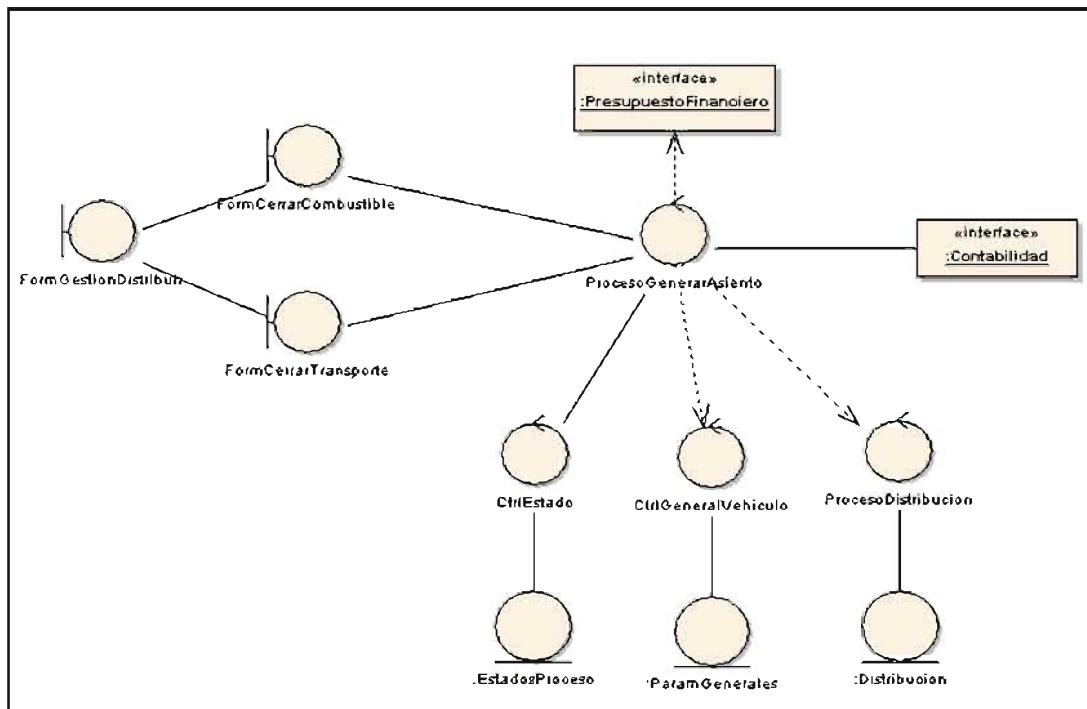


Figura F11. Diagrama de Colaboración Revertir cierre vehículos

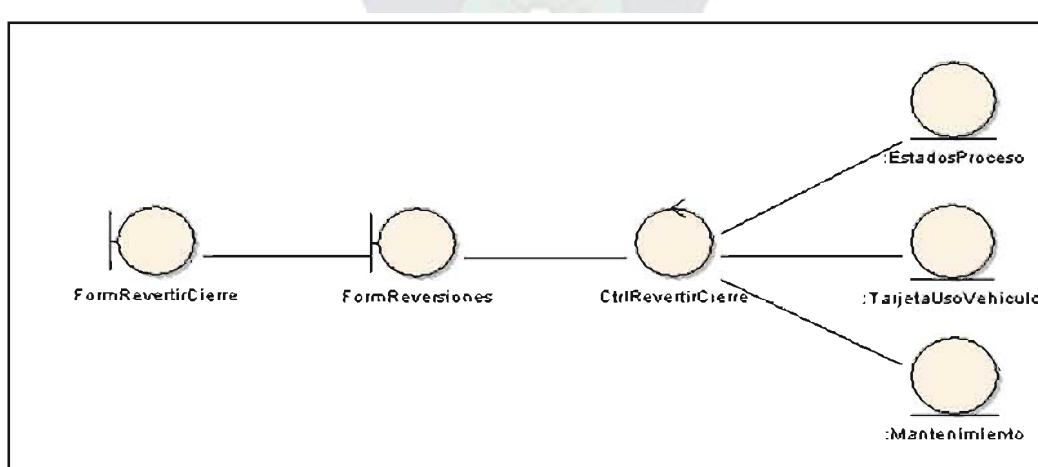


Figura F12. Diagrama de Colaboración Revertir cierre contable

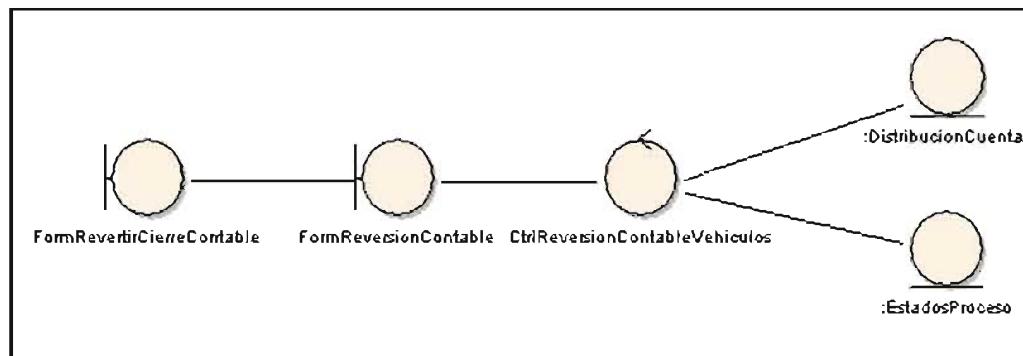
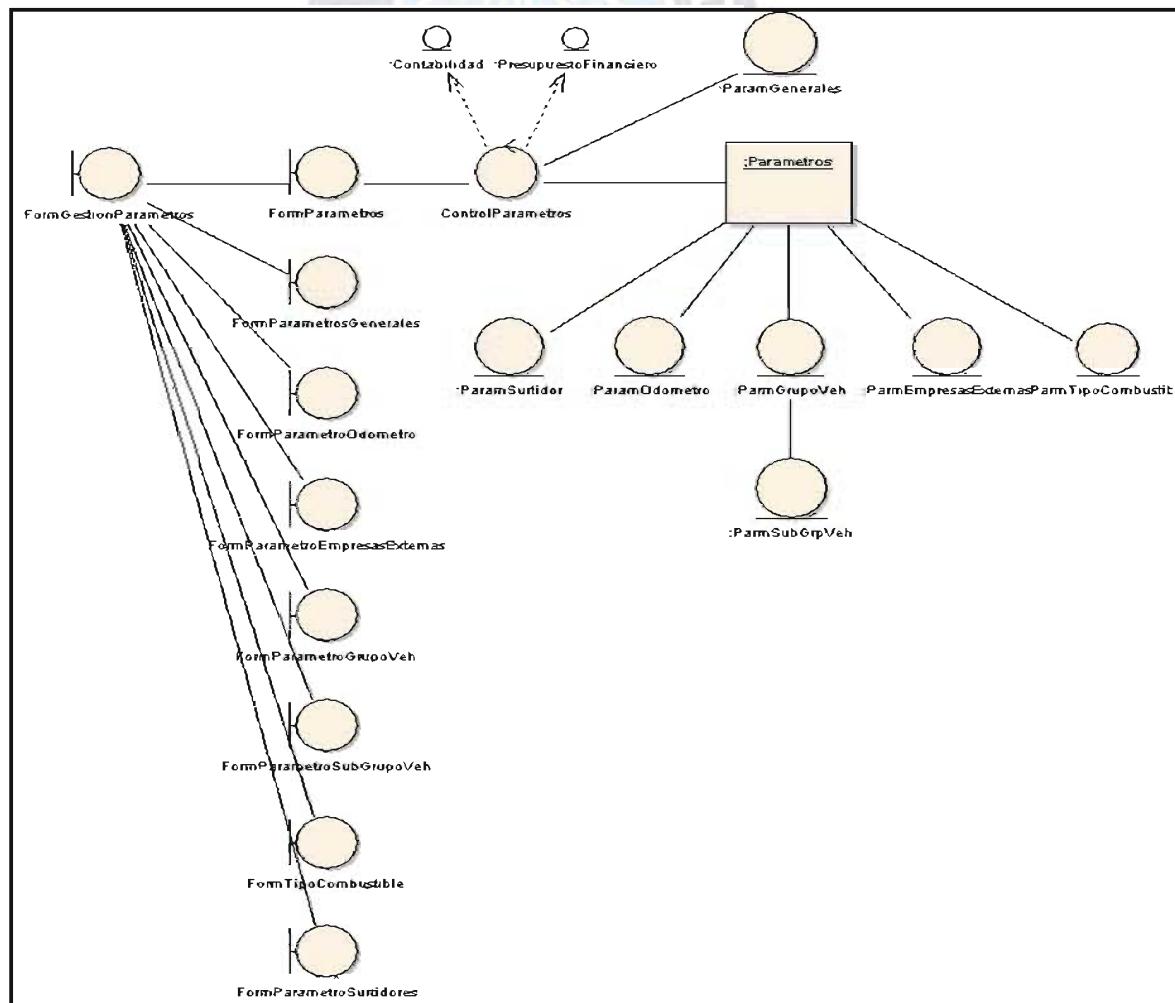


