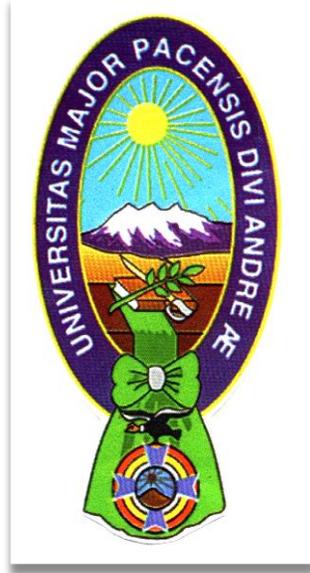


UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

**SISTEMA DE INFORMACION Y CONTROL PARA EL SINDICATO DE GRUAS,
MONTACARGAS Y EQUIPO PESADO "LA PAZ"**

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS

POSTULANTE: CHEJO MAMANI JUAN CARLOS

TUTOR : LIC. EFRAIN SILVA SANCHEZ

REVISOR : LIC. JAVIER REYES PACHECO

LA PAZ - BOLIVIA

2011

INDICE

CAPITULO I

MARCO REFERENCIAL

1.1.	INTRODUCCION.....	1
1.2.	ANTECEDENTES.....	2
1.3.	PLANTEAMIENTO DEL PROBLEMA.....	4
1.3.1.	ANALISIS DEL PROBLEMA.....	4
1.3.2.	DEFINICION DEL PROBLEMA.....	5
1.4.	OBJETIVOS.....	5
1.4.1.	OBJETIVO GENERAL.....	5
1.4.2.	OBJETIVOS ESPECÍFICOS.....	5
1.5.	JUSTIFICACIONES.....	6
1.5.1.	JUSTIFICACION TECNICA.....	6
1.5.2.	JUSTIFICACION ECONOMICA.....	6
1.5.3.	JUSTIFICACION SOCIAL.....	7
1.6.	ALCANCES.....	7
1.7.	APORTES.....	7
1.8.	METODOS Y TECNICAS.....	8

CAPITULO II

MARCO TEORICO

2.1.	INTRODUCCION.....	9
2.1.1.	ANTECEDENTES DEL SINDICATO DE GRUAS, MONTACARGAS Y EQUIPO PESADO "LA PAZ".....	9
2.1.2.	CARACTERISTICAS DE LA INSTITUCION.....	9

2.1.3.	ORGANIGRAMA INSTITUCIONAL.....	10
2.1.4.	APORTES.....	11
2.2.	EL PROCESO DE SOFTWARE.....	11
2.3.	INGENIERÍA DE SOFTWARE.....	11
2.3.1.	ANÁLISIS DE REQUERIMIENTOS.....	12
2.3.2.	ESPECIFICACIÓN.....	12
2.3.3.	ARQUITECTURA.....	13
2.3.4.	PROGRAMACIÓN.....	13
2.3.5.	PRUEBAS.....	13
2.3.6.	DOCUMENTACIÓN.....	14
2.3.7.	MANTENIMIENTO.....	14
2.4.	LENGUAJE DE MODELADO UNIFICADO (UML)	14
2.4.1.	ESPECIFICACIÓN DE CASOS DE USO.....	15
2.4.2.	DIAGRAMAS.....	15
2.4.3.	DIAGRAMAS DE CLASES.....	16
2.4.4.	DIAGRAMAS DE SECUENCIA.....	16
2.4.5.	DIAGRAMAS DE ESTADOS.....	17
2.4.6.	DIAGRAMAS DE COMPONENTES.....	17
2.4.7.	DIAGRAMAS DE DESPLIEGUE.....	18
2.5.	PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP).....	19
2.5.1.	CARACTERÍSTICAS.....	19
2.5.2.	CICLO DE VIDA DE UN PROCESO UNIFICADO.....	20
2.5.3.	FASE DE INICIO.....	21
2.5.4.	FASE DE ELABORACIÓN.....	21
2.5.5.	FASE DE CONSTRUCCIÓN.....	21
2.5.6.	FASE DE TRANSICIÓN.....	21
2.6.	SISTEMAS DE INFORMACIÓN.....	21
2.6.1.	BASE DE DATOS.....	22
2.6.2.	ESTRUCTURA DE UNA BASE DE DATOS.....	22
2.6.3.	VENTAJAS DE LA BASE DE DATOS.....	22
2.7.	PROGRAMACIÓN ORIENTADA A OBJETOS.....	23
2.7.1.	MODELO ORIENTADO A OBJETOS.....	24
2.7.2.	ANÁLISIS ORIENTADO A OBJETOS.....	24
2.7.3.	DISEÑO ORIENTADO A OBJETOS.....	24
2.8.	MODELO VISTA CONTROLADOR (MVC).....	26
2.8.1.	MODELO TRES CAPAS.....	27

2.8.2.	VENTAJAS DEL MVC.....	28
2.9.	HERRAMIENTAS DE PROGRAMACIÓN.....	29
2.9.1.	LENGUAJE DE DESARROLLO EN .NET.....	29
2.9.2.	GESTOR DE BASE DE DATOS SQL SERVER.....	31
2.9.3.	SERVIDOR IIS (INTERNET INFORMATION SERVER).....	31
2.10.	SEGURIDAD.....	32
2.11.	MÉTRICAS DE CALIDAD.....	33

CAPITULO III

MARCO APLICATIVO

a.	FASE DE INICIO.....	36
i.	FUNCIONES PRINCIPALES DEL SINDICATO DE GRUAS Y MONTACARGAS Y EQUIPO PESADO "LA PAZ".....	36
i.	RECOPIACIÓN DE REQUISITOS.....	37
i.	ANÁLISIS.....	38
i.	IMPLEMENTACION.....	38
i.	PRUEBAS.....	38
i.	ANÁLISIS Y GESTIÓN DE RIESGOS.....	39
i.	MODELO DEL NEGOCIO Y ANÁLISIS DE LA SITUACIÓN ACTUAL.....	39
3.1.7.1.	SITUACIÓN ACTUAL.....	39
b.	FASE DE ELABORACION.....	44
3.2.1.	DOCUMENTO DEL ANÁLISIS DEL SISTEMA DE INFORMACION.....	44
3.2.2.	IDENTIFICACIÓN DE ACTORES PARA EL NUEVO SISTEMA.....	45
3.2.3.	MODELO DE CASOS DE USO DEL NUEVO SISTEMA.....	46
3.2.3.1	ESPECIFICACION DE REQUERIMIENTOS.....	46
3.2.3.2	FUNCIONES DEL SISTEMA.....	47
3.3.	DIAGRAMA DE CASOS DE USO.....	47

3.3.1. FUNCIONES BASICAS DEL SISTEMA.....	48
3.4. ANALISIS Y DISEÑO.....	51
3.4.1. DESCRIPCION DE LOS CASOS DE USO EXPANDIDOS.....	51
3.5. ETAPA DE CONSTRUCCION.....	70
3.5.1. DIAGRAMAS DE SECUENCIA PARA EL DISEÑO.....	
3.5.2. DIAGRAMA DE ESTADO.....	70
3.5.3. DIAGRAMA DE COMPONENTES.....	
3.5.4. DIAGRAMA DE DESPLIEGUE.....	73
3.5.5. MODELO ENTIDAD – RELACION	
3.5.6. DIAGRAMA DE CLASES.....	75
3.6. FASE DE IMPLEMENTACION.....	
3.7. PRUEBAS.....	75
3.8. FASE DE TRANSICION.....	
3.9. IMPLEMENTACION DEL SISTEMA.....	76
3.10. INSTALACION DEL SISTEMA.....	77

CAPITULO IV

CALIDAD DE SOFTWARE

1.1 LA NORMA DE CALIDAD ISO 9126.....	85
1.1.3 CONFIABILIDAD.....	85
1.1.4 FUNCIONALIDAD.....	86
1.1.5 USABILIDAD.....	90
1.1.6 FACILIDAD DE MANTENIMIENTO.....	91
1.1.7 EFICIENCIA.....	92
1.1.8 PORTABILIDAD.....	92
1.2 MODELO CONSTRUCTIVO DE COSTES.....	93

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1.	CONCLUSIONES.....	95
5.2.	RECOMENDACIONES.....	96

REFERENCIA BIBLIOGRAFICA

ANEXOS

DOCUMENTACION

INDICE DE FIGURAS

Fig. 2.1 Organigrama de la empresa.....	10
Fig. 2.2 Ejemplo de casos de uso.....	15
Fig. 2.3 Diagrama de Secuencias.....	16
Fig. 2.4 El proceso del modelado unificado.	20
Fig. 2.5 Arquitectura de 3 capas.....	28
Fig. 3.1: Diagrama de la Situación Actual.....	40
Fig. 3.2: Diagrama de Casos de Uso: del Nuevo Sistema.....	47
Fig. 3.3: Diagrama de Casos de Uso: Autenticación de Cuentas	51
Fig. 3.4: Diagrama de Secuencia: Registra cuenta de usuario	52
Fig. 3.5: Diagrama de Secuencia: Elimina cuenta de usuario	53
Fig. 3.6: Diagrama de Secuencia: Asigna prioridad de usuario	54
Fig. 3.7: Diagrama de Casos de Uso: Registrar cuenta de afiliación	55
Fig. 3.8: Diagrama de secuencia: Registro de Socios.....	56
Fig. 3.9: Diagrama de secuencia: Registro de Vehículos.....	58
Fig. 3.10 Diagrama de secuencia: Adicionar (Socios – Vehículos).....	59
Fig. 3.11 Diagrama de secuencia: Modificar (Socios – Vehículos).....	60
Fig. 3.12 Diagrama de secuencia: Eliminar (Socios – Vehículos).....	61
Fig. 3.13 Diagrama de Casos de Uso: Registrar aportes de afiliación.....	62
Fig. 3.14 Diagrama de secuencia: Registrar aporte de ingreso.....	63
Fig. 3.15 Diagrama de secuencia: Registrar aportes temporales.....	65
Fig. 3.16 Diagrama de secuencia: Registrar aportes de sanciones o accidentes.....	66
Fig. 3.17 Diagrama de secuencia: Registrar faltas y accidentes	67
Fig. 3.18 Diagrama de secuencia: Registra faltas.....	68
Fig. 3.19 Diagrama de secuencia: Registra accidentes.....	69
Fig. 3.20 Diagrama de secuencia: Autenticación de cuentas.....	70
Fig. 3.21 Diagrama de secuencia: Registro de afiliación de socios.....	71
Fig. 3.22 Diagrama de secuencia: Registro de Aportes de Afiliación.....	72
Fig. 3.23 Diagrama de secuencia: Registro faltas y accidentes	73
Fig. 3.24 Diagrama de Estado: Autenticación de Cuentas	74
Fig. 3.25 Diagrama de Estado: Registrar Aportes	74
Fig. 3.26 Diagrama de Estado: Registrar Afiliación... ..	74
Fig. 3.27 Diagrama de Componentes del Sistema	75
Fig. 3.28 Diagrama de Despliegue: Base de Datos - Sistema	76
Fig. 3.29 Modelo Entidad – Relación	77
Fig. 3.30 Diagrama de Clases	78
Fig. 3.31 Página de Inicio de Interfaz con el sistema	79
Fig. 3.32 Registrar nuevo socio.....	80

Fig. 3.33 Eliminar Socios.....	81
Fig. 3.34 Actualizar socio.....	82

INDICE DE TABLAS

Tabla 3.1: Funciones del nuevo sistema	47
Tabla 3.2: Atributos del nuevo sistema en las especificaciones de funciones	49
Tabla 4.1: Descripción de parámetros de medición.....	87
Tabla 4.2: Métricas de Puntos de Función.....	88
Tabla 4.3: Valores de ajuste de complejidad.....	89
Tabla 4.4: Evaluación del sistema.....	90

RESUMEN

El manejo y la organización de la información por entidades públicas o privadas es un tema que amerita atención, control y protección. Por ello, es que varias instituciones se encuentran prestas a incluir entre sus activos sistemas de información acorde a sus necesidades.

Entre ellas se distingue el sindicato de Grúas, Montacargas y Equipo Pesado "La Paz", ubicada en la ciudad de El Alto, siendo su principal actividad de dar servicio a empresas e industrias que manejan equipos pesados, apoya con sus medios de transporte que en la mayoría son grúas que ayudan al montaje, equipamiento y traslado de estos equipos pesados. Su actividad es todo lo que se refiere a la logística del transporte, como es el de cargar y descargar con sumo cuidado el objeto pesado que solamente una grúa o varias podrían hacerlo, valiéndose de herramientas y transporte que la empresa dispone.

Debido a la carencia de personal y recursos limitados, el sindicato de Grúas, Montacargas y Equipo Pesado "La Paz" se vio restringida a trabajar con procesos manuales, almacenando toda la información a su cargo, en archivadores y hojas electrónicas de Excel, lo cual provoca retraso en la entrega de documentos e informes al personal de Administración y duplicación de datos.

Ante tal necesidad se adoptó la decisión de desarrollar el presente Proyecto, con el objeto de que el área de Servicios de la empresa cuente con una aplicación flexible y confiable que le permita almacenar, consultar y actualizar datos de los vehículos, personal encargado de operar los vehículos, los técnicos, clientes y los servicios realizados con los vehículos.

Para el desarrollo del proyecto se utilizó la metodología RUP, con el lenguaje de modelado UML, también se aplicó el lenguaje Visual .Net y la base de datos Sql Server.

Terminado el trabajo, el sistema obtenido cumplió con el objetivo trazado, y se encuentra enfocado a convertirse no solo en una aplicación más, sino en una herramienta de utilidad en el Área de Servicios.