Figura F13. Diagrama de Colaboración Registrar parámetros



ANEXO G

Figura G1. Diagrama de Actividad Cerrar transcripción

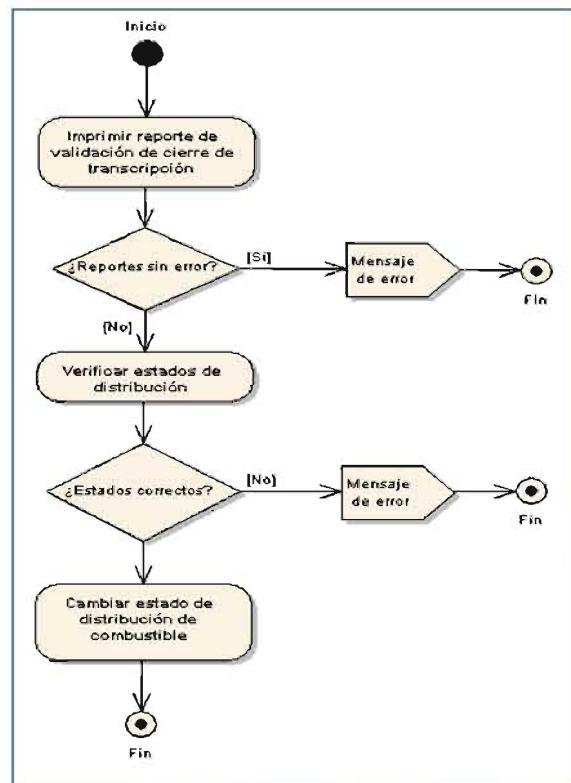


Figura G2. Diagrama de Actividad Revertir cierre contable

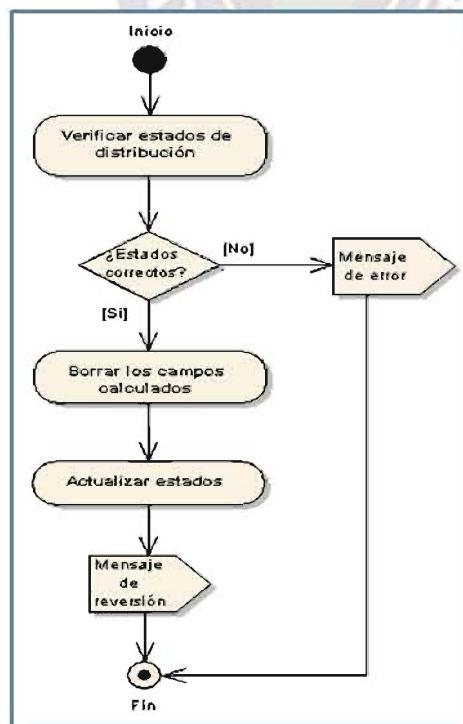


Figura G3. Diagrama de Actividad Confirmar factores