AGRADECIMIENTOS

- Agradezco a la persona que hizo posible todo esto Marysol, a mis padres Raulin y July por todo el apoyo incondicional que me dieron. Al destino que me dio todo lo que puedo esperar, a mi hijo Fenrir mi más bello regalo
- También quiero agradecer de manera muy especial al Lic. Victor Arce Lecoña que es el Secretario Ejecutivo de el sindicato de Grúas, Montacargas y Equipo Pesado "La Paz", por darme la oportunidad de mostrar mis conocimientos adquiridos durante mis estudios, desarrollando este trabajo que beneficia de alguna manera al sindicato, y por darme los consejos que siempre tomaré en cuenta
- Particularmente quiero agradecer a mi Docente Tutor el Lic. Efraín Silva Sánchez a quien debo las ideas constructivas y recomendaciones sobre el desarrollo de este trabajo.
- Agradezco a mi Docente Revisor el Lic. Javier Reyes Pacheco por toda la dedicación y paciencia que tuvo para explicarme y revisar mi proyecto durante toda la etapa del desarrollo del trabajo. Y por sus consejos y sugerencias desinteresadas que me serán muy importantes toda mi vida.
- Deseo agradecer también a los docentes de la Carrera de Informática, porque me dieron los conocimientos básicos y la oportunidad de cursar sus materias y así lograr ganarme el derecho de ser profesional.

CAPITULO 1

MARCO REFERENCIAL

1.1. INTRODUCCION

Para aumentar el nivel de competencia, las empresas van evolucionando con las técnicas de administración de sus recursos, permitiendo y conociendo nuevas tecnologías. Es por esto que deben tomar en cuenta la automatización de los procesos de negocio, por lo que necesitarán sistemas de información que puedan dar conocimiento, agilizar y mejorar las transacciones comerciales que se realizan dentro de las empresas, a la vez de dar interés a los que son consumidores de la empresa.

En el presente proyecto que será desarrollado, la automatización de las funciones dentro de la empresa es para aquellas actividades de comercio que realiza y que proporciona apoyo a la toma de decisiones. Proponiendo un Sistema de Información de Prestación de Servicios de Maquinaria Pesada para una empresa de servicios de Transporte pesado es una solución adecuada y elaborada para garantizar la eficiencia. Y sabiendo que en la empresa se tiene los servicios de medios de transporte pesado, los servicios son parte fundamental de esta en la existencia y tener un adecuado control de la prestación de servicios que la empresa proporciona a sus clientes mediante sus medios de transporte.

El sindicato de Grúas, Montacargas y Equipo Pesado "La Paz" siendo su principal actividad de dar servicio a empresas e industrias que manejan equipos pesados, apoya con sus medios de transporte que en la mayoría son grúas que ayudan al montaje, equipamiento y traslado de estos equipos pesados. Su actividad es todo lo que se refiere a la logística del transporte, como es el de cargar y descargar con sumo cuidado el objeto pesado que solamente una grúa o varias podrían hacerlo, valiéndose de herramientas y transporte que la empresa dispone.

El sindicato de Grúas, Montacargas y Equipo Pesado "La Paz" es una afiliación de varias empresas de servicio de grúas, montacargas y equipo pesado que cuenta con variedad de herramientas y equipos de transporte y un sinnúmero de clientes que optan por los servicios

de las empresas, es por tal motivo la implementación de un sistema que automatice sus actividades manuales en el área de comercialización de sus servicios, también del control de sus transportes, clientes y empleados; así se podrá tener un control eficaz de las transacciones que realiza el sindicato.

1.2. ANTECEDENTES

El sindicato de Grúas, Montacargas y Equipo Pesado "La Paz", se encarga de la administración de todo el personal que se encuentra afiliado a éste y de la información vehicular que tiene cada afiliado. Para este trabajo necesita la ayuda de personal capacitado, con conocimientos y habilidades apropiadas en el área.

El sindicato está formado por recursos humanos y por los siguientes transportes: Montacargas, Grúas, Camiones Grúas, Remolques, Camiones de Transporte, Loboys, materiales y herramientas necesarias para la posible manipulación de maquinaria pesada que necesiten sus clientes y desde la creación del sindicato constantemente incrementa el reconocimiento de diferentes firmas, esto debido a que existen pocas empresas de servicio que tengan estos tipos de transporte pesado y es el único y primer sindicato que existe en la ciudad de La Paz.

Los transportes y montajes que realiza el personal que se halla afiliado al sindicato satisface ampliamente las necesidades de servicio a sus clientes como ser: fábricas, industrias, entidades bancarias y empresas pequeñas o individuales. Toda la maquinaria pesada y equipo industrial que forma parte del sindicato de Grúas, Montacargas y Equipo Pesado "La Paz" es de gran importancia y utilidad para sus clientes, colocando su confianza en esta empresa de servicios y siendo clientes asiduos.

En cuanto a proyectos relacionados con prestación de servicios y control de transportes, se mencionan algunos junto con una lista de problemas, que fueron desarrollados en la carrera de Informática, de la Universidad Mayor de San Andrés:

- Sistema de Gestión Vehicular para el Sindicato de Transporte Mixto Simón Bolívar, realizado por José Hernán Yujra Magnani, que es un sistema que proporciona

información rápida y coherente tanto al área administrativa como en el área de recursos humanos y técnicos (vehículos).

Se encontraron los siguientes problemas:

- Control inadecuado en el registro de documentación interna del sindicato.
- Retraso en la entrega de documentos solicitados por los pocos funcionarios que administran la entidad.
- Poca confiabilidad en el manejo de información.

➤ Sistema de Control y Seguimiento del uso de vehículos vía Web para la empresa Cadeb, realizado por Nidia Ingrid Martínez Larico, es un sistema que permite administrar la información de manera eficiente, agilizando y mejorando los procesos que se realizan en la institución para una buena toma de decisiones.

Los problemas encontrados fueron:

- 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, porque se lo hace manualmente ocasionando errores en los datos.
- Obtención de reportes lento y no es oportuna para tomar decisiones.
- La información que se envía del área de mantenimiento es manual.

➤ Sistema de Distribución, Ventas y Control de Inventarios para Sobolma Ltda., realizado por Gregorio Cala Cori, el proyecto genera información y ayuda a la administración de los responsables de la fábrica en el control de distribución, ventas e inventarios que se tiene en el almacén, la fábrica realiza la industrialización de la madera, creando tableros, láminas y tablas.

Los problemas que se encontraron fueron:

- La recolección de la información es tediosa y lenta, y esto provoca muchos retrasos en los diferentes procesos.
- Las ventas que se realizan de los diferentes productos se realizan de forma inadecuada esto provoca una confusión en el control de la existencia de productos.
- La entrega de productos a los distribuidores tiene una demora de bastante tiempo debido a que no se cuenta con un registro adecuado de los pedidos.

- Hay confusión en la entrega de productos tanto a los clientes como a los distribuidores y esto genera molestia en ellos.
 - La atención a los clientes como a los distribuidores lo realizan entre varios empleados y cada uno tiene su respectivo apunte que no es actualizado esto causa perdida de tiempo al registrar todos los movimientos realizados en el día.
 - No se tiene un Kardex actualizado de los productos vendidos, ni distribuidos por lo que hay una demora en los procesos de inventarios.
- Sistema de control de Ventas e Inventarios para la empresa H & S, realizado por Erick Peñaranda Rocha; dicho sistema mejora el control de la información de las ventas e inventarios de los productos, optimizando los procesos de comparación y reportes para los departamentos de ventas, producción, comercialización y finanzas.

Se identificaron los siguientes problemas:

- La institución requiere informes estadísticos acerca de la oferta y demanda de los artículos para realizar una adquisición más eficiente.
- Dificultad y lentitud en la manipulación de la información almacenada en hojas electrónicas Excel referente a la venta y cotización de equipos y materiales.
- Demora en la obtención de reportes con referente a la venta del material disponible.
- No existe un control de ventas, lo que impide una buena toma de decisiones respecto a las distintas operaciones que se realiza.
- Demora del vendedor al generar los pedidos en el formulario de forma manual, buscando códigos de clientes y de productos.
- Incertidumbre de parte de los vendedores respecto a los productos disponibles existentes en depósito.

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. ANALISIS DEL PROBLEMA

La información con la que cuenta el sindicato de Grúas, Montacargas y Equipo Pesado "La Paz" sobre la gestión administrativa de recursos humanos y vehiculares es muy abundante, porque esta entidad alberga varios tipos de transporte de equipo pesado,

por lo tanto requiere de reportes diarios y detallados sobre las actividades que ejecutan. Por ello con el diagnóstico realizado, se tienen los siguientes problemas:

- No existen registros ordenados y actualizados de los datos de los recursos humanos, el proceso de registro es manual, provocando duplicación y demora en la búsqueda de sus datos.
- El registro de inventarios de los vehículos lo manejan en hojas electrónicas de Excel, lo que evita un control adecuado y consistente de estas herramientas.
- Existe demora al realizar informes periódicos del reporte de los recursos humanos y de los distintos vehículos.
- No existen registros adecuados ni actualizados de los recursos humanos, ni de sus vehículos, esto ocasiona confusiones y una mala organización de los registros.

1.3.2. DEFINICION DEL PROBLEMA

De acuerdo al análisis de los problemas que surgieron se define el problema principal:

¿De qué manera se puede agilizar, actualizar, controlar y consultar los procesos del manejo de registros de recursos humanos y de sus vehículos para tener una mejor información y administración eficaz, oportuna y confiable de sus elementos y servicios que presta el Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz"?

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un Sistema de Información de administración y control que permita proporcionar información rápida y mejorar la administración y buen control en el área de recursos humanos y de vehículos de manera eficaz para el Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz".

1.4.2. OBJETIVOS ESPECÍFICOS

- Desarrollar módulos de administración y control que permitan consultar, actualizar y administrar la información.

- Desarrollar un módulo de registro de los recursos humanos afiliados al sindicato para generar consultas y reportes según los requerimientos encontrados en la entidad.
- Diseñar una base de datos que brinde seguridad en cuanto a la información almacenada de los recursos humanos y vehiculares.
- Desarrollar una interfaz Web informativa acerca de todos los servicios que presta cada vehículo.
- Modulo de registros y reportes acerca de los vehículos (maquinaria pesada).

1.5. JUSTIFICACIONES

1.5.1. JUSTIFICACION TECNICA

Para lograr el desarrollo del sistema, la empresa cuenta con equipos de computación y herramientas disponibles para la implementación del proyecto propuesto y desea adecuarse a la nueva tecnología para mejorar sus servicios y crecer, teniendo todos los recursos tecnológicos disponibles tanto en Hardware como en Software.

Por tanto, el sistema a desarrollar permitirá soportar de forma adecuada los posibles cambios y adiciones de nuevos módulos de operación debido a la flexibilidad en el desarrollo de información, usando tecnología de punta y software adecuado.

1.5.2. JUSTIFICACION ECONOMICA

Se justifica económicamente el proyecto porque proporcionará mayores beneficios a corto y largo plazo que se observará a través de la publicidad que se realizará mostrando una interfaz informativa del sistema en la Web.

El proceso de control, tendrá como resultado una reducción de tiempos y esfuerzos, permitiendo a los miembros de la administración que puedan generar y consultar información en tiempo real obteniendo información confiable e inmediata, lo que implica un ahorro de tiempo e incremento de los beneficios en el control eficiente en las cobranzas e incremento de clientes.

Para los clientes será un beneficio la implementación del proyecto porque podrán reducir los costos y el tiempo en el momento que busquen información necesaria del tema debido a que el costo de navegar en Internet es bastante reducido.

1.5.3. JUSTIFICACION SOCIAL

Al finalizar el proyecto, los que se beneficiarán serán los administradores de los servicios, comercializadores y al personal encargado de la atención al cliente, porque podrán contar con información rápida, accesible con lo cual se generará un ámbito de satisfacción, y a la vez esta información estará disponible para los gerentes y éstos tendrán criterios de decisión y solución de las tomas de decisiones.

Por otro lado los clientes de la empresa serán beneficiados por contar con una interfaz web donde puedan obtener información sobre los servicios de transporte en cualquier momento que sea necesario para ellos, con la opción de realizar pedidos con anticipación, enviando mensajes mediante el sistema.

1.6. ALCANCES

Con respecto al sistema de información de los servicios los alcances del proyecto serán:

- Los alcances del presente proyecto cubren la realización del control de recursos humanos y vehiculares para el sindicato de Grúas, Montacargas y Equipo Pesado "La Paz", que se encuentra en la región de La Paz, encontrándose la empresa en el sector de El Alto.
- La realización del registro y reportes de recursos humanos afiliados al sindicato y vehiculares disponibles.
- El sistema registrará los documentos como: registros de vehículos, registro de afiliados de recursos humanos, modificación, eliminación, según las categorías a las que pertenecen.

1.7. APORTES

En los aportes para el sindicato, es dar mejor respuesta y consolidación en la atención puesto que contará con un sistema de gestión vehicular para administrar de forma eficiente a la entidad.

Un aporte principal es el mismo proyecto, debido a que se automatizará varias acciones que se realizan manualmente y registros hechos en hojas electrónicas de Excel, el proyecto permitirá el buen control, manejo y a la vez mayores incrementos monetarios y la disminución del tiempo de registros y control.

Con el sistema se logrará tener una mejor administración y decisión sobre los servicios y de las maquinarias que son las principales herramientas de trabajo pertenecientes al sindicato, es de una gran importancia tener el control de los recursos.

Otro aporte importante, es realizar la publicidad que tendrá el sindicato mediante Web, esto incrementará el nivel de clientes para las varias empresas afiliadas a éste.

1.8. METODOS Y TECNICAS

Para poder alcanzar los objetivos propuestos en el presente proyecto se recurrirá a los siguientes métodos y técnicas.

En la recopilación de información se hará uso de técnicas de Investigación, como ser:

- Entrevistas
- Cuestionarios
- Revisión de documentos

Para el análisis del problema se utilizará:

- La matriz del marco lógico.
- El árbol de problemas.
- El árbol de objetivos.

Como metodología de desarrollo de software se empleará:

- La metodología de desarrollo RUP (Proceso Unificado de Rational).
- Y la herramienta UML (El Lenguaje Unificado de Modelado).

CAPITULO II

MARCO TEORICO

2.1. INTRODUCCION

El presente capítulo detalla algunos de los conceptos y definiciones que se necesitan para el desarrollo del proyecto, con el fin de ayudar a la familiarización del tema y son la base para la descripción y explicación del problema a solucionar...

Es importante que consideremos el camino más adecuado, en la utilización de herramientas, métodos y técnicas que permita el análisis, para lograr alcanzar los objetivos planteados en el anterior capítulo.

2.1.1. ANTECEDENTES DEL SINDICATO DE GRUAS, MONTACARGAS Y EQUIPO PESADO "LA PAZ"

El sindicato La Paz es una institución de servicios de transporte de todo tipo de maquinaria pesada, para el beneficio de la sociedad paceña, es una entidad profundamente vinculada al campo de la ayuda, unión y apoyo de las empresas afiliadas que dan el servicio de maquinaria pesada en cuanto se refiere a grúas, montacargas, lowboys y todo tipo de maquinaria pesada.

Mencionamos que el sindicato de grúas, montacargas y equipo pesado La Paz, cuenta con alrededor de veinte socios afiliados, estos afiliados son los dueños propietarios de cada empresa de servicios de maquinaria pesada, también cuenta con los vehículos (que son las maquinarias pesadas de servicio) que están registrados de acuerdo a la pertenencia de cada socio y cuenta con siete personas que administran las funciones y tareas del sindicato.

2.1.2. CARACTERISTICAS DE LA INSTITUCION

El sindicato La Paz funciona desde hace siete años, con varios propósitos según su estatuto orgánico:

- Mantener la unidad sindical de todos los socios contra cualquier intento de división que atente en contra de la entidad.

- Luchar por las cotizaciones justas y balanceadas que tienen que ofrecer cada socio afiliado a su respectivo cliente al que le otorgará el servicio de grúas.
- Mantener la unidad entre los afiliados de la institución, incentivando a la capacitación y al deporte con la finalidad de conseguir mejores condiciones de vida y amistad.
- Realizar los movimientos económicos y financieros posibles para alcanzar el bien común, para establecer formas de recapitalización de la economía del socio propietario.
- Precautelar los bienes habidos y por haber de económicos.
- Los aportes que realizan los socios también es para cuidar y mantener al sindicato y sus oficinas, como también para apoyar a los socios en caso de que exista algún accidente.

2.1.3. ORGANIGRAMA INSTITUCIONAL

Fig. 2.1. Organigrama del Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz"



Fuente: Sindicato de grúas, montacargas y equipo pesado "La Paz"

El secretario general tiene la principal función de administrar a todo el sindicato, es el indicado para afiliar o no a un nuevo socio. Representa a todo el sindicato, debe hacer cumplir todos los reglamentos y estatutos que el directorio aporte, también representa ante las autoridades del gobierno, municipalidades, organismo operativo de tránsito, en todo acto sindical, cívico, social, cultural, deportivo y de carácter público.

El secretario general también tiene la obligación de velar por los intereses del sindicato y por el constante progreso y bienestar de sus asociados. Conocer todos los casos donde existan conflictos entre socios, o ya sea por cuestiones de servicios.

2.1.4. APORTES

- **Aporte de ingreso:** Todo socio nuevo que quiera ingresar al sindicato, tiene la obligación de realizar el aporte de ingreso estipulado en el reglamento interno. El aporte de ingreso de un socio es también de poder hacer ingresar a cada vehículo perteneciente al socio, por lo tanto el socio debe aportar el ingreso de dinero por cada vehículo perteneciente a éste.
- **Aporte temporal:** El secretario de hacienda registran los aportes ya sean semanales o mensuales de cada socio afiliado, esto es para subvencionar gastos de mantenimiento para el sindicato, auxiliar a otros en caso de accidentes, para incentivar al deporte, etc.
- **Aporte de sanciones o accidentes:** En caso de que los socios cometan alguna infracción.

2.2. EL PROCESO DE SOFTWARE

Un proceso de software define el enfoque que se toma cuando dicho software es tratado por la ingeniería. Además la ingeniería de software también comprende las tecnologías que tiene el proceso como los métodos técnicos y las herramientas automatizadas. El proceso que se adopte, dependerá del tipo de software que se quiera desarrollar. Sin embargo la calidad, oportunidad y viabilidad a largo plazo del producto que se está construyendo, son los mejores indicadores de la eficiencia del proceso que se utiliza [PRESSMAN- 2005].

2.3. INGENIERÍA DE SOFTWARE

La ingeniería de software es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

Esta ingeniería trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos, o desarrollos Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a infinidad de áreas: negocios, investigación

científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, Internet, Intranet, etc. Según autores tienen las siguientes definiciones:

- Ingeniería de Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software (Zelkovitz, 1978).
- Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como Desarrollo de Software o Producción de Software (Bohem, 1976).

Se requiere llevar a cabo numerosas tareas, estas etapas son:

2.3.1. ANÁLISIS DE REQUERIMIENTOS

Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se plasma en el documento ERS, *Especificación de Requerimientos del Sistema*, cuya estructura puede venir definida por varios estándares. Asimismo, se define un diagrama de Entidad/Relación, en el que se plasman las principales entidades que participarán en el desarrollo del software.

La captura, análisis y especificación de requisitos (incluso pruebas de ellos), es una parte crucial; de esta etapa depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos procesos de trabajo para estos fines. Aunque aún no está formalizada, ya se habla de la Ingeniería de requisitos.

2.3.2. ESPECIFICACIÓN

La Especificación de Requisitos describe el comportamiento esperado en el software una vez desarrollado. Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio (definidas por la alta dirección), así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requisitos del software.

Entre las técnicas utilizadas para la especificación de requisitos se encuentran: los Casos de Uso.

2.3.3. ARQUITECTURA

La integración de infraestructura, desarrollo de aplicaciones, bases de datos y herramientas gerenciales, requieren de capacidad y liderazgo para poder ser conceptualizados y proyectados a futuro, solucionando los problemas de hoy. El rol en el cual se delegan todas estas actividades es el del Arquitecto. El Arquitecto de Software es la persona que añade valor a los procesos de negocios gracias a su valioso aporte de soluciones tecnológicas. La Arquitectura de Sistemas en general, es una actividad de planeación, ya sea a nivel de infraestructura de red y hardware, o de Software. La Arquitectura de Software consiste en el diseño de componentes de una aplicación (entidades del negocio), generalmente utilizando patrones de arquitectura. El diseño arquitectónico debe permitir visualizar la interacción entre las entidades del negocio y además poder ser validado, por ejemplo por medio de diagramas de secuencia. Un diseño arquitectónico describe en general el cómo se construirá una aplicación de software. Para ello se documenta utilizando diagramas.

2.3.4. PROGRAMACIÓN

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

2.3.5. PRUEBAS

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; sin perjuicio de lo anterior el programador debe hacer sus propias pruebas. En general hay dos grandes formas de organizar un área de pruebas, la primera es que esté compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma se evalúa que la documentación entregada sea de calidad, que los

procesos descritos son tan claros que cualquiera puede entenderlos y el software hace las cosas tal y como están descritas. El segundo enfoque es tener un área de pruebas conformada por programadores con experiencia, personas que saben sin mayores indicaciones en qué condiciones puede fallar una aplicación y que pueden poner atención en detalles que personal inexperto no consideraría.

2.3.6. DOCUMENTACIÓN

Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por modelaciones (UML), diagramas, pruebas, manuales de usuario, manuales técnicos, etc; todo con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

2.3.7. MANTENIMIENTO

Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de 2/3 de toda la ingeniería de software tiene que ver con dar mantenimiento. Una pequeña parte de este trabajo consiste en arreglar errores, o *bugs*. La mayor parte consiste en extender el sistema para hacer nuevas cosas. De manera similar, alrededor de 2/3 de toda la ingeniería civil, arquitectura y trabajo de construcción es dar mantenimiento.

2.4. LENGUAJE DE MODELADO UNIFICADO (UML)

“El Lenguaje de Modelado Unificado UML es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software.”

El UML es el Lenguaje de Modelado Unificado Orientado a Objetos, UML no es un método porque no tiene noción de proceso el cual es una parte importante de un método.

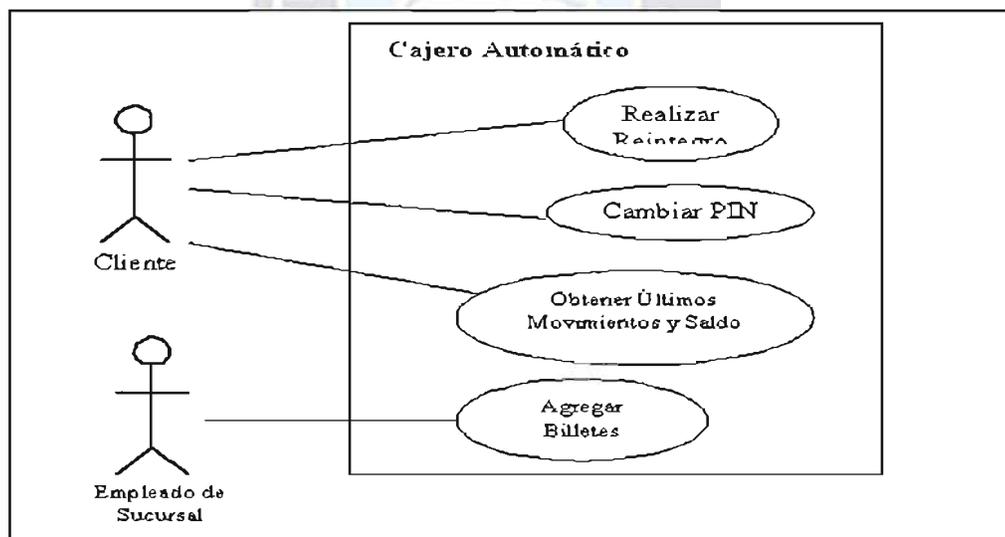
El desarrollo de sistemas con UML siguiendo el proceso unificado incluye actividades específicas, cada una de ellas a su vez contienen otras subactividades las cuales sirven como una guía de cómo deben ser las actividades desarrolladas y secuenciadas con el fin de obtener sistemas exitosos; consecuentemente el desarrollo de los sistemas puede variar de

desarrollador en desarrollador, de proyecto en proyecto, de empresa en empresa adoptando siempre un Proceso de Desarrollo.

2.4.1. ESPECIFICACIÓN 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 se muestra un ejemplo de Diagrama de Casos de Uso para un cajero automático.

Figura. 2.2 Ejemplo de casos de uso



Fuente: [Javier Ferré - 2004]

2.4.2. DIAGRAMAS

Existen varios diagramas para el modelado de sistemas, se mencionará a continuación, los más importantes.

2.4.3. DIAGRAMAS DE CLASES

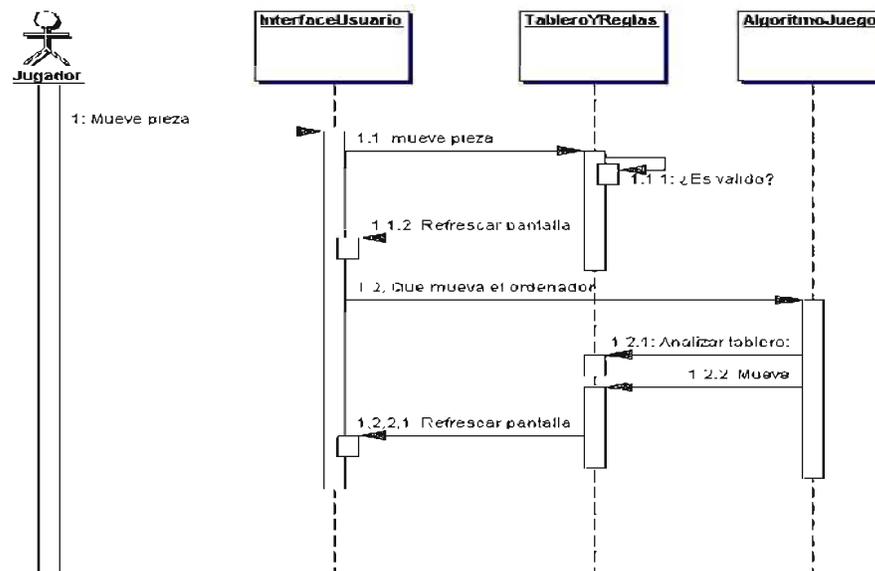
Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

2.4.4. DIAGRAMAS DE SECUENCIA

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Mientras que el diagrama de casos de uso permite el modelado de una vista *business* del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

En nuestro ejemplo de ajedrez, podemos hacer diagramas de secuencia para "jugar partida" o bien para partes de "jugar partida", como puede ser "mover pieza".

Fig. 2.3 Diagrama de Secuencias



Fuente: [Schmuller - 2001]

Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se tiene modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

2.4.5. DIAGRAMAS DE ESTADOS

Un Diagrama de Estados muestra la secuencia de estados por los que pasa bien un caso de uso, bien un objeto a lo largo de su vida, o bien todo el sistema. En él se indican qué eventos hacen que se pase de un estado a otro y cuáles 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.

2.4.6. DIAGRAMAS DE COMPONENTES

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los

diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

Debido a que estos son más parecidos a los diagramas de casos de usos estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

En él se situarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema.

Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

2.4.7. DIAGRAMAS DE DESPLIEGUE

Se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones.

Un artefacto puede ser algo como un archivo, un programa, una biblioteca, o una base de datos construida o modificada en un proyecto. Estos artefactos implementan colecciones de componentes. Los nodos internos indican ambientes, un concepto más amplio que el hardware propiamente dicho, ya que un ambiente puede incluir al lenguaje de programación, a un sistema operativo, un ordenador o un cluster de terminales.

La mayoría de las veces el modelado de la vista de despliegue implica modelar la topología del hardware sobre el que se ejecuta el sistema. Aunque UML no es un lenguaje de especificación hardware de propósito general, se ha diseñado para modelar muchos de los aspectos hardware de un sistema a un nivel suficiente para que un ingeniero software pueda especificar la plataforma sobre la que se ejecuta el software del sistema.

2.5. PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP)

Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Mantiene al equipo enfocado en producir incrementalmente software operativo a tiempo, con las características requeridas y con la calidad requerida. Las mejores prácticas probadas en la industria, contenidas en el RUP, incorporan las lecciones aprendidas de cientos de líderes de la industria y miles de proyectos. Ya no hay necesidad de re-inventar soluciones a desafíos de la ingeniería de software bien conocidos. Siguiendo el acercamiento al desarrollo iterativo del RUP, es posible entregar a tiempo y con confianza el software.

2.5.1. CARACTERÍSTICAS

Una de las mejores prácticas centrales de RUP es la noción de desarrollar iterativamente. Rational Unified Process organiza los proyectos en términos de disciplinas y fases, consistiendo cada una en una o más iteraciones. Con esta aproximación iterativa, el énfasis de cada workflow variará a través del ciclo de vida. La aproximación iterativa ayuda a mitigar los riesgos en forma temprana y continua, con un progreso demostrable y frecuentes releases ejecutables.