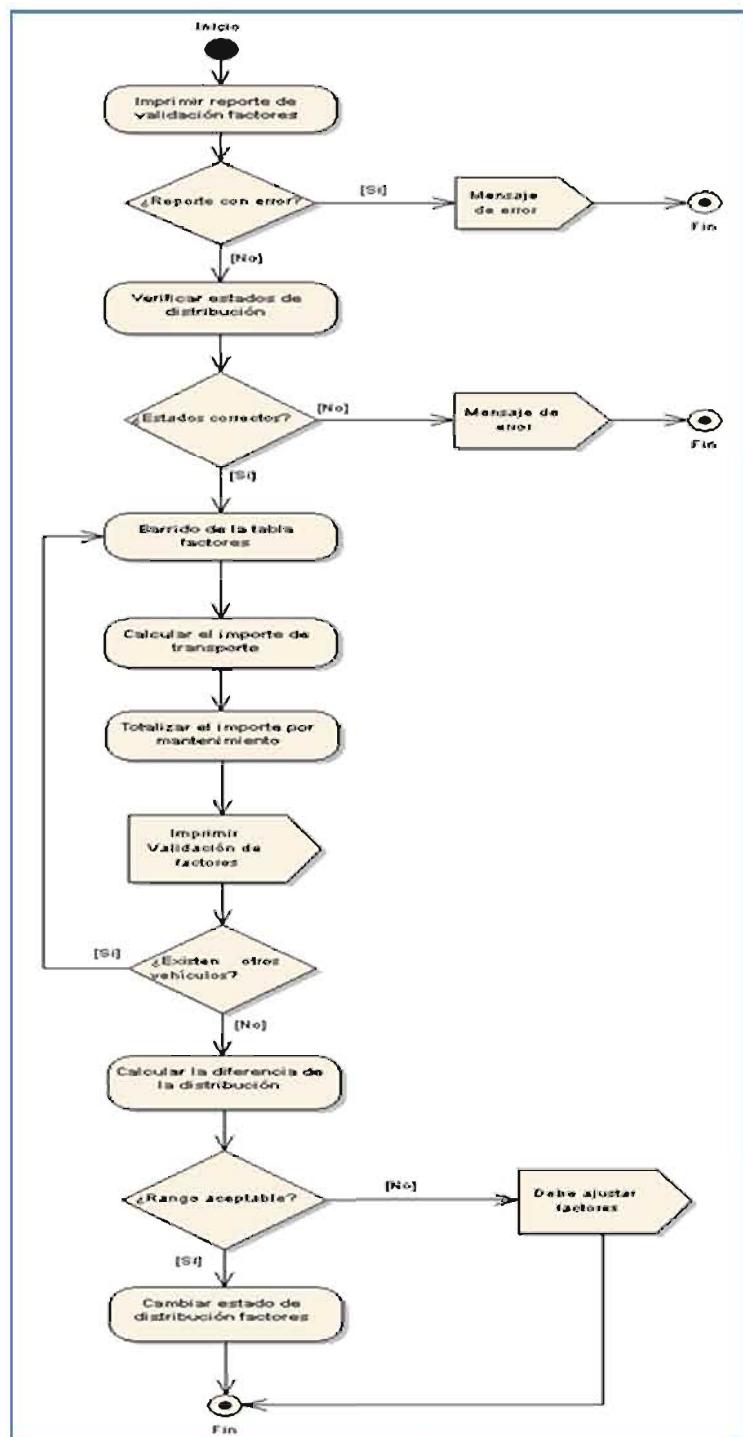


Figura G4. Diagrama de Actividad Generar asiento contable de combustible

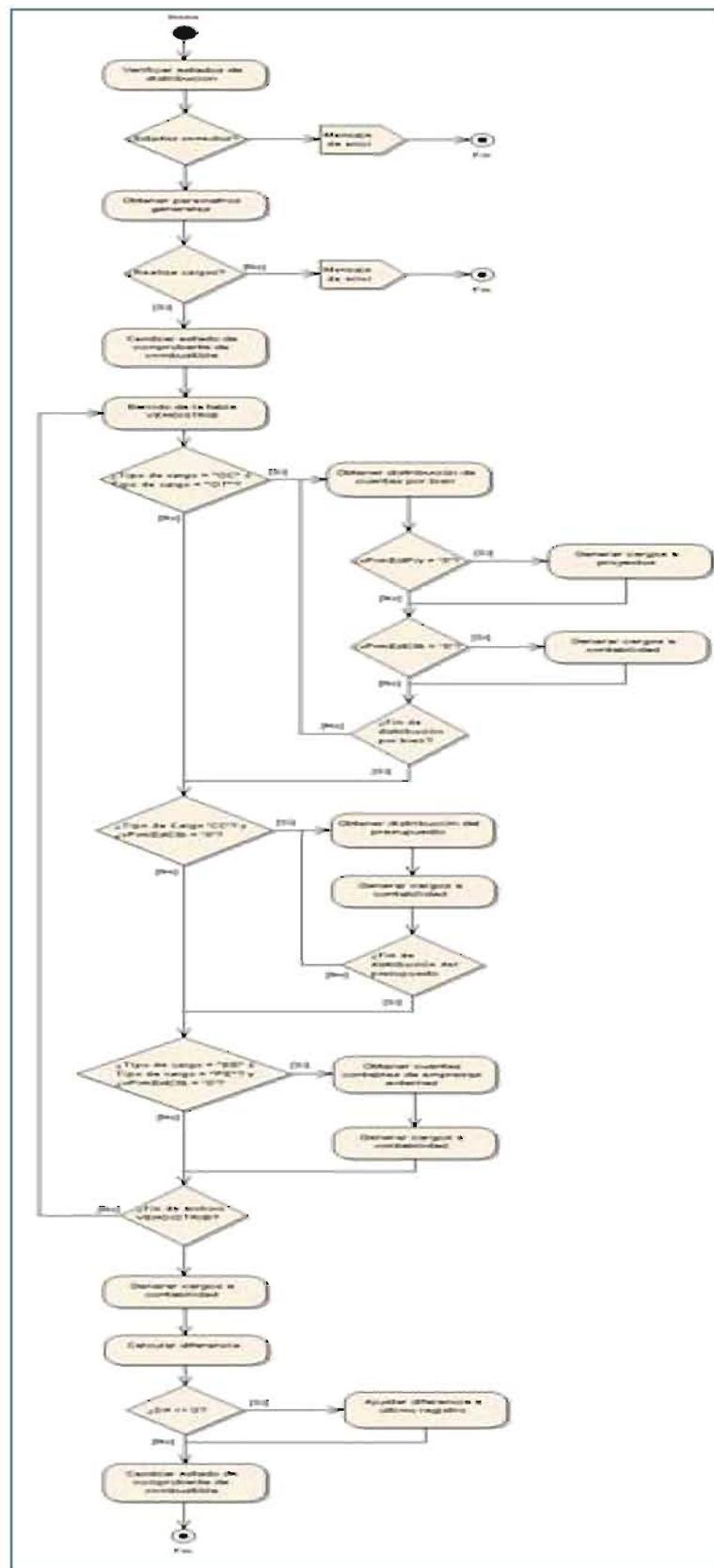


Figura G5. Diagrama de Actividad Generar asiento contable de transporte

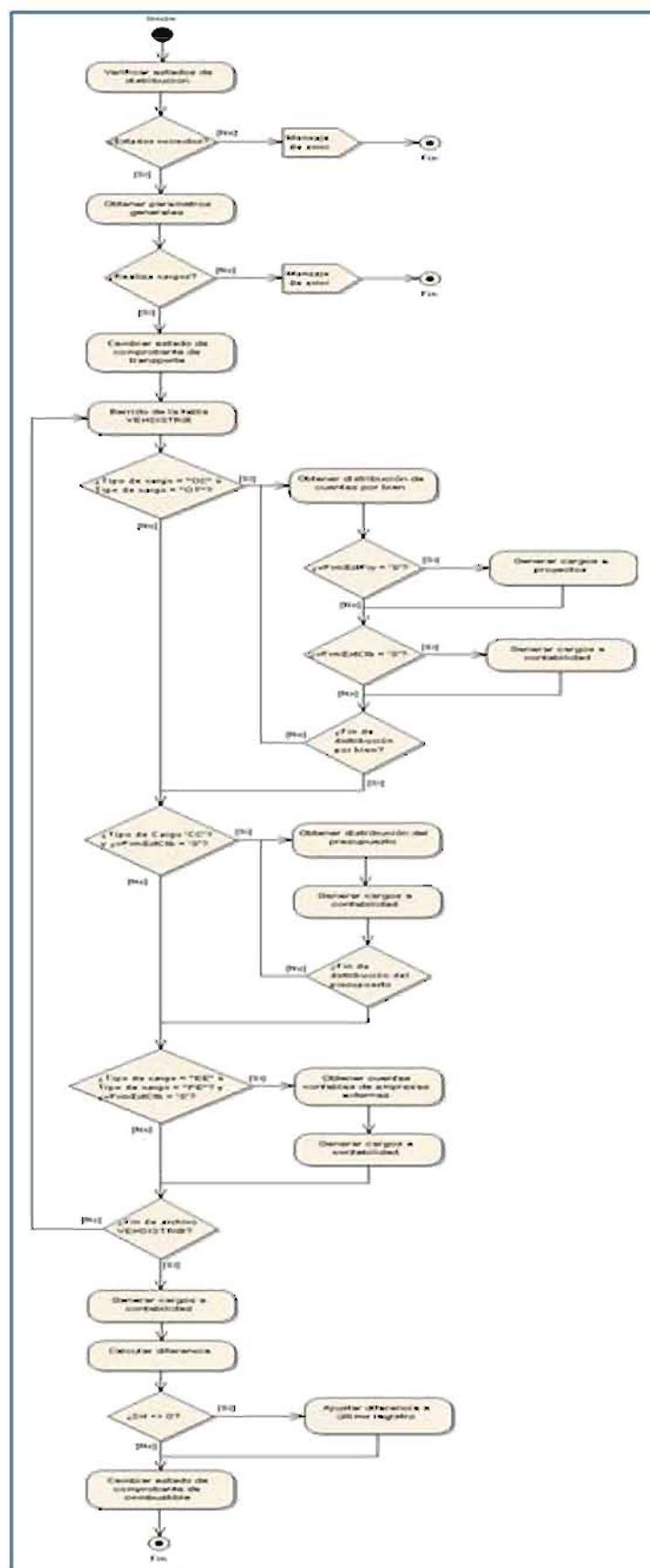


Figura G6. Diagrama de Actividad Distribuir combustible