- Las mejores prácticas más probadas de la industria - Son las mejores prácticas de desarrollo adoptadas en cientos de proyectos mundialmente y enseñadas como parte de la currícula en cientos de universidades, la metodología RUP se convirtió rápidamente en el estándar de facto para el proceso de desarrollo en la industria de software.
- Proceso hecho práctico - Diferente que otras metodologías comerciales, la plataforma RUP hace que el proceso sea práctico con bases de conocimiento y guías para ayudar en el despegue de la planificación del proyecto, integrar rápidamente a los miembros del equipo y poner en acción el proceso personalizado.
- Se adapta a las necesidades de los proyectos - Solo la plataforma RUP proporciona un framework de proceso configurable que permite seleccionar e implantar los componentes

específicos de proceso necesarios para proporcionar un proceso consistente y customizado para cada equipo y proyecto.

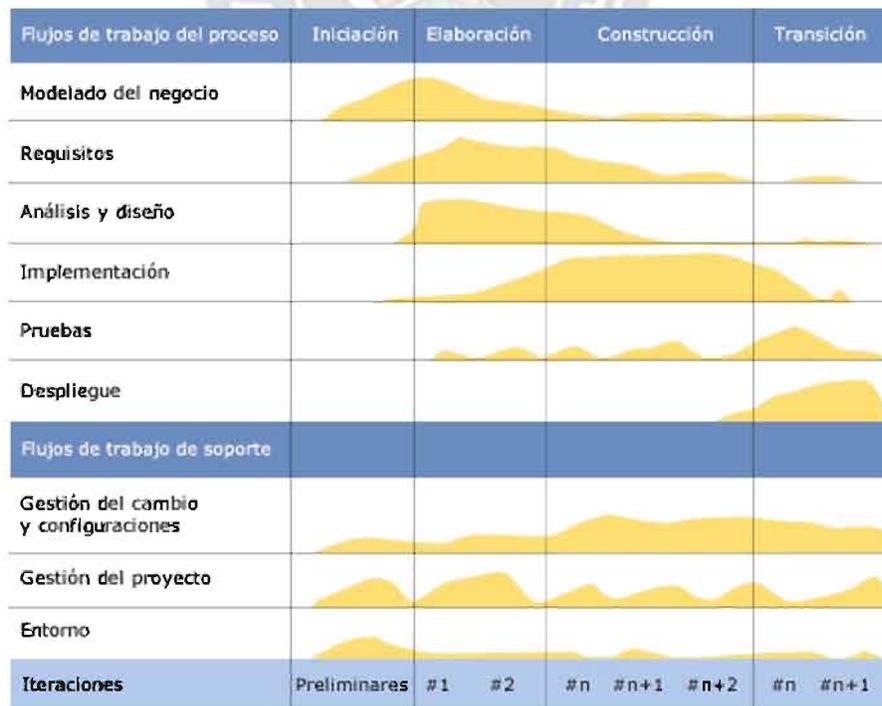
2.5.2. CICLO DE VIDA DE UN PROCESO UNIFICADO

El ciclo de vida RUP es una implementación del Desarrollo en espiral. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de Línea Base de la arquitectura.

Fig. 2.4 El proceso del modelado unificado.



Fuente: [WIKI01]

2.5.3. FASE DE INICIO

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.

2.5.4. FASE DE ELABORACIÓN

En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

2.5.5. FASE DE CONSTRUCCIÓN

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

2.5.6. FASE DE TRANSICIÓN

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

2.6. SISTEMAS DE INFORMACIÓN

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

El equipo computacional: el hardware necesario para que el sistema de información pueda operar.

El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información. Es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo).

2.6.1. BASE DE DATOS

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Una base de datos es un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

2.6.2. ESTRUCTURA DE UNA BASE DE DATOS

La estructura de una base de datos hace referencia a los tipos de datos, los vínculos o relaciones y las restricciones que deben cumplir esos datos (integridad de datos y redundancia de datos).

La estructura de una base de datos es diseñada o descrita empleando algún tipo de modelo de datos.

2.6.3. VENTAJAS DE LA BASE DE DATOS

- Control sobre la redundancia de datos, no se almacenan varias copias de los mismos datos.
- Consistencia de datos, si un dato está almacenado una sola vez, cualquier actualización se debe realizar solo una vez.
- Compartición de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.
- Mejora en la integridad de datos, con restricciones o reglas que no se pueden violar.

Y existen muchas otras ventajas en las bases de datos.

2.7. PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.

Con la POO tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas en términos de objetos, propiedades, métodos y otras cosas que veremos rápidamente para aclarar conceptos y dar una pequeña base que permita soltarnos un poco con este tipo de programación.

La programación orientada a objetos es una forma de programar que trata de encontrar una solución a estos problemas. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

- ✓ Clase: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.
- ✓ Herencia: (por ejemplo, herencia de la clase C a la clase D) Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (private) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Esto es así para mantener hegemónico el ideal de OOP.
- ✓ Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.

- ✓ Método: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

2.7.1. MODELO ORIENTADO A OBJETOS

Los métodos de diseño orientado a objetos han surgido para ayudar a los desarrolladores a explorar la potencia explosiva de los lenguajes de programación basados en objetos y orientados a objetos, utilizando las clases y los objetos como bloques básicos de construcción.

En realidad, el modelo orientado a objetos ha recibido la influencia de una serie de factores no solo de la programación orientada a objetos. Por el contrario, el modelo de objetos ha demostrado ser un concepto unificador en la informática, aplicable no solo a los lenguajes de programación, sino también al diseño de interfaces de usuario, Bases de Datos e incluso arquitecturas de computadoras.

El diseño orientado a objetos representa así un desarrollo evolutivo, no revolucionario; no rompe con los avances del pasado, sino que se basa en avances ya probados.

2.7.2. ANÁLISIS ORIENTADO A OBJETOS

El modelo de objetos ha influido incluso en las fases iniciales del ciclo de vida del desarrollo del software. La técnica de análisis estructurado tradicional cuyo mejor ejemplo es el trabajo de Yourdon.

El análisis orientado a objetos enfatiza la construcción de modelos del mundo real, utilizando una visión del mundo orientado a objetos: El análisis orientado a objetos es un método de análisis que examina los requisitos desde las perspectivas de clases y objetos que se encuentran en el vocabulario del dominio del problema. [BOOCH96].

Los documentos básicos de análisis orientado a objetos son:

- Documentos de análisis.

- Especificación de requisitos o requerimientos.
- Diagramas de casos de uso.
- Escenarios y subescenarios.
- Prototipos y su evaluación.

2.7.3. DISEÑO ORIENTADO A OBJETOS

Crea una representación del campo del problema del mundo real y la hace corresponder con el ámbito de la solución que es el software. El Diseño orientado a objetos produce un diseño que interconecta objetos de datos (elementos de datos) y operaciones de procesamiento en una forma que modulariza la información y el procesamiento, en lugar de dejar aparte el procesamiento.

El diseño orientado a objetos es una metodología de programación que define programas en términos de "clases de objetos", entidades que combinan, datos, procedimientos y propiedades. El diseño expresa un programa como un conjunto de estos objetos para realizar tareas.

De esta forma un objeto contiene toda la información que permite definirlo e identificarlo, dispone de mecanismos de interacción que favorecen la comunicación entre objetos y el cambio de datos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan la información y procesamiento.

El diseño orientado a objetos es totalmente distinto a los lenguajes de programación tradicionales, anima al programador a pensar sobre todo en términos de procedimientos o funciones.

Otros elementos importantes dentro de las nociones del análisis orientado a objetos son los conceptos de modulo que es explicado por Pressman como: la propiedad que tiene un sistema que ha sido descompuesto en módulos cohesivos y débilmente acoplados [PRESS98]. La modularidad crea una serie de fronteras bien definidas y documentadas dentro de un programa estas fronteras o interfaces, tienen un incalculable valor para la comprensión del programa. Y los procesos que encaminan a la construcción ordenada de los modelos del sistema.

El proceso comienza con la comprensión de la manera en la que se usará el sistema; por las personas, si el sistema es de iteración con el hombre; por otras maquinas. Una vez que se ha definido el escenario, comienza el modelado del software.

2.8. MODELO VISTA CONTROLADOR (MVC)

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

El sistema de arquitectura MVC está destinado a separar una aplicación o una parte de la aplicación en 3 partes:

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la *vista* y su *controlador* facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. 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.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista. El regulador traduce interacciones con la visión a las acciones que se realizarán 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.

2.8.1. MODELO DE TRES CAPAS

La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

2. Capa de presentación: es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.
3. Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
4. Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

Si, por el contrario, fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más ordenadores que realizarían

solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de negocio, y otra serie de ordenadores sobre los cuales corre la base de datos.

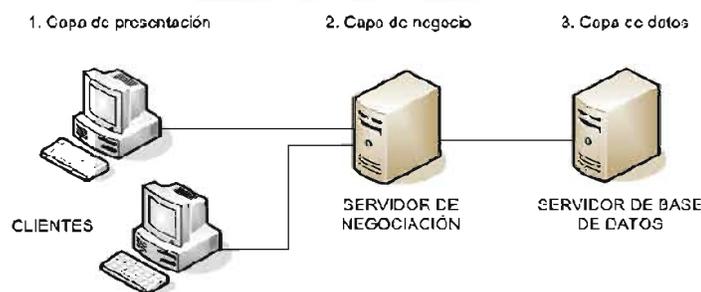
En una arquitectura de tres niveles, los términos "capas" y "niveles" no significan lo mismo ni son similares.

El término "capa" hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación, Lógica de Negocio y Datos.

En cambio, el término "nivel" corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo:

- Una solución de tres capas (presentación, lógica del negocio, datos) que residen en un solo ordenador (Presentación+lógica+datos). Se dice que la arquitectura de la solución es de tres capas y *un nivel*.
- Una solución de tres capas (presentación, lógica del negocio, datos) que residen en dos ordenadores (presentación+lógica, lógica+datos). Se dice que la arquitectura de la solución es de tres capas y *dos niveles*.

Fig. 2.5 Arquitectura de 3 capas



FUENTE: [CO, 08]

2.8.2. VENTAJAS DEL MVC

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que

revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos.

Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API (ej.: Microsoft Framework .NET) que existe entre niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

El diseño más utilizado actualmente es el diseño en tres niveles (o en tres capas).

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.

2.9. HERRAMIENTAS DE PROGRAMACIÓN

Las herramientas de programación, son aquellas que permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que la parte física del computador u ordenador, funcione y pueda producir resultados.

2.9.1. LENGUAJE DE DESARROLLO EN .NET

Aunque ASP.NET no es un lenguaje de programación en sí mismo, sino una arquitectura de desarrollo web en la que se pueden usar por debajo distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET o VBScript/JScript para ASP).

Se utilizan para servir los datos adecuados a las necesidades del usuario, en función de cómo hayan sido definidos por el dueño de la aplicación. Los datos se almacenan en alguna base de datos estándar.

ASP.NET es un Framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Existen muchas ventajas del porqué usar .NET para aplicaciones web:

1. Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
2. No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas.
3. No ocupan espacio en nuestro disco duro.
4. Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
5. Consumo de recursos bajo: Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
6. Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
7. Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
8. La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
9. Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
10. Colaboración: Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.

2.9.2. GESTOR DE BASE DE DATOS SQL SERVER

Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

(Sistema de gestión de base de datos) o en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

En la manipulación de una base de datos, los SGBD deben incluir un control de concurrencia, o sea, deben permitir a varios usuarios tener acceso "simultáneo" a la base de datos. Controlar la concurrencia implica que si varios usuarios acceden a la base de datos, la actualización de los datos se haga de forma controlada para que no haya problemas.

Un SGBD también debe encargarse de cumplir las reglas de integridad y redundancias.

Otra función importante en un SGBD es su capacidad de realizar copias de seguridad y de recuperación de datos.

Suministra múltiples interfaces de usuario, restringe accesos no autorizados y representa relaciones complejas entre los datos.

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés *structured query language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre ella. Es un lenguaje de cuarta generación.

2.9.3. SERVIDOR IIS (INTERNET INFORMATION SERVER)

Un servidor web es un programa que se está ejecutando en un equipo, normalmente un servidor y que proporciona páginas web a los "clientes" que le piden. Los clientes son los navegadores web como Internet Explorer o NetScape

Los servicios de Internet Information Server (o IIS), son los servicios de software que admiten la creación, configuración y administración de sitios Web, además de otras funciones de

Internet. Los servicios de Microsoft Internet Information Server incluyen el Protocolo de transferencia de noticias a través de la red (NNTP), el Protocolo de transferencia de archivos (FTP) y el Protocolo simple de transferencia de correo (SMTP).

Los servicios de Internet Information Server facilitan la publicación de información en una intranet o en Internet; IIS permite autenticación robusta y segura de los usuarios, así como comunicaciones seguras vía SSL; además podemos crear contenido dinámico utilizando los componentes y secuencias de comandos del servidor para crear contenido dinámico independiente del explorador mediante páginas Active Server (ASP).

Por tanto si en un navegador o explorador web escribimos una dirección o una página intentará localizar el servidor escrito y buscar la página solicitada. Para que ocurra esto debemos tener instalado en nuestro servidor el programa Internet Information Server. Internet Information Server, que llamaremos IIS, es el servidor de páginas web de Microsoft. Éste viene de forma gratuita con el sistema operativo Windows NT, 2000 y XP y descargable para los demás.

2.10. SEGURIDAD

La seguridad del software es una actividad de garantía de calidad del software que se centra en la identificación y evaluación de los riesgos potenciales que pueden producir un impacto negativo en el software y hacer que falle el sistema completo. Si se pueden identificar pronto los riesgos en el proceso de ingeniería del software podrán especificarse las características del diseño del software que permitan eliminar o controlar los riesgos potenciales.

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 usuario 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.

2.11. MÉTRICAS DE CALIDAD

El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software. Este estándar permitirá medir el grado de calidad del sistema a construirse y se empleará algunos puntos clave de calidad como ser:

- a) 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 el programa reaccione en forma inesperada y el usuario se encuentre cómodo usándolo ya que estos fallos no traen graves problemas.