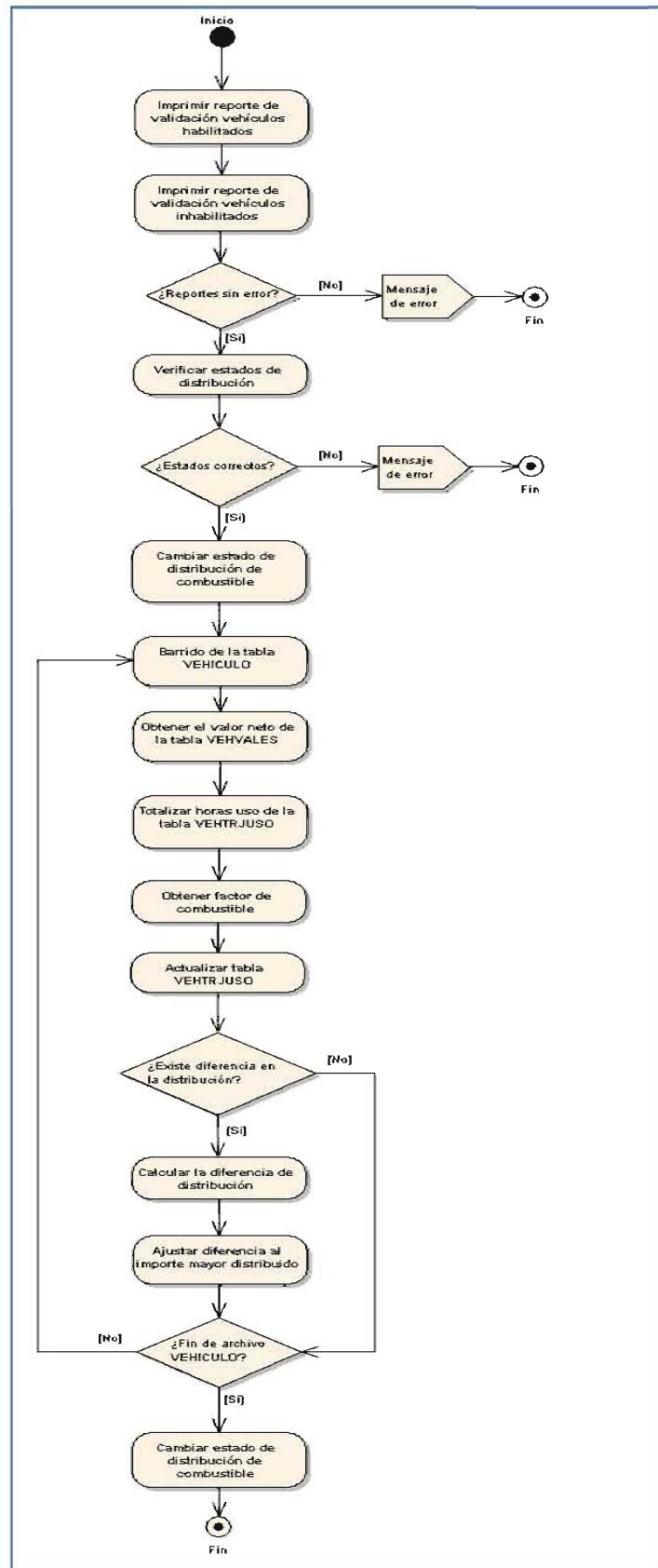


Figura G7. Diagrama de Actividad Distribuir transporte

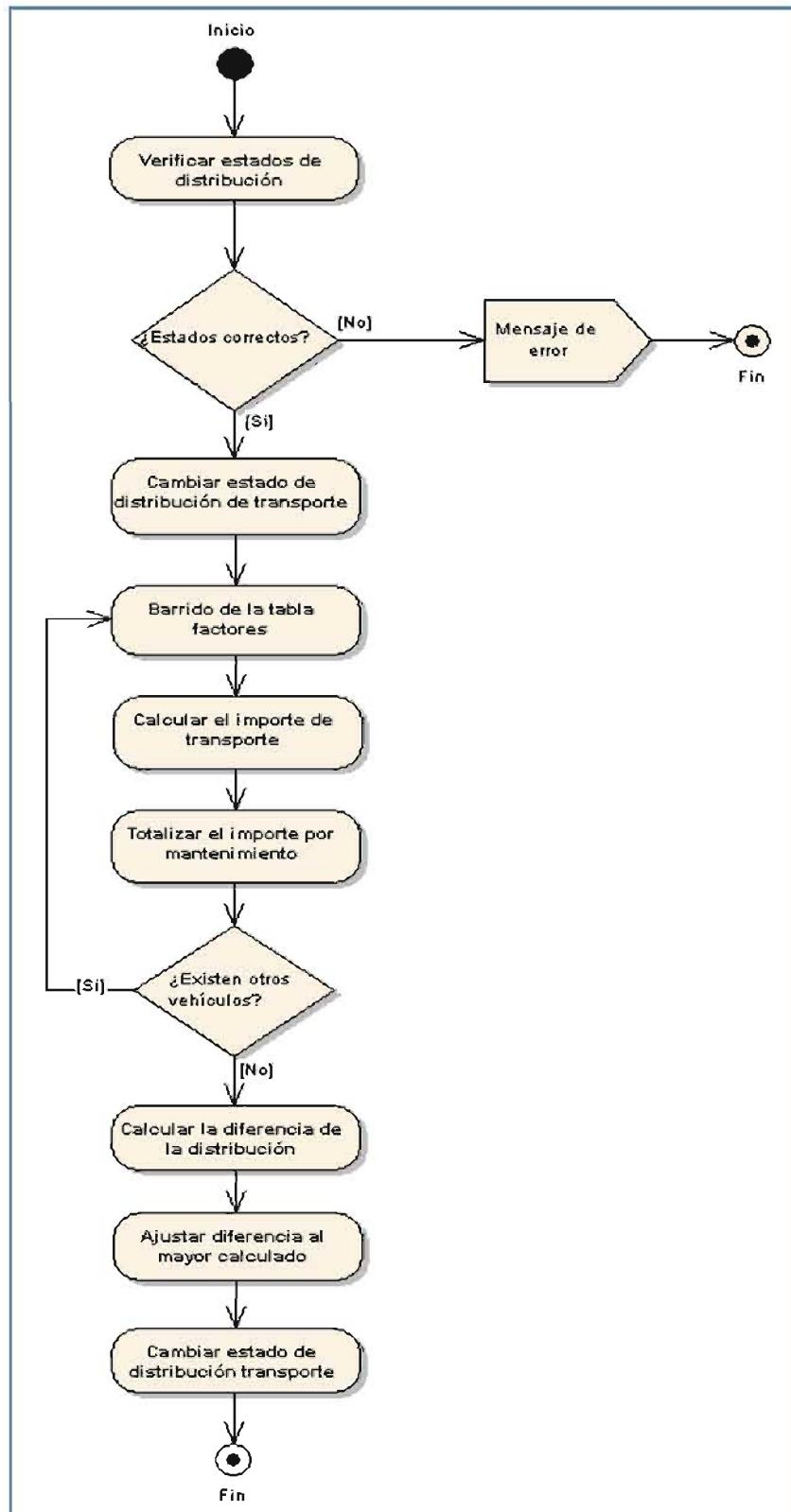


Figura G8. Diagrama de Actividad Revertir factores

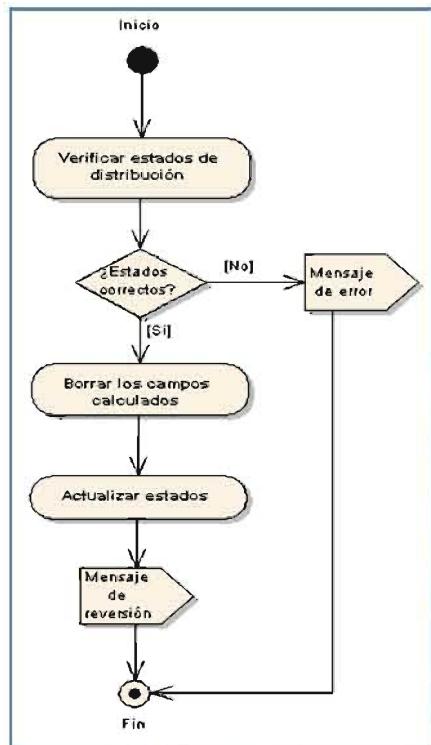


Figura G9. Diagrama de Actividad Revertir mantenimiento

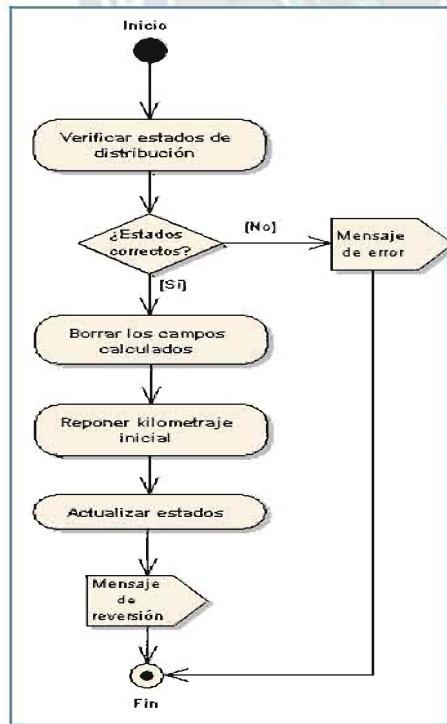


Figura G10. Diagrama de Actividad Revertir distribución de transporte

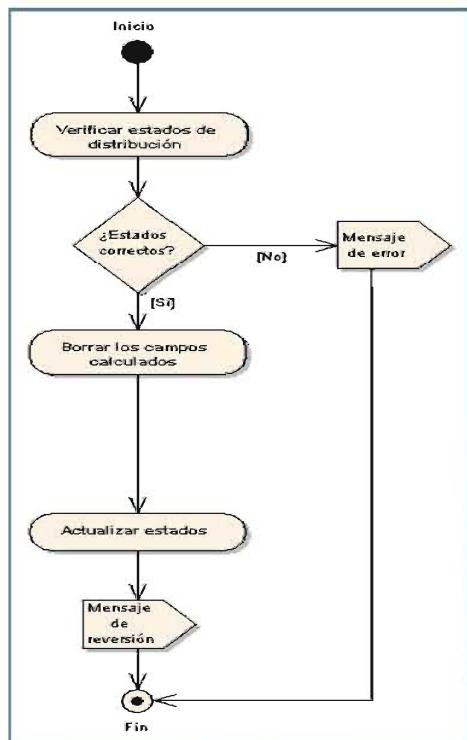


Figura G11. Diagrama de Actividad Revertir vales

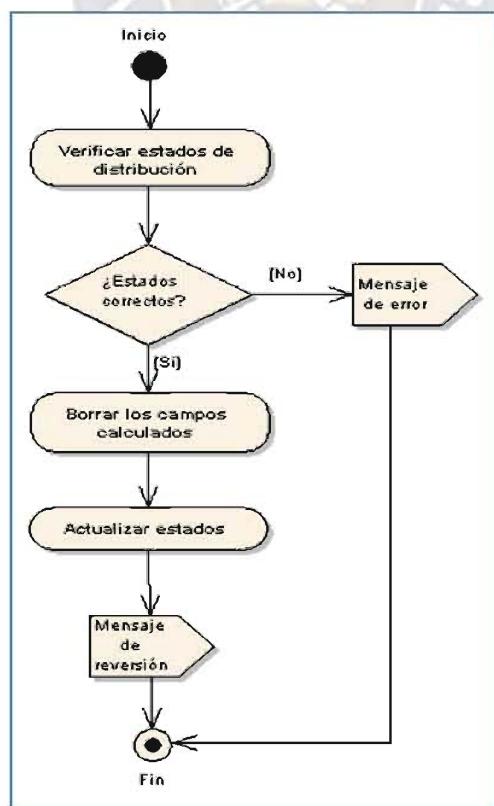


Figura G12. Diagrama de Actividad Totalizar recorrido

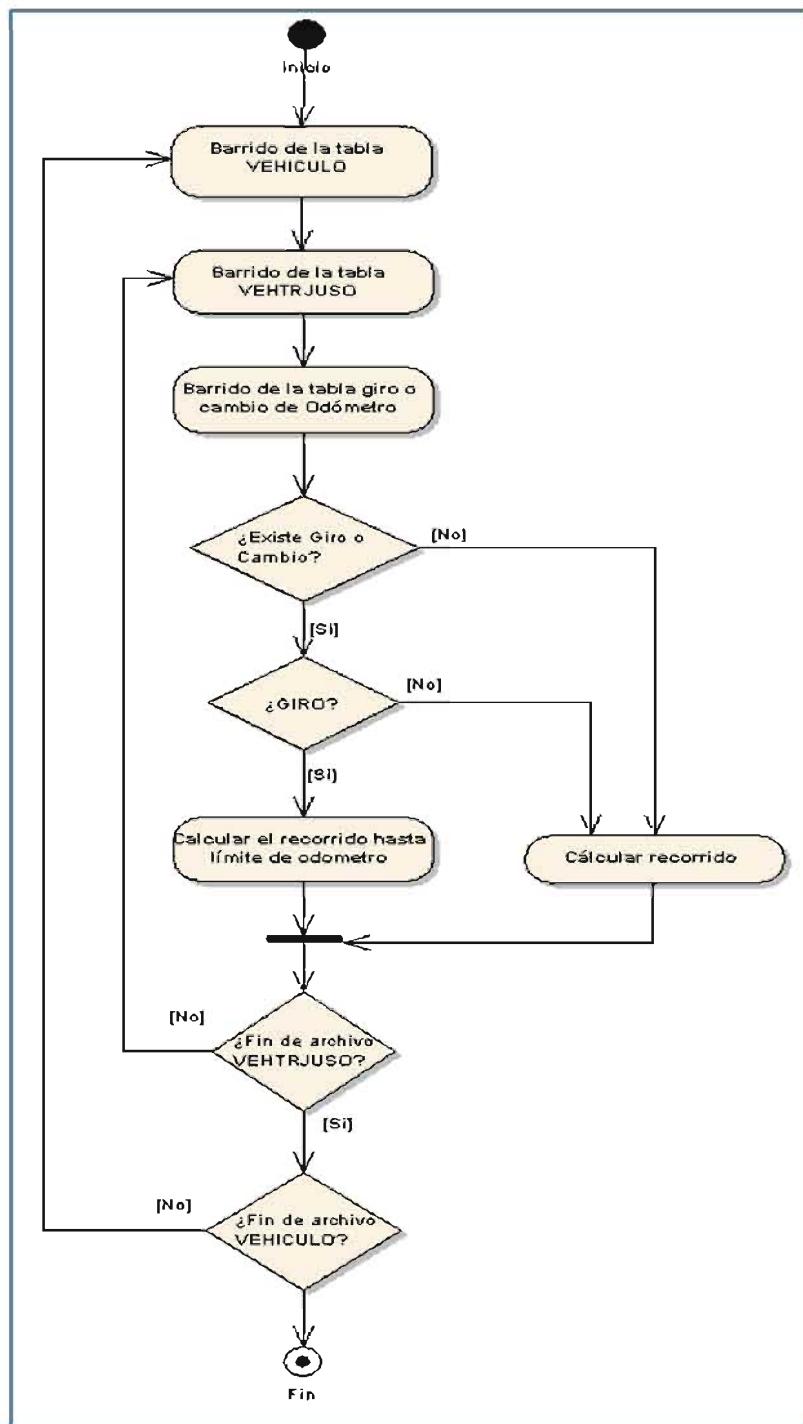
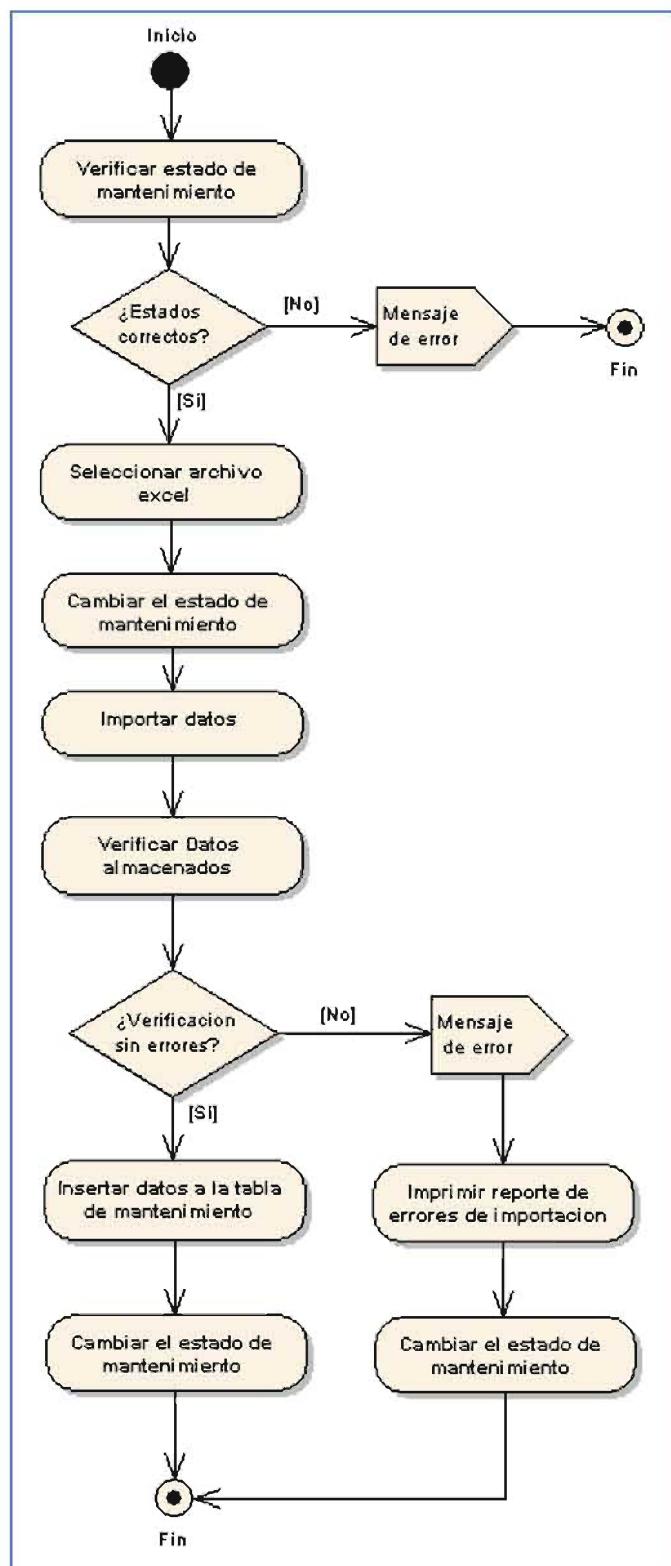