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):

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

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:

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

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

Donde:

λ : Es la tasa constante de fallo

t : Período de operación en tiempo.

Luego de obtener 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, se considera dos situaciones: primero se considera que el sistema entero falla y si falla cualquiera de sus componentes, segundo 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: 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]:

Números de: entradas de usuario, salidas de usuario, peticiones de usuario, archivos y de interfaces externas.

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

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

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.

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

Una vez detalladas cada una de las características, ahora estas cinco las colocamos en una tabla de evaluación.

- c) Mantenimiento de Software: 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 4 tipos de mantenimiento [SOM 95], y son:

- 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 más efectivos o simplemente que amplíe las funciones del mismo.
- Mantenimiento Adaptivo: este tipo de mantenimiento es frecuentemente más 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.

d) Portabilidad: La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

Es la capacidad de llevar exitosamente el sistema a otro entorno. El proceso de portabilidad tiene dos componentes que son:

- 1) Transportación: se refiere a la fase física del programa, esto 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.

CAPITULO III

MARCO APLICATIVO

En este capítulo aplicaré los fundamentos teóricos y técnicos planteados en el capítulo anterior, llevándolos a la práctica para el logro del desarrollo del Sistema de información de administración y control para el Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz". Con la aplicación de la metodología RUP en el análisis y desarrollo del sistema.

3.1 FASE DE INICIO

La fase del inicio de la creación de un proyecto es un tanto complicado, pero se tiene que persistir, por ello se tiene en cuenta que cualquier problema que ya sea grande o pequeño siempre tendrá una solución. Con tal motivación se inicia el presente proyecto realizando las actividades correspondientes a esta fase.

En esta fase se realizará una planificación de las actividades que se realizaran dentro del marco del Proceso Unificado de Desarrollo, señalando los hilos principales, las funciones y todo lo que se involucra en cada actividad, considerando también los tiempos que fueron necesarios para su correspondiente realización.

3.1.1 FUNCIONES PRINCIPALES DEL SINDICATO DE GRUAS, MONTACARGAS Y EQUIPO PESADO "LA PAZ"

Los procesos que se realizan en el Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz", es el de prestar servicios a empresas e industrias que manejan equipos pesados mediante el uso de maquinaria pesada que son todo lo relacionado con grúas, camiones, montacargas y demás vehículos; todos estos vehículos junto con los dueños son empresas que están afiliadas al sindicato, el sindicato es la unión de todas las empresas que otorgan servicios de maquinaria pesada y se encuentran afiliados.

Distintos clientes solicitan servicios a cada empresa afiliada al sindicato, para que transporten equipos pesados de alto tonelaje, estos servicios se realizan mediante una cotización que la empresa entrega anticipadamente a sus clientes o se realiza el trabajo por hora.

El Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz", es el encargado de administrar y controlar a todo el personal afiliado, asimismo controlar la cantidad de vehículos de transporte pesado que se hallan afiliados, todo con el motivo de tener la unión y potestad de que no puedan ser allanados por otros tipos de sindicatos ni sectores sociales y así cada empresa afiliada pueda trabajar con mayor tranquilidad en su parada o puesto de trabajo. De esta manera según la especificación del entorno de su trabajo, se realizan las siguientes tareas:

- El directorio general, representa al sindicato en todos sus actos particulares o colectivos.
- Hacen cumplir los estatutos, reglamentos y demás imposiciones, así también las disposiciones emanadas de la federación, confederación y congresos.
- Reuniones que se van generando de acuerdo a los problemas que comienzan a surgir.
- Administrar cuotas de ingreso de nuevos socios, vehículos, otras cuotas mensuales, diarias, contribuciones económicas especiales, dirigido por el secretario de hacienda.
- Generan registros manuales, listados de todos los recursos que son las empresas y sus correspondientes vehículos.
- Decisiones sobre la aceptación de que ingresen nuevas empresas al sindicato.
- Operadores, dueños de cada empresa y vehículos que están afiliados al sindicato, son registrados en los libros de actas.
- A modo de distracción, también realizan partidos que suelen jugarse los fines de semana, organizado por el secretario de deportes.
- Conocer todos los casos donde existan conflictos entre socios, o de socios con respecto a cuestiones particulares, para lograr dar soluciones a los mismos, mediante la administración del secretario de conflictos.

3.1.2 RECOPIACIÓN DE REQUISITOS

Los requerimientos identificados dentro del Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz" en el marco de desarrollo del sistema a implementarse, se detallan a continuación:

1. Registrar a los afiliados al sindicato que son los recursos humanos (secretarios de administración del sindicato y dueños de cada empresa) y a los vehículos.
2. Mostrar reportes según los requerimientos encontrados en la entidad.
3. El diseño de una interfaz web del sistema.

4. Construir una base de datos que otorgue integridad al sindicato.
5. Mostrar reportes de los cobros a clientes que se deben realizar.
6. Generar un subsistema de control y administración de kardex de personal de socios.
7. Crear una interfaz que permita actualizar los datos vehiculares y socios en el módulo correspondiente a infracciones y accidentes para su posterior sanción.

3.1.3 ANALISIS

La recopilación de requisitos fue realizada considerando ciertas necesidades que exige el sindicato.

El análisis de los problemas del sistema manual que se emplea actualmente son:

1. Incapacidad de obtener rápidamente los datos de los afiliados cuando se los necesita.
2. Se tarda en obtener la lista de actas que debe realizar el sindicato para su correspondiente lectura.
3. Existe un control ineficiente acerca de los recursos humanos y vehiculares.
4. Duplicación de registros en archivos físicos.

3.1.4 IMPLEMENTACION

En esta actividad del flujo de trabajo de implementación se realizará a partir de los resultados del diseño, de acuerdo a las versiones realizadas se permitió ir formando la versión del sistema. En esta fase el software será preparado para la transición, el cual se instalará en la secretaria del sindicato.

3.1.5 PRUEBAS

Las pruebas se realizarán acabando el prototipo del sistema, concordando con el secretario general del sindicato.

3.1.6 ANALISIS Y GESTION DE RIESGOS

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

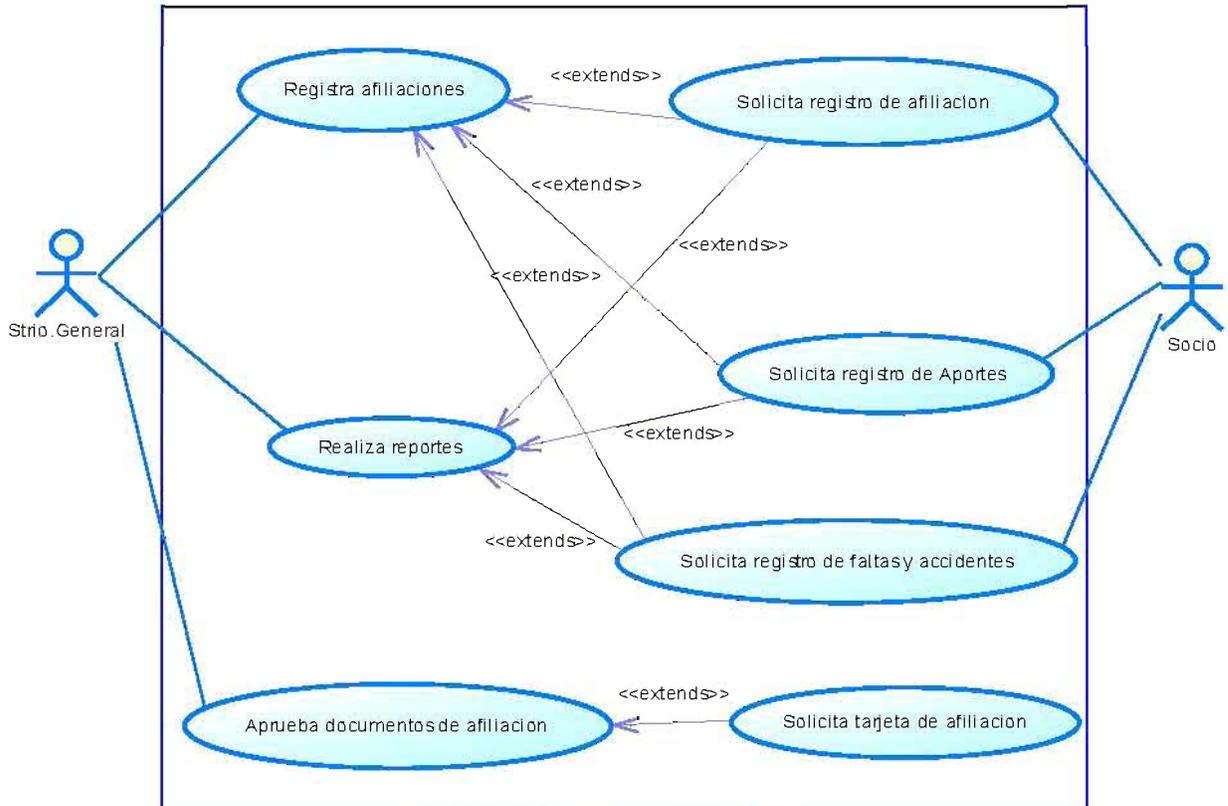
El riesgo implica las siguientes características:

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

3.1.7 MODELO DEL NEGOCIO Y ANALISIS DE LA SITUACION ACTUAL

En la siguiente figura se identifica y se delimita los "procesos del negocio" según los objetivos y ejecuciones de actividades del Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz" que comienza a partir de un análisis de la situación actual. Las personas identificadas en el modelo del negocio se utilizan como punto de partida para exponer a un conjunto de actores y sus casos de uso correspondientes.

Fig. 3.1 Diagrama de la Situación Actual



Fuente. Propia Elaboración

3.1.7.1 SITUACION ACTUAL

CURSO NORMAL DE EVENTOS

Caso de uso: Registrar afiliaciones

Actor: Secretario General

Propósito: Realizar un registro y control de socios en el sindicato.

Resumen: El secretario general realiza un control de afiliación permanente, solicitando a los secretarios de cada área un informe de registros para así tomar decisiones.

Acción del Actor	Respuesta del Sistema
1. El Strio. Gral. solicita un informe de todos los registros de socios al Strio. del área correspondiente.	2. El strio. del área correspondiente busca y verifica en los archivos la información requerida. 3. Revisa la información para enviar el informe solicitado de registro de socios,

	aportes, faltas o accidentes. 4. El strio. del área emite el informe al Strio. Gral.
5. El Strio. Gral. revisa el informe y realiza una copia para el control administrativo. 6. El Strio. Gral. Evalúa el informe del seguimiento y control de la afiliación.	

Caso de uso: Solicita registro de afiliación

Actor: Socio

Propósito: solicitar un registro de afiliación o actualización de datos de socios.

Resumen: el socio solicita un registro de afiliación al Strio. Gral., previa aprobación para su afiliación verificando documentos.

Acción del Actor	Respuesta del Sistema
1. El socio dueño de cada empresa de maquinaria pesada solicita un registro de afiliación al Strio. Gral.	2. Strio. Gral. solicita al socio dueño los documentos de requisitos para su afiliación.
3. El socio entrega la documentación aprobada.	4. Strio. Gral. verifica la documentación y registra los datos personales del socio y los datos vehiculares de operación. 5. Strio. Gral. elabora un contrato de afiliación según el tipo de socio y el grupo al que pertenece. 6. Strio. Gral. aprueba la afiliación.
7. El socio acepta las normas y reglas de la institución y el contrato de afiliación.	8. Strio. Gral. entrega la copia del contrato de afiliación al nuevo socio.
9. El socio recibe la copia del contrato de afiliación y la designación del grupo de trabajo en el sindicato.	

Caso de uso: Solicita registro de aportes

Actor: socio

Propósito: solicitar un registro y actualización de aportes de socios.

Resumen: el socio solicita al strio. de hacienda el registro semanal o mensual de los aportes realizados actualizando así su ficha de aporte, además de otros tipos de aporte.

Acción del Actor	Respuesta del Sistema
1. El socio solicita al Strio. de hacienda el registro o actualización de aportes realizados	2. Strio. de hacienda busca su kardex personal del socio, para actualizarlo en el punto de aportes
3. El socio entrega su tarjeta de aportes y efectúa el importe indicado según el tipo de aporte realizado.	4. El Strio. de hacienda recibe el importe del socio, actualiza su tarjeta de aporte con fecha y monto. 5. Strio. de hacienda elabora un recibo de

	aporte de comprobante de pagos, entrega una copia al socio y actualiza su kardex.
6. El socio recibe su tarjeta de aportes actualizados y su recibo de aportes.	7. Strio. de hacienda notifica al socio el próximo aporte con fecha.

Caso de uso: solicita registro de faltas y accidentes

Actor: socio

Propósito: solicita el registro de faltas y accidentes verificando sus aportes realizados y actualizado su kardex personal.

Resumen: el Strio. de conflictos realiza un registro de faltas o accidentes cometidos por el socio, actualizando su kardex personal y emitiendo su correspondiente sanción al socio infractor.

Acción del Actor	Respuesta del Sistema
1. Comienza cuando el socio afiliado comete un accidente o falta considerable durante el servicio a la institución.	2. Strio. de conflictos solicita al socio en caso de accidente un informe médico y vehicular o el tipo de infracción.
3. El socio entrega al Strio. de conflictos el informe correspondiente al caso.	4. Strio. de conflictos verifica los aportes realizados por el socio, para cubrir el seguro que comprende al socio y vehículo en caso de accidentes. 5. Strio. de conflictos emite una sanción ya sea económica o según las normas internas en caso de faltas o infracciones.
6. El socio acepta la notificación de la sanción correspondiente en caso de infracción cometida.	7. Strio. de conflictos entrega el informe al Strio. Gral. para su aprobación y realiza un recibo de aporte de seguro para su desembolso en el area de hacienda del sindicato.
8. Socio recibe el aporte de seguro y acepta el plan de pagos.	

Caso de uso: Realiza Reportes

Actor: Secretario General

Propósito: Realiza reportes necesarios en Excel solicitados por el RUA o por otras instituciones.

Resumen: El Strio. Gral. realiza reportes solicitados por el RUA o por otras instituciones para verificar que los socios afiliados cuentan con el soat respectivo de operación y los impuestos pagados al día.