Figura G13. Diagrama de Actividad Importar mantenimiento



ANEXO H

Figura H1. Diagrama de secuencia Determinar Factores Costo/Hora

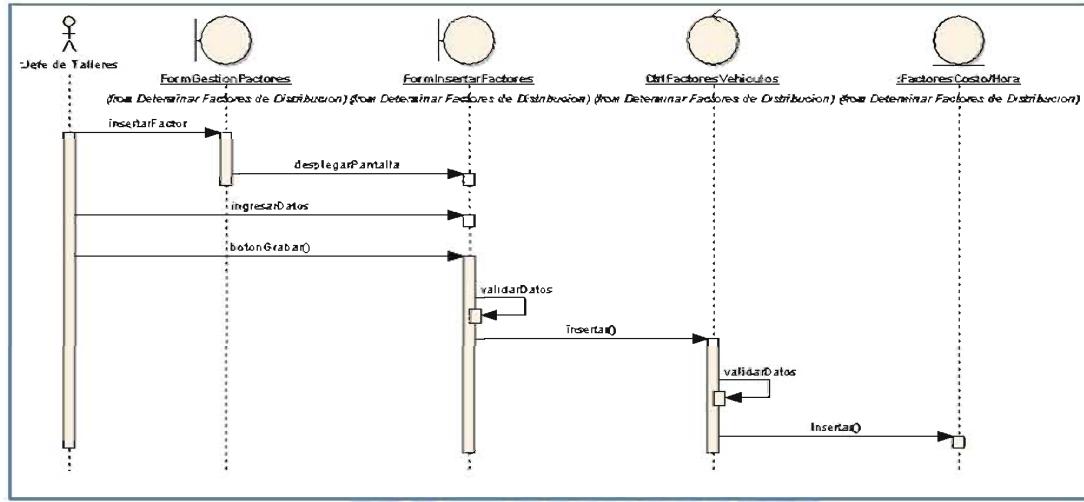


Figura H2. Diagrama de secuencia Registrar consumo de combustible

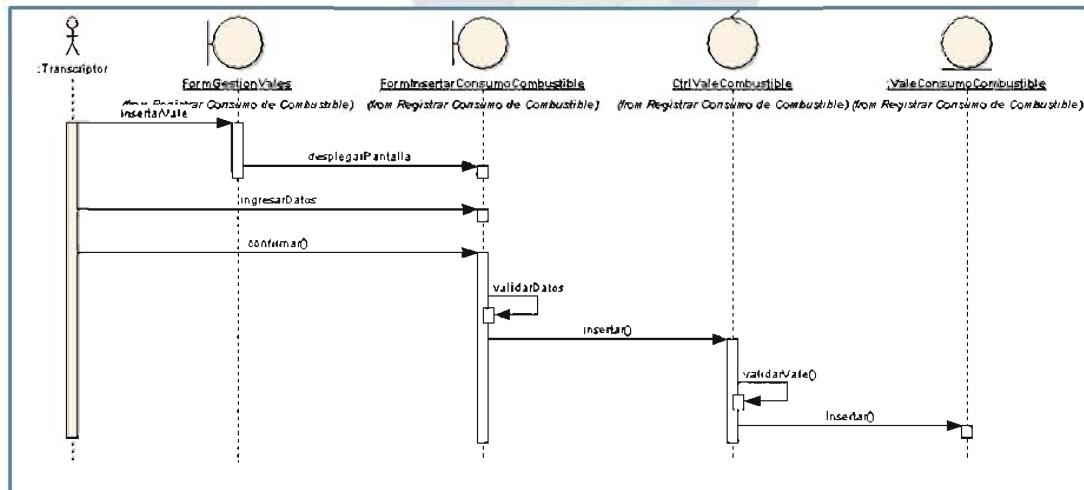


Figura H3. Diagrama de secuencia Parámetros de empresas externas

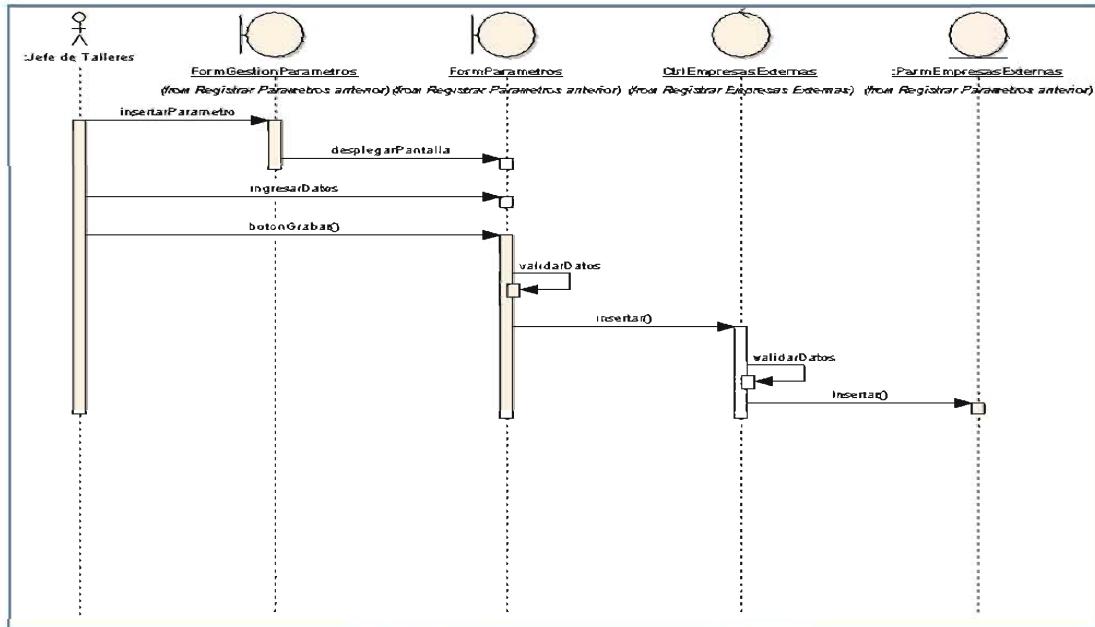


Figura H4. Diagrama de secuencia Parámetros grupos vehículos