Acción del Actor	Respuesta del Sistema
1. El Strio. Gral. solicita al Strio. del área correspondiente los registros de socios y vehículos afiliados al sindicato para su revisión y aprobación antes de enviar los registros al organismo pertinente.	2. El Strio. del area correspondiente revisa y actualiza los datos de los socios afiliados. 3. Envía la información pedida ya sea de registro de socios y vehículos, aportes, faltas y accidentes de los afiliados.
4. El Strio. Gral. revisa los reportes para su aprobación y lo envía al organismo que lo requiere.	

Caso de uso: Emite y aprueba documentos de afiliación

Actor: Secretario general

Propósito: el Strio. Gral. realiza documentos de afiliación y kardex personal verificando los requisitos establecidos en las normas internas del sindicato.

Resumen: El Strio. Gral. genera documentos de afiliación y kardex personal verificando los requisitos establecidos en las normas internas del sindicato.

Acción del Actor	Respuesta del Sistema
1. El Strio. Gral. solicita al strio. correspondiente la solicitud de afiliación del socio y requisitos para su aprobación.	2. El Strio. correspondiente entrega la solicitud y el contrato de afiliación elaborado.
3. El Strio. Gral. revisa y aprueba la solicitud de afiliación.	4. El Strio. correspondiente entrega al socio el memorándum de aprobación y afiliación al sindicato.

Caso de uso: Solicita tarjeta de afiliación

Actor: Socio

Propósito: solicita la respectiva tarjeta de afiliación.

Resumen: el socio solicita al Strio. Gral. la tarjeta de afiliación.

Acción del Actor	Respuesta del Sistema
1. El socio solicita la tarjeta de afiliación para operar en su área correspondiente.	2. El Strio. Gral. revisa sus aportes por el strio. de hacienda para la entrega de su tarjeta.
4. El socio recibe la nueva tarjeta de afiliación para el mes establecido.	3. Si los aportes están al día, el Strio. Gral. entrega al socio sus tarjetas, caso contrario no le entrega hasta su cancelación total de deudas pendientes.

3.2 FASE DE ELABORACION

Ingresando a la fase de elaboración, se analiza el dominio del problema de manera que facilite su comprensión y que nos proporcione una visión general del sistema como una primera aproximación al modelo del diseño; y para este cometido se identificaron los requerimientos obtenidos en la primera fase y luego eliminar los riesgos para esta segunda fase.

3.2.1 DOCUMENTO DE ANALISIS DEL SISTEMA DE INFORMACION

Se tiene que diseñar un sistema capaz de mantener una base de datos con la siguiente información: socios (referente a los propietarios activos y asalariados), los vehículos (que son todo lo que se refiere a las maquinarias pesadas como ser: grúas, camiones grúas, montacargas, loboy y camiones), el tipo de aporte que realizan, los accidentes que comenten los socios, infracciones y su sanción.

A continuación se muestra los elementos que están implicados en el sistema y en la base de datos:

Vehículos	Kardex personal
Socios	Aportes
Administrativos	Infracciones
Sanciones	Accidentes
Tarjetas de afiliación	Licencia de operación

Es importante tener un estudio detallado del comportamiento de la base de datos de las entidades encontradas junto a sus relaciones correspondientes para así lograr controlar, actualizar y registrar datos en el sistema mediante inserciones, modificaciones y eliminaciones. Es imprescindible tener datos del socio afiliado: código de afiliación, cedula de identidad, nit (número de identificación tributaria) si lo tiene, nombres, apellidos, dirección, teléfonos, nombre de empresa o si es individua, cargo que ocupa, fax, email y sitio web.

Para los administrativos se debe conocer: cedula de identidad, apellidos, nombres, dirección, cargo que ocupa, etc.

Para los vehículos se debe conocer: número de placa, clase, marca, capacidad, modelo, color, número de soat, vehículo a nombre de propiedad, número de motor, chasis, tipo de transporte y descripción.

Para el aporte conocer: fecha del aporte, código, cédula de identidad del socio, importe realizado, número de recibo y total del importe realizado hasta esa fecha, tipo de aporte

De los accidentes es necesario saber: código y descripción del accidente correspondiente.

De las infracciones saber: código y descripción de la infracción. Para el socio que comete infracción saber: código de infracción, cédula de identidad del socio, fecha de infracción y motivo correspondiente.

Para la sanción correspondiente es necesario saber: código de sanción, código de infracción, descripción y monto a descontar.

3.2.2 IDENTIFICACION DE ACTORES PARA EL NUEVO SISTEMA

Los actores que se encuentran involucrados en el desarrollo de los procesos y acciones que realiza cada uno de ellos son los siguientes:

Nombre de actor: Administrador del sistema

Definición: Es el encargado principal de administrar todo el sistema. Tendrá todos los permisos y libertad de movimiento en el sistema.

Notas: El administrador es el encargado de manipular la información contenida en el sistema. Tiene acceso a toda la información que se especificó. Es la única persona que puede realizar cambios de claves de acceso a otro usuario al sistema, previa autorización del strio. general del sindicato.

Nombre de actor: Secretario General

Definición: Es la persona que aprueba la solicitud de afiliación de socios, previa solicitud de requerimientos aceptados por el sindicato, aprueba solicitudes internas, tiene acceso a toda la información del sistema.

Notas: Es el responsable de toda la información contenida en el sistema.

Nombre de actor: Secretario de actas

Definición: persona autorizada para acceder al modulo de registros de personal de socios y vehículos, realizar consultas en el sistema.

Notas: Es la persona indicada para acceder al modulo de registros de personal que son los socios y vehículos, realizar consultas.

Nombre de actor: Secretario de hacienda

Definición: Encargado de administrar la información contable del sindicato, elaborará planillas de aportes por grupos en el sindicato; genera datos estadísticos contables y descuentos por faltas y accidentes.

Notas: registra los aportes realizados semanalmente o mensualmente por los socios en la base de datos, obteniendo las tarjetas de aportes asignadas a cada socio.

Nombre de actor: Secretario de conflictos

Definición: Encargado de elaborar informes de las faltas y accidentes cometidos por los socios afiliados, actualizando la base de datos, el kardex personal para su posterior sanción.

Notas: El secretario de conflictos trabajará junto con el secretario de hacienda, porque cualquier sanción o accidente cometido por un socio se verá obligado a afectar su cuenta de aportes por el secretario de hacienda.

3.2.3 MODELO DE CASOS DE USO DEL NUEVO SISTEMA

Para esta nueva etapa se identificaron los requerimientos, los casos de uso que sostendrá el nuevo sistema, diseñando los nuevos modelos de casos de uso del sistema con sus respectivas tareas o funciones.

3.2.3.1 ESPECIFICACION DE REQUERIMIENTOS

Como resultado de la recopilación de requerimientos de las necesidades de un producto. Se obtuvo un documento de lo que en realidad necesita el sindicato, en una forma que claramente se lo comunique a los miembros y socios.

3.2.3.2 FUNCIONES DEL SISTEMA

Las funciones del sistema son lo que éste habrá de hacer. Es muy importante identificarlas y listarlas en grupos de manera que tengan lógica.

Tabla 3.1: Funciones del nuevo sistema

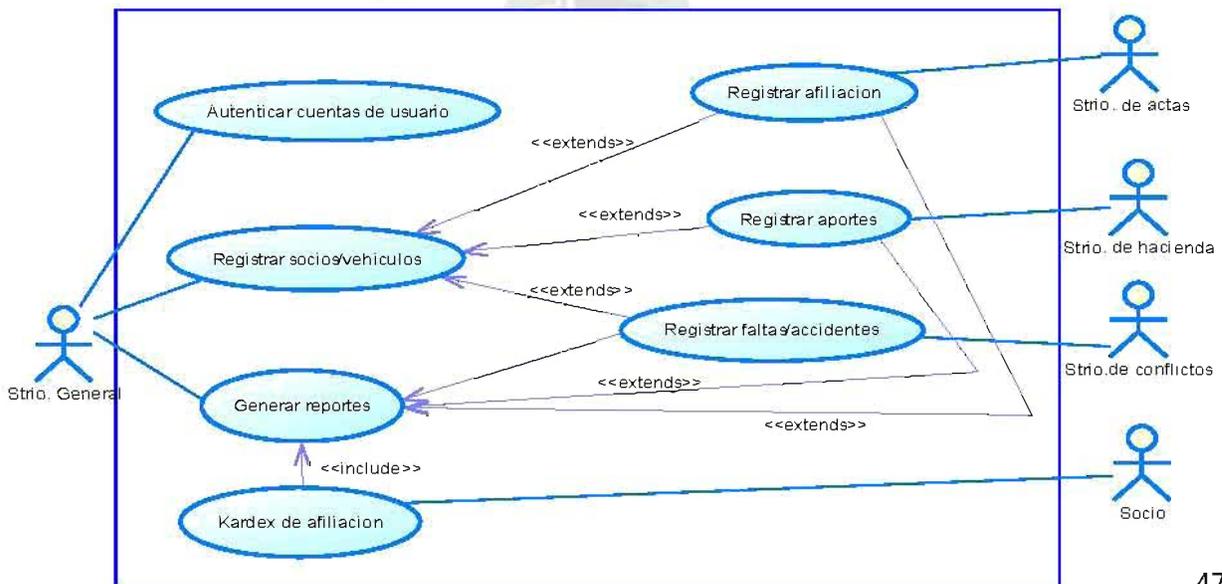
Nº	Funciones
1	El sistema deberá registrar socios, administrativos, diferentes acciones y vehículos.
2	El sistema deberá notificar las infracciones correspondientes a cada socio.
3	El sistema deberá mostrar reportes de varias acciones como ser: aportes, sanciones, infracciones, afiliaciones, etc.
4	El sistema deberá modificar a socios, administrativos, acciones y vehículos.

Fuente: Elaboración Propia

3.3 DIAGRAMA DE CASOS DE USO

La recopilación de requisitos fue realizada considerando la línea base de la arquitectura del sistema y a la priorización de los riesgos percibidos, a consecuencia de ello, se realiza el diagrama de casos de uso de primer nivel, el cual se muestra a continuación en la siguiente figura:

Fig. 3.2 Diagrama de Casos de Uso del nuevo Sistema



Fuente: Elaboración Propia

3.3.1 FUNCIONES BASICAS DEL SISTEMA

Luego de una exhaustiva investigación de las necesidades que tienen los administradores del sindicato y de recopilar los requisitos realizando un análisis con los administrativos, se obtuvo el siguiente documento de análisis del sistema propuesto, el que contiene la información requerida por el cliente y los aportes correspondientes realizados.

Las siguientes funciones del sistema en la aplicación del control y administración del sindicato con una muestra representativa; no pretenden en lo absoluto ser exhaustivas. El objetivo es entender los detalles del análisis y del diseño, no el funcionamiento detallado de una administración y control.

Existen categorías que se clasifican a fin de establecer prioridades entre ellas e identificar las que de lo contrario pasarían inadvertidos (pero que consumen tiempo y otros recursos), estas son:

- Evidente, debe realizarse y el usuario debería saber que se ha realizado.
- Oculta, debe realizarse, aunque no es visible para los usuarios. Esto aplica a muchos servicios técnicos subyacentes, como *guardar información en un mecanismo persistente de almacenamiento*.
- Superflua, son opcionales, su inclusión no repercute significativamente en el costo ni en otras funciones.

Tabla. 3.2: Atributos del sistema en las especificaciones de funciones

Ref	Función	Categoría	Atributo	Detalles y Restricciones	Categoría
R.1	El sistema deberá controlar y verificar que el tipo de usuario introduzca su código y contraseña de acceso para ingresar al módulo correspondiente.	Evidente	Acceso restringido	Acceso por medio de código y contraseña única.	Obligatorio
R.2	El sistema deberá restringir el manejo de opciones del menú principal de acuerdo al tipo de usuario registrado en la base de datos	Oculto	Acceso restringido	Pantalla de interfaz adecuada a lo determinado al tipo de usuario.	Obligatorio
R.3	El sistema deberá modificar las cuentas de usuario para el acceso al sistema.	Evidente – oculto	Acceso restringido	Solo el Administrador podrá modificar cuentas de usuario.	Obligatorio
R.4	El sistema deberá mostrar ventanas independientes de operación de registros y consultas.	Evidente	Metáfora de interfaz	Pantallas basadas en módulos del sistema.	Obligatorio
R.5	El sistema deberá tener la opción de registrar nuevos datos de los administradores, vehículos, socios, secretarios, etc. que están involucrados en las afiliaciones y otros asuntos del sindicato.	Evidente	Metáfora de interfaz	Pantallas basadas en los registros ya establecidos.	Obligatorio
			Acceso restringido	Solo los administradores pueden modificar ciertos datos en el sistema	Obligatorio
R.6	El sistema deberá modificar datos cuando sea necesario de los administradores, vehículos, socios, etc.	Evidente	Metáfora de interfaz	Pantallas basadas en registros establecidos.	Obligatorio
			Acceso restringido	Solo los administradores pueden modificar ciertos datos en el sistema.	Obligatorio
R.7	El sistema deberá eliminar datos cuando sea necesario de los administradores, vehículos, socios, etc.	Evidente	Metáfora de interfaz	Pantallas basadas en registros establecidos.	Obligatorio
			Acceso restringido	Solo los administradores pueden modificar ciertos datos en el sistema.	Obligatorio
R.8	El sistema deberá mostrar listados de datos de módulos correspondientes.	Evidente	Metáfora de interfaz	Pantallas basadas en listados de módulos.	Obligatorio

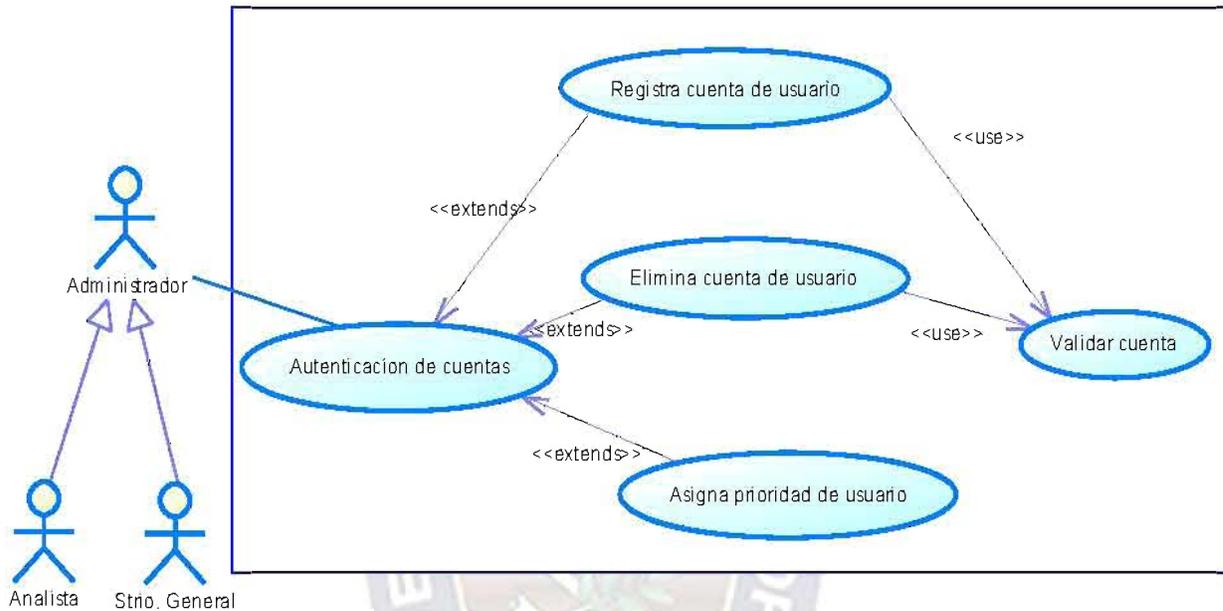
			Acceso libre	Todos los usuarios podrán visualizar los listados que genera el sistema.	Obligatorio
R.9	El sistema deberá generar reportes de datos de módulos correspondientes.	Evidente	Metáfora de interfaz	Pantallas basadas de acuerdo a la petición de los administradores	Obligatorio
			Acceso restringido	Solo los administrativos podrán generar reportes de listados en el sistema.	Obligatorio
R.10	El sistema deberá permitir cambios y modificaciones de datos de socios y vehículos.	Evidente	Metáfora de interfaz	Pantallas basadas en formularios de modificaciones.	Obligatorio
			Acceso restringido	Solo los administrativos podrán modificar la base de datos	Obligatorio
R.11	El sistema deberá generar datos estadísticos o consultas para su posterior impresión.	Oculto	Acceso restringido	Solo los administrativos podrán generar datos estadísticos.	Obligatorio
R.12	El sistema deberá registrar los aportes realizados por los socios actualizando su tarjeta de aporte correspondiente.	Evidente Oculto	Acceso restringido	Solo los administrativos o el strio. de actas podrá registrar los aportes.	obligatorio
R.13	El sistema deberá generar un listado de impresión de las tarjetas de aportes, kardex personal, etc.	Evidente Oculto	Acceso restringido	Solo los administrativos podrán imprimir los listados correspondientes.	Obligatorio
R.14	El sistema deberá contar con la debida seguridad respecto a los datos almacenados en la base de datos y los módulos del sistema.	Oculto	Acceso restringido	Solo el administrador podrá actualizar y verificar la seguridad del sistema.	Obligatorio
R.15	En el sistema se deberá realizar copias de seguridad de la base de datos por dispositivos de salida.	Oculto	Acceso restringido	Solo el Administrador podrá realizar copias de seguridad permanente.	Obligatorio
R.16	El sistema deberá contar con la debida seguridad respecto a los datos almacenados en la base de datos y los módulos del sistema.	Oculto	Acceso restringido	Solo el Administrador podrá actualizar, dar mantenimiento y verificar la seguridad del sistema.	Obligatorio

Fuente: Elaboración Propia

3.4 ANALISIS Y DISEÑO

3.4.1 DESCRIPCION DE LOS CASOS DE USO EXPANDIDOS

Fig. 3.3 Diagrama de casos de uso: Autenticación de cuentas



Fuente: Elaboración propia

CASO DE USO: Registra cuenta de usuario

Actor: Administrador

Propósito: Registrar o actualizar usuarios al sistema.

Resumen: El responsable administrativo podrá registrar a nuevos usuarios en el sistema y asignarle funciones como administrador, activando su cuenta y contraseña correspondiente con la opción de ser modificado por el usuario.

Tipo: Primario esencial

Curso normal de Eventos:

Acción del actor	Respuesta del Sistema
1. El responsable Administrativo ingresa a la opción de cuentas de usuario y adiciona un nuevo usuario.	2. Muestra el formulario para el registro de datos del nuevo usuario.

3. El responsable Administrativo introduce los datos personales del nuevo usuario y acepta la adición.	4. Registra y guarda los datos personales del nuevo usuario en la base de datos del sistema.
--	--

Diagrama de secuencia:

Fig. 3.4 Diagrama de secuencia: Registra cuenta de usuario



Fuente: Elaboración propia

Contrato de operaciones:

Nombre: Activa cuenta de usuario

Responsabilidad: Mostrar el fomulario para registrar nuevos usuarios.

Precondición: La solicitud del nuevo usuario estará predeterminada por el cargo que tiene en el sindicato.

Postcondición: Se muestra el formulario de adición de usuarios.

Nombre: Introduce datos

Responsabilidad: Registra los datos personales del usuario administrativo.

Precondición: Llenado del formulario correctamente.

Postcondición: Los datos introducidos en el formulario son aceptados.

Nombre: Guardar datos

Responsabilidad: Guarda los datos del usuario en la base de datos.

Precondición: Si el usuario ya existe en la base de datos, entonces no es agregado.

Postcondición: Los datos del usuario son registrados y guardados en la base de datos.

CASO DE USO: Elimina cuenta de usuario

Actor: Administrador

Propósito: Permitir la eliminación de usuarios del sistema.

Resumen: El responsable administrativo eliminará los datos personales y sus atributos del usuario de la base de datos del sistema.

Tipo: Primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El administrador Ingresa a la opción de cuentas de usuario y elimina usuario.	2. Muestra el fomulario para eliminar el usuario.
3. El administrador introduce la cédula de identidad del usuario para eliminar.	4. Verifica en la base de datos y selecciona los datos del usuario a eliminar.
5. Confirma la opción eliminar	6. Elimina los datos del usuario en la base de datos del sistema.

Diagrama de secuencia:

Fig. 3.5 Diagrama de secuencia: Elimina cuenta de usuario



Fuente: Elaboración propia

Contrato de operaciones:

Nombre: Selecciona opción eliminar

Responsabilidad: Muestra el formulario de eliminar usuarios.

Precondición: El formulario estará predeterminado por el cargo que tiene el usuario.

Postcondición: Se muestra el formulario de eliminación de usuarios.

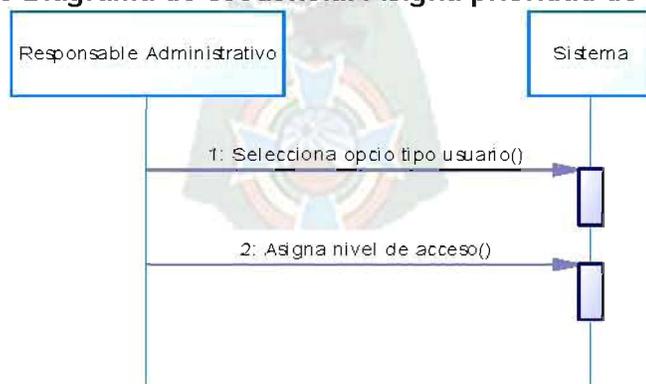
Nombre: Introducir login

- Responsabilidad:** Verificar los datos de usuario administrativo.
- Precondición:** Verificar que la cédula del usuario exista en la base de datos.
- Postcondición:** Selecciona los datos del usuario a ser eliminado
- Nombre:** Elimina los datos
- Responsabilidad:** Eliminar los datos del usuario de la base de datos.
- Precondición:** Los datos del usuario existen en la base de datos.
- Postcondición:** Los datos del usuario fueron eliminados de la base de datos del sistema.
- CASO DE USO:** **Asigna prioridad de usuario**
- Actor:** Responsable administrativo
- Propósito:** Asignar prioridad a las cuentas de usuario para el sistema.
- Resumen:** El responsable administrativo al momento de registrar al usuario, deberá asignarle el tipo de usuario que establecerá el acceso al nuevo usuario registrado dentro del sistema.
- Tipo:** Primario esencial
- Curso normal de eventos:**

Acción del actor	Respuesta del sistema
2. El responsable administrativo registra al nuevo usuario y selecciona la opción del tipo de usuario.	3. Muestra el formulario de adición junto con el tipo de acceso de usuario.
4. El responsable administrativo asigna el tipo de acceso del usuario.	5. Guarda la información del usuario en la base de datos del sistema.

Diagrama de secuencia:

Fig. 3.6 Diagrama de secuencia: Asigna prioridad de usuario



Fuente: Elaboración propia

Contrato de operaciones:

Nombre: Selecciona opción de tipo de usuario

Responsabilidad: Muestra formulario de adición conjuntamente el nivel, tipo de usuario.

Precondición: El formulario estará predeterminado por la jerarquía de usuario.

Postcondición: Se muestra el formulario del usuario.

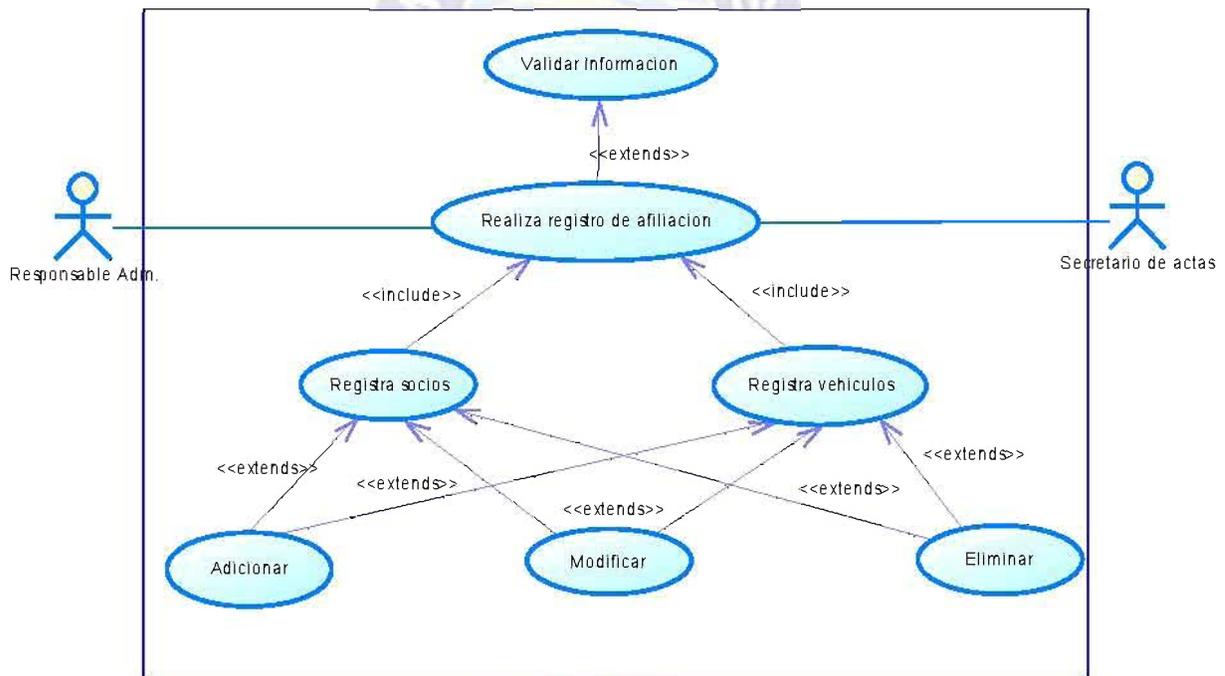
Nombre: Asigna el nivel de acceso.

Responsabilidad: Asigna el nivel de acceso y el tipo de usuario en el sistema.

Precondición: Los niveles de acceso y el tipo de usuario son definidos en el sistema.

Postcondición: Los datos fueron guardados en el sistema.

Fig. 3.7 Diagrama de casos de uso: Registrar cuenta de afiliación



Fuente: Elaboración Propia

CASO DE USO: Registra socios

Actor: Responsable administrativo, secretario de actas.

Propósito: Ingresar al módulo de registros de grúas y maquinaria pesada de socios.

Resumen: El responsable administrativo o el strio. de actas ingresará los datos al módulo correspondiente, previa autenticación, para su adición, modificación o eliminación de socios.

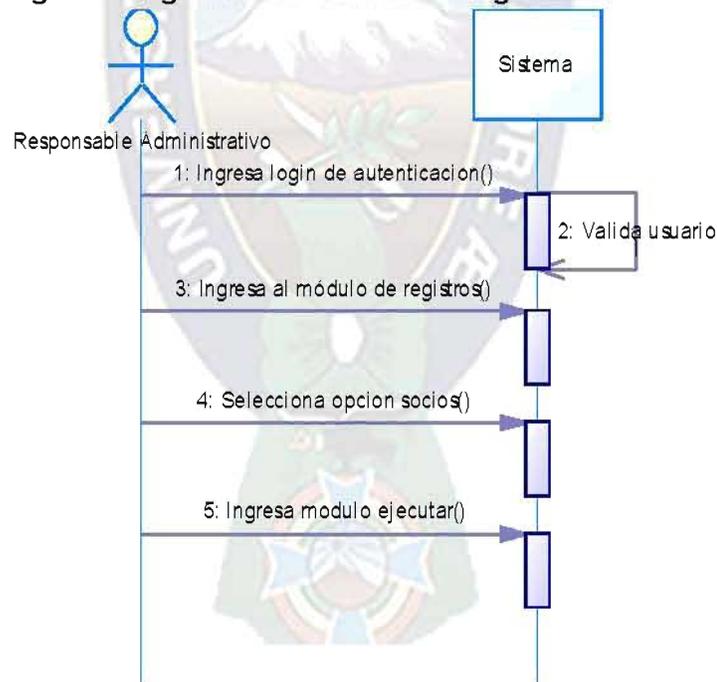
Tipo: primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El responsable administrativo o el secretario de actas ingresa al módulo de maquinaria pesada previa autenticación de cuenta.	2. Verifica y valida el código y contraseña del usuario. 3. Muestra la página correspondiente al módulo de registros.
4. El responsable Adm. o el strio. de actas selecciona la opción de socios e ingresa al módulo correspondiente.	5. Muestra la página de afiliación de socios. 6. Muestra los módulos a realizar(adicionar, eliminar, modificar socios)

Diagrama de secuencia:

Fig. 3.8 Diagrama de secuencia: Registro de socios



Fuente: Elaboración Propia

Contrato de operaciones:

Nombre: Ingresar login de autenticación

Responsabilidad: Muestra el formulario de autenticación de usuario del sistema.