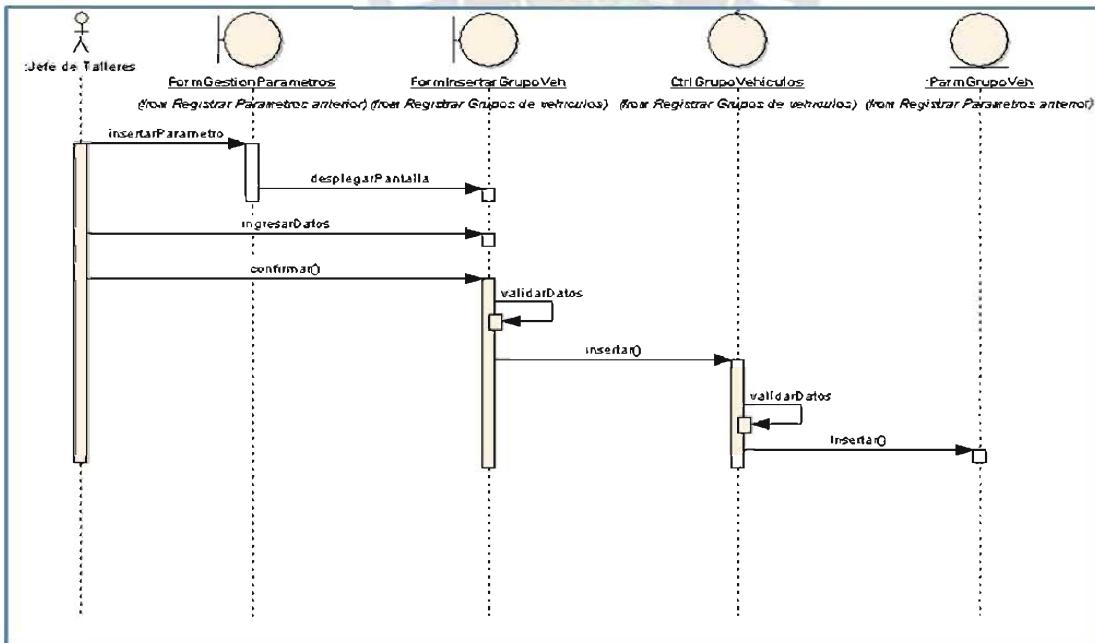


Figura H5. Diagrama de secuencia Registrar mantenimiento

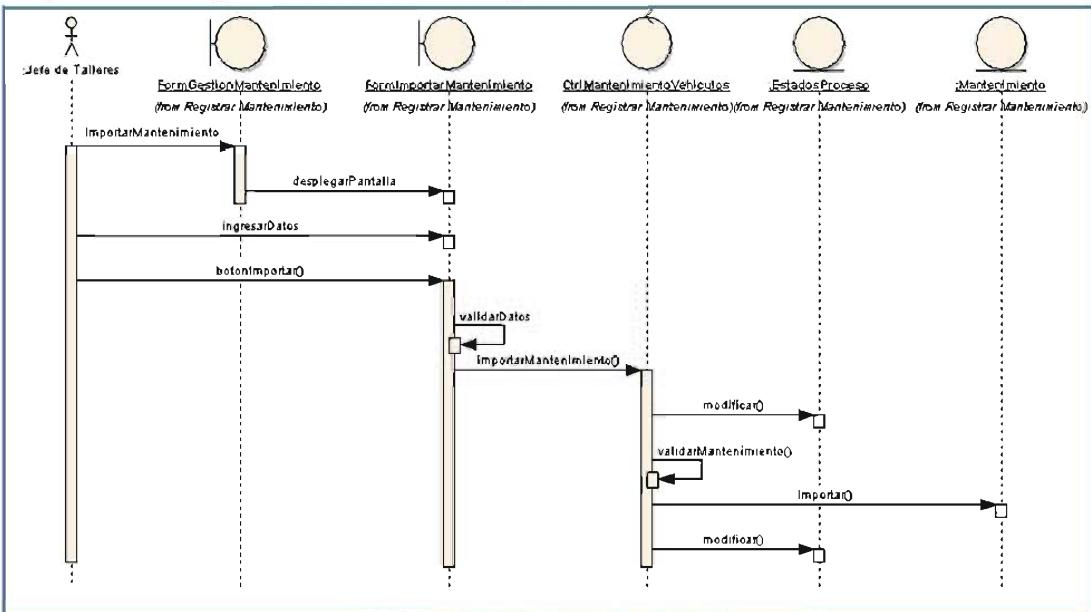


Figura H6. Diagrama de secuencia Registrar odómetro

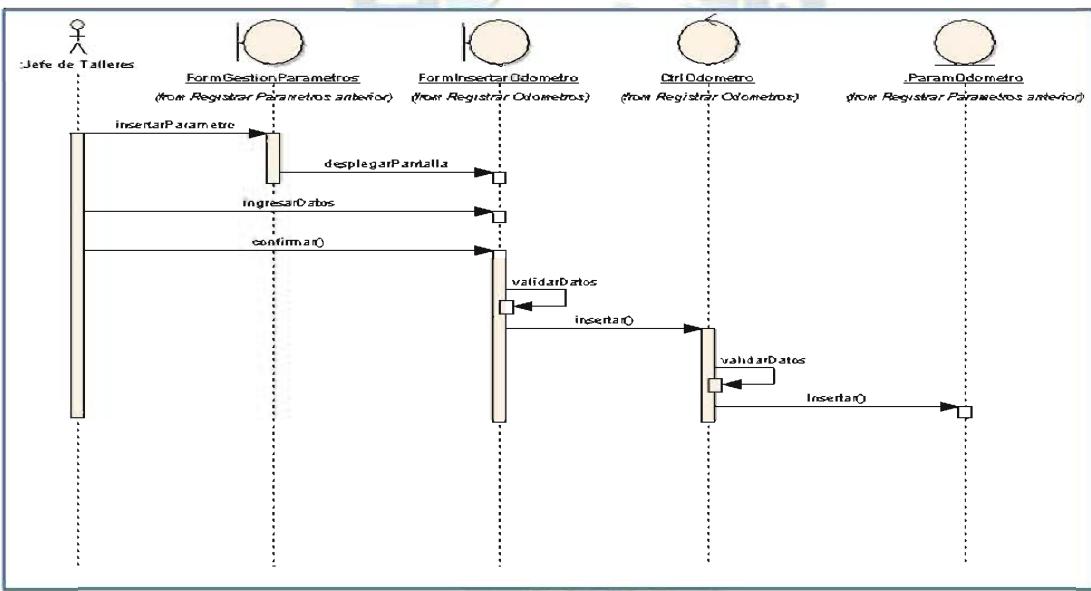


Figura H7. Diagrama de secuencia Registrar parámetros generales

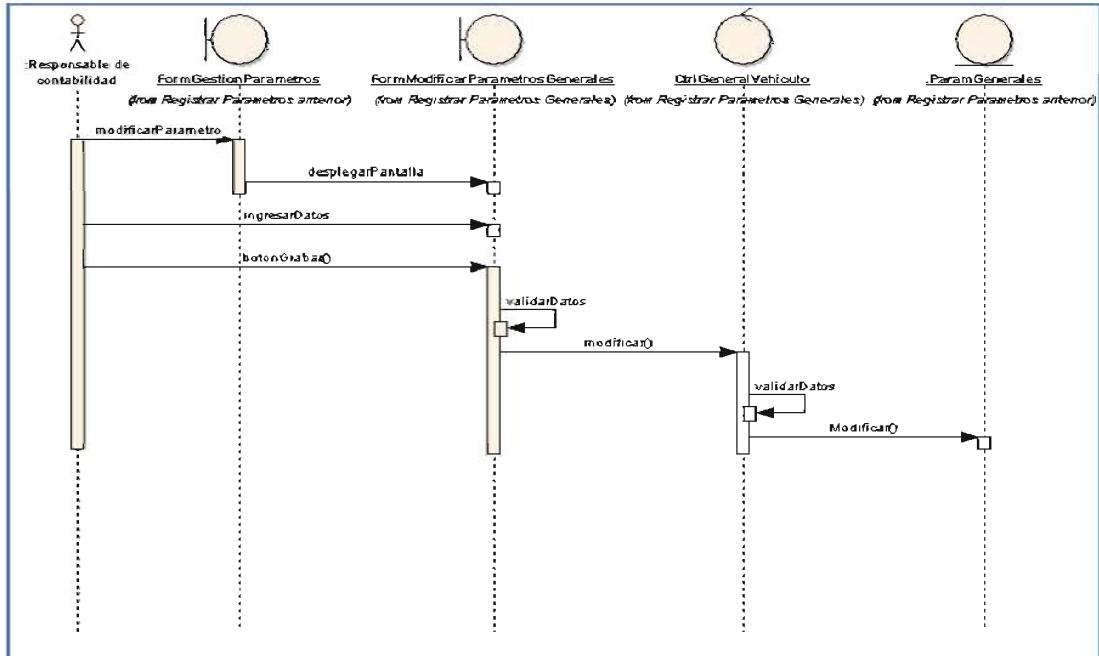


Figura H8. Diagrama de secuencia Parámetros Subgrupo

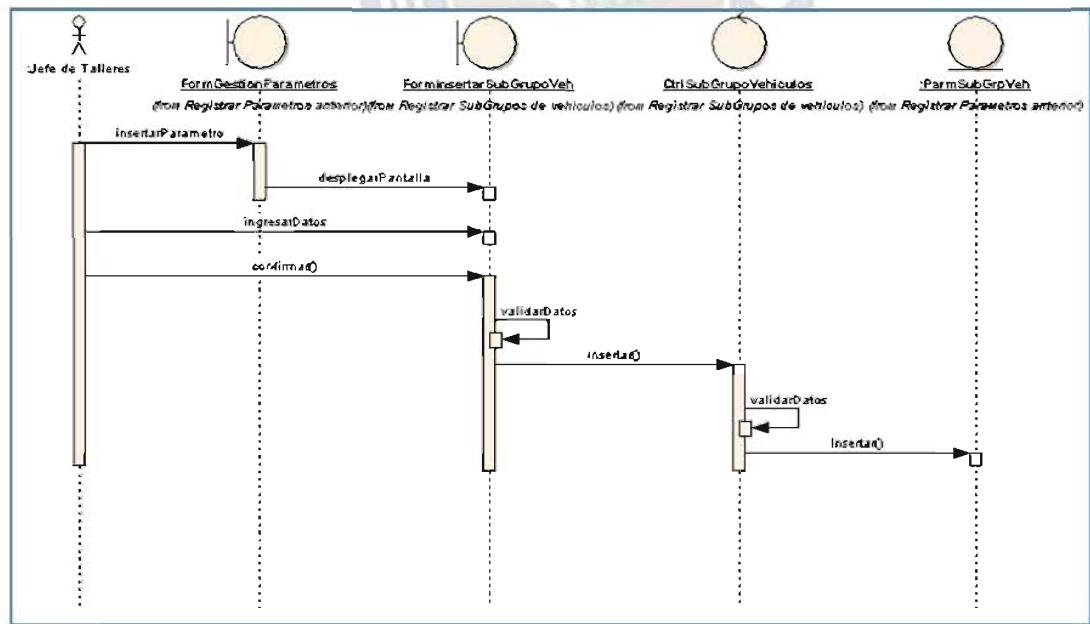


Figura H9. Diagrama de secuencia Parámetros Surtidores

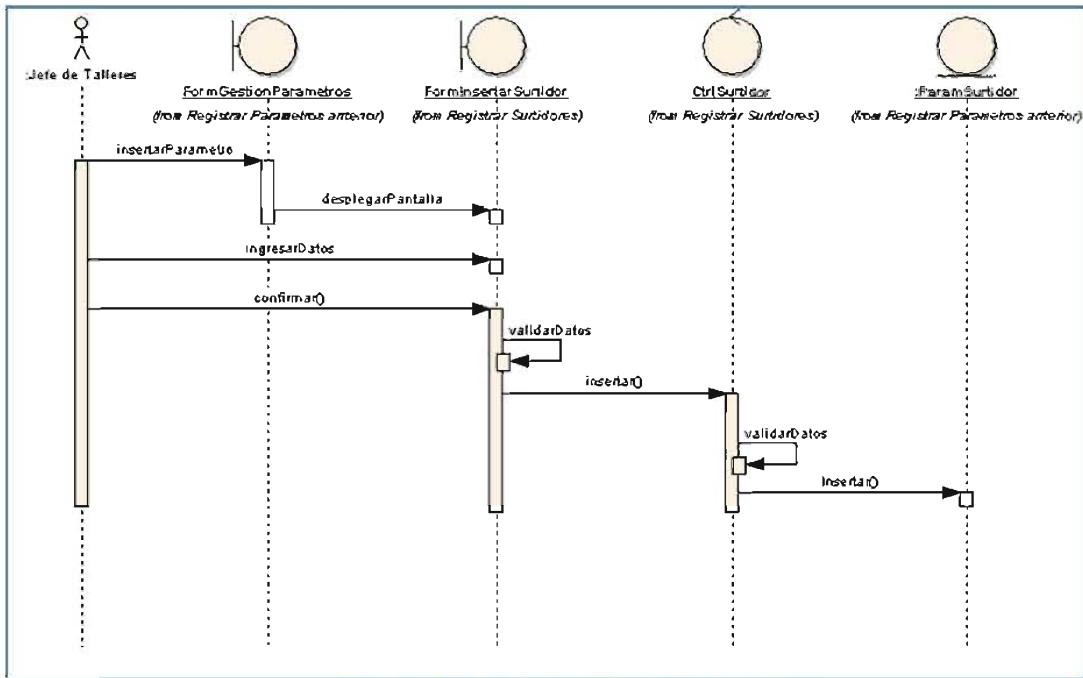


Figura H10. Diagrama de secuencia Parámetros Combustible

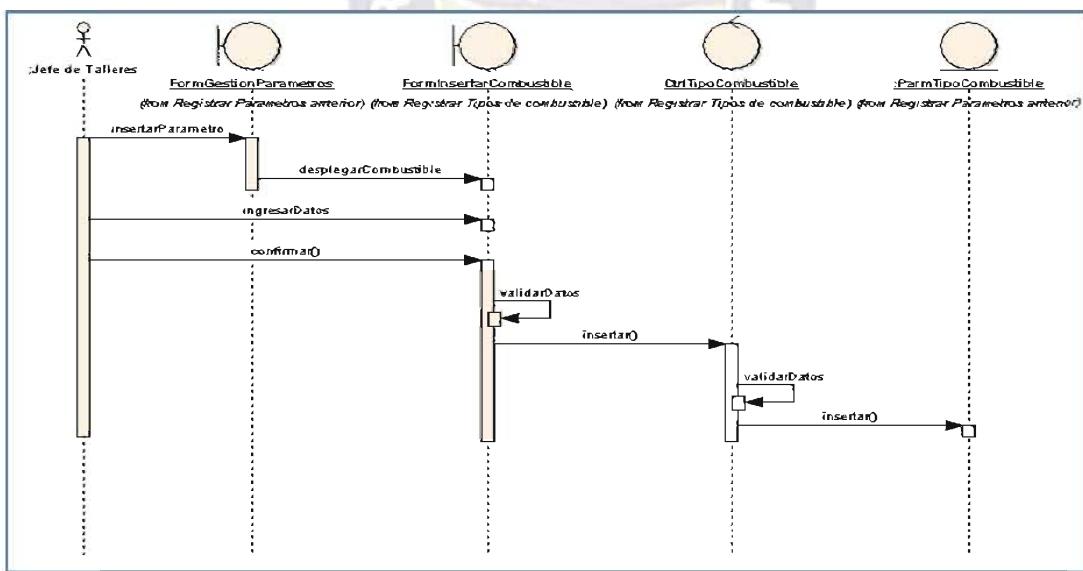


Figura H11. Diagrama de secuencia Revertir cierre contable

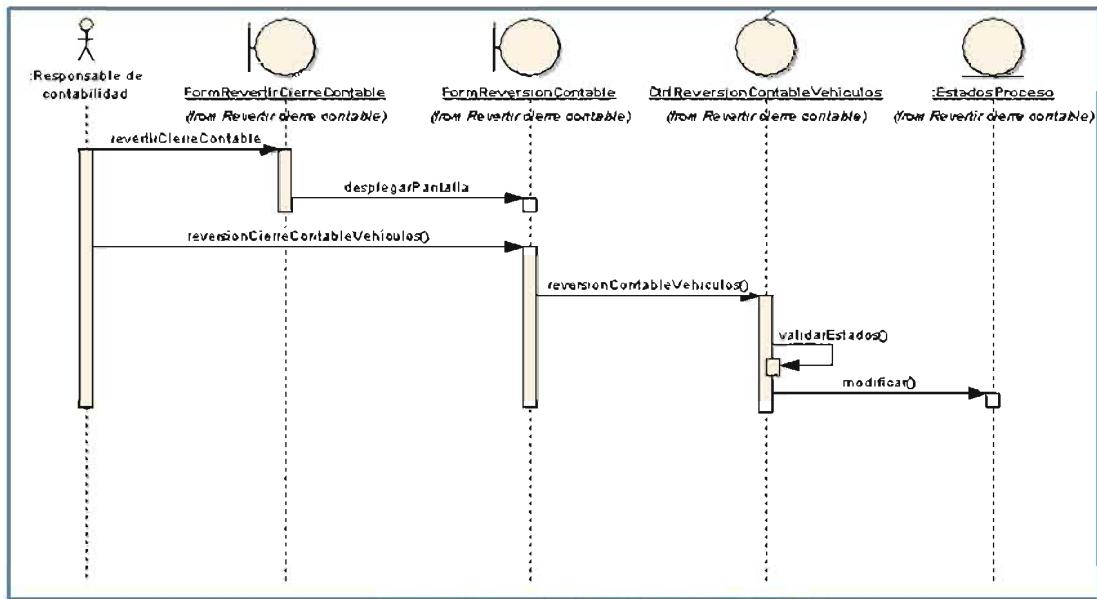


Figura H12. Diagrama de secuencia Revertir cierre vehículos

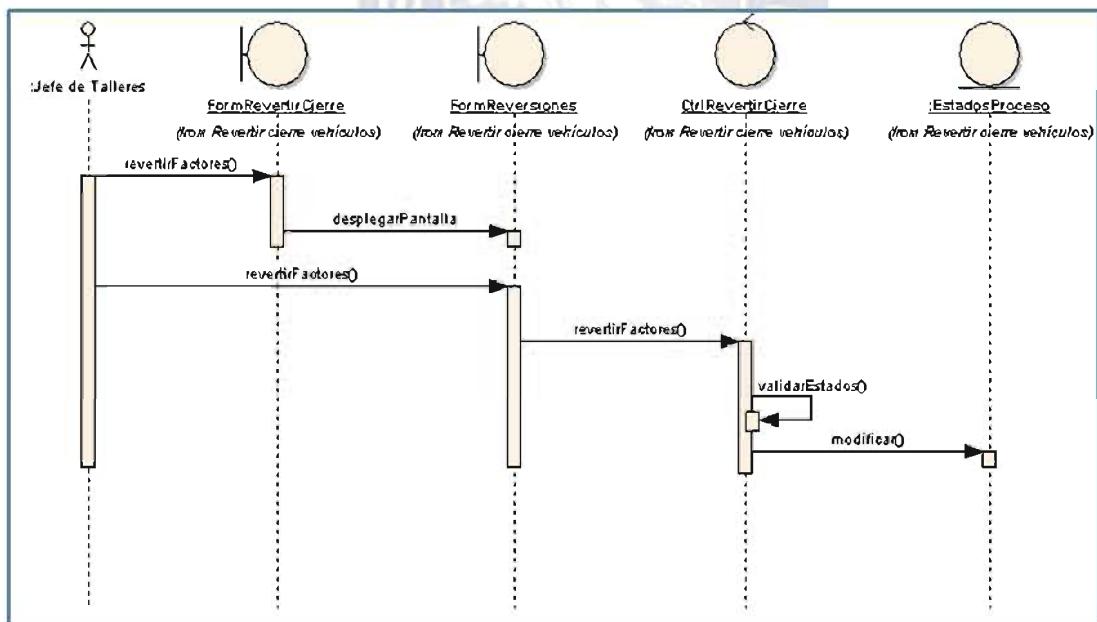


Figura H13. Diagrama de secuencia Distribuir gastos