Precondición: El usuario deberá estar registrado en la base de datos del sistema.

- Postcondición:** Se muestra el módulo de registro de Vehículos.
- Nombre:** Selecciona opción de socios.
- Responsabilidad:** Muestra el módulo de socios (propietarios de vehículos)
- Precondición:** El usuario selecciona el socio para su registro.
- Postcondición:** Se muestra el módulo seleccionado según el tipo de socio afiliado.
- Nombre:** Ingresa al módulo a ejecutar
- Responsabilidad:** Muestra los módulos de adición, modificación o eliminación socios.
- Precondición:** El usuario selección el tipo de consulta a ejecutar
- Postcondición:** Se muestra el módulo correspondiente de ejecución según la cédula de identidad del socio.

CASO DE USO: Registra vehículos

Actor: Responsable administrativo, secretario de actas

Propósito: Ingresar al módulo de registro de vehículos.

Resumen: El responsable administrativo o el secretario de actas ingresará al módulo correspondiente, previa autenticación, para su adición, modificación o eliminación de vehículos.

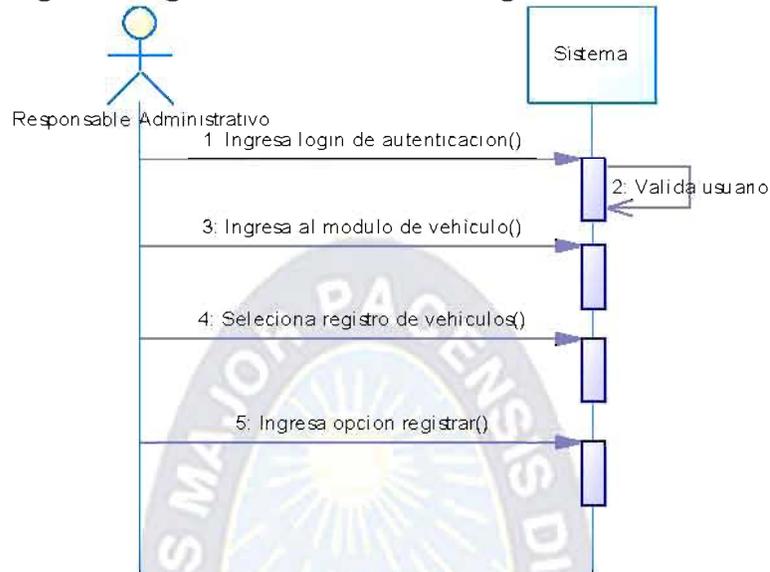
Tipo: Primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El responsable administrativo o el strio. de actas ingresa al módulo de vehículos previa autenticación de cuenta de usuario.	2. Verifica y valida el código y contraseña del usuario. 3. Muestra la página correspondiente al módulo de vehículos.
4. El responsable adm. o el strio. de actas selecciona la opción de vehículos e ingresa al módulo correspondiente.	5. Muestra la página de afiliación de vehículos. 6. Muestra los módulos a realizar como ser: adicionar, eliminar, modificar vehículos.

Diagrama de secuencia:

Fig. 3.9 Diagrama de secuencia: Registro de Vehículos



Fuente: Elaboración Propia

Contrato de operaciones:

Nombre: Ingresar login de autenticación

Responsabilidad: Muestra el formulario de autenticación de usuario del sistema.

Precondición: El usuario deberá estar registrado en la base de datos del sistema.

Postcondición: Se muestra el módulo de registro de Información.

Nombre: Seleccionar opción de Vehículos.

Responsabilidad: Muestra el módulo de Información de Vehículos.

Precondición: El usuario selecciona el tipo de vehículo para su registro.

Postcondición: Se muestra el módulo seleccionado según el tipo de vehículo.

Nombre: Ingresar al módulo a ejecutar

Responsabilidad: Muestra los módulos de adición, modificación o eliminación de Vehículos.

Precondición: El usuario selecciona el tipo de consulta a ejecutar

Postcondición: Se muestra el módulo correspondiente de ejecución según el número de placa que tiene cada vehículo y la cédula de identidad del socio.

CASO DE USO: Adicionar (socios-vehículos)

Actor: Responsable administrativo, secretario de actas.

Propósito. Registrar nuevos socios y vehículos correspondientes en el sistema.

Resumen: el strio. de actas registrará los datos requeridos para los socios y vehículos afiliados en el sindicato y guardar en la base de datos.

Tipo: primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El resp. Adm. o el strio. de actas ingresa en adicionar socio.	2. Muestra el formulario para el registro de datos del socio.
3. El resp. Adm. o el strio. de actas introduce los datos personales del socio y confirma la adición.	4. Registra y guarda los datos personales del socio en la base de datos del sistema.
5. El resp. Adm. o el strio. de actas ingresa en la opción adicionar vehículo.	6. Muestra y confirma la cuenta del socio registrado en la base de datos.
7. Introduce el ci del socio afiliado para adicionar los datos del vehículo correspondiente de operación.	8. Muestra el formulario para el registro de datos del vehículo.
9. El resp. Adm. o el strio de actas introduce datos del vehículo y confirma la adición.	10. Registra y guarda los datos vehiculares en la base de datos del sistema.

Diagrama de secuencia:

Fig. 3.10 Diagrama de secuencia: Adicionar (Socios - Vehículos)



Fuente: Elaboración propia

CASO DE USO: Modificar (socios-vehículos)

Actor: Responsable administrativo, secretario de actas.

Propósito: Modificar datos de socios y vehículos correspondientes en el sistema.

Resumen: el strio. de actas o el responsable administrativo modificará los datos requeridos para los socios y vehículos afiliados en el sindicato por medio del ci del socio y el nro. de placa correspondiente.

Tipo: primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El resp. Adm. o el strio. de actas ingresa en modificar socio o vehículo.	2. Muestra una cuenta para el registro de datos del socio o vehículo a modificar según el tipo de socio.
3. El resp. Adm. o el strio. de actas introduce el ci o placa del vehículo registrado para su actualización correspondiente.	4. Verifica los datos personales del socio en la base de datos del sistema. 5. Muestra el formulario de modificación de datos del vehículo o socio registrado.
6. El resp. Adm. o el strio de actas actualiza y confirma su modificación.	7. Guarda los datos vehiculares o del socio en la base de datos del sistema.

Diagrama de secuencia:

Fig. 3.11 Diagrama de secuencia: Modificar (Socios - Vehículos)



Fuente: Elaboración Propia

CASO DE USO: Eliminar (socios - vehículos)

Actor: Responsable administrativo, secretario de actas.

Propósito: Eliminar datos de socios y vehículos correspondientes en el sistema.

Resumen: el strio. de actas o el responsable administrativo eliminará los datos de los socios y vehículos afiliados en el sindicato por medio del ci del socio y el nro. de placa correspondiente.

Tipo: primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El resp. Adm. o el strio. de actas ingresa en la opción eliminar socio o vehículo.	2. Muestra una cuenta del registro del socio o vehículo a eliminar.
3. El resp. Adm. o el strio. de actas introduce el ci o placa del vehículo registrado para su eliminación correspondiente.	4. Verifica los datos personales del socio o vehículo en la base de datos del sistema. 5. Confirma eliminar los datos del socio o vehículo.
6. El resp. Adm. o el strio de actas elimina datos.	7. Elimina los datos vehiculares o del socio en la base de datos del sistema.

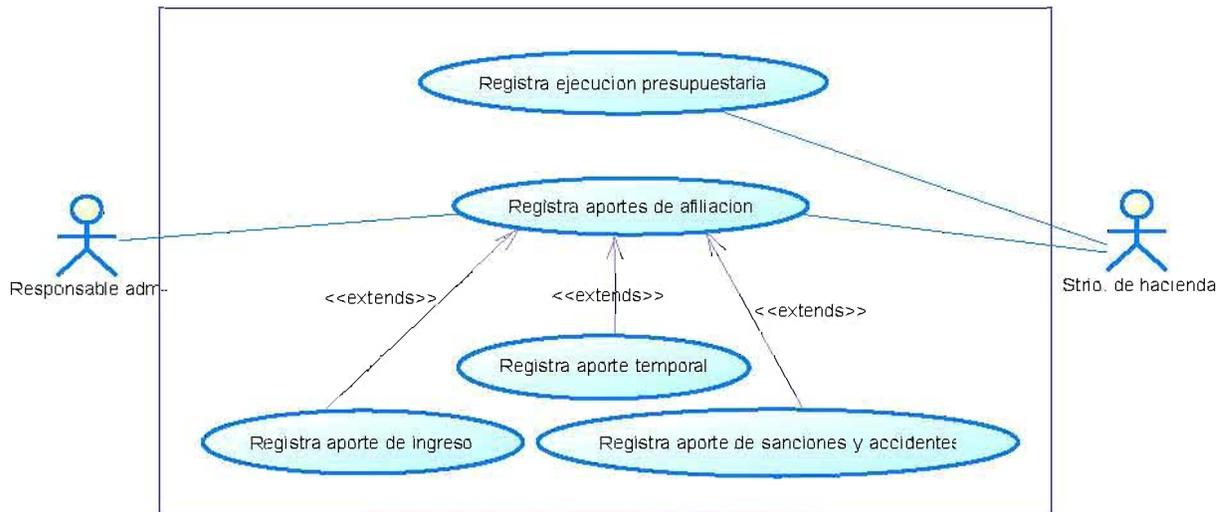
Diagrama de secuencia:

Fig. 3.12 Diagrama de secuencia: Eliminar (Socios – vehículos)



Fuente: Elaboración propia

Fig. 3.13 Diagrama de casos de uso: Registrar aportes de afiliación



Fuente: Elaboración Propia

CASO DE USO: Registrar aporte de ingreso

Actor: Responsable administrativo, secretario de hacienda

Propósito: Registrar el aporte de ingreso del nuevo socio afiliado al sindicato.

Resumen: El responsable adm. o el strio. de hacienda registran los datos necesarios para el único aporte de ingreso del socio por cada vehículo de su propiedad registrado en la base de datos.

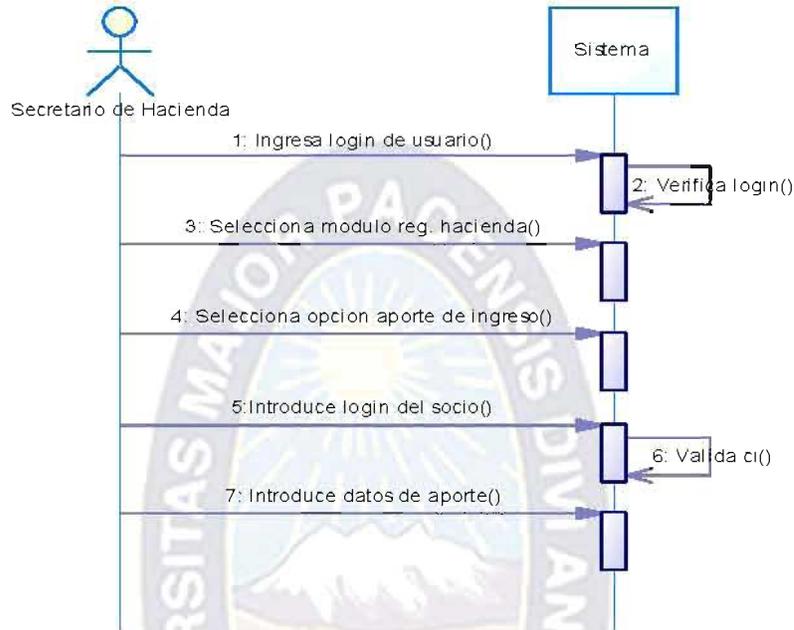
Tipo: Primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El resp. adm. o el strio. de hacienda ingresa al módulo de registro de hacienda para registrar los tipos de aportes que hace el socio previa autenticación de usuario.	2. Verifica y valida el código y contraseña del usuario. 3. Muestra modulo correspondiente al depto. de hacienda.
4. El resp. adm. o el strio. de hacienda selecciona la opción de aporte de ingreso e introduce la placa del vehículo.	5. Muestra la cuenta de registro del Vehículo del socio. 6. Muestra formulario de aporte de ingreso para el socio.
7. Introduce los datos necesarios para dicha tarea.	8. Registra y guarda los datos de aporte inicial por cada vehículo de propiedad del socio en la base de datos.

Diagrama de secuencia:

Fig. 3.14 Diagrama de secuencia: Registrar aporte de ingreso



Fuente: Elaboración Propia

Contrato de operaciones:

Nombre: Ingresar login de autenticación

Responsabilidad: Muestra el formulario de autenticación de usuario del sistema.

Precondición: El usuario deberá estar registrado en la base de datos del sistema.

Postcondición: Se muestra el módulo de régimen de hacienda.

Nombre: Selecciona opción de aporte de ingreso

Responsabilidad: Muestra el módulo de aporte de ingreso por cada vehículo registrado de su propiedad.

Precondición: Realizan el aporte de ingreso los socios propietarios

Postcondición: Se muestra la cuenta de socio y vehículo registrado.

Nombre: Ingresar número de placa del vehículo

Responsabilidad: Registrar los datos de aporte de ingreso del vehículo a la entidad.

Precondición: El vehículo deberá estar registrado en la base de datos.

Postcondición: Se guarda los datos de aporte por cada vehículo de propiedad del socio en la base de datos

CASO DE USO: Registrar aportes temporales

Actor: Responsable administrativo, secretario de hacienda

Propósito: Registrar los aportes temporales del socio afiliado

Resumen: El resp. administrativo o el Strio. de hacienda registran los datos de aportes temporales (semanal o mensual) en el sistema de acuerdo al tipo de socio y por cada vehículo de operación registrado.

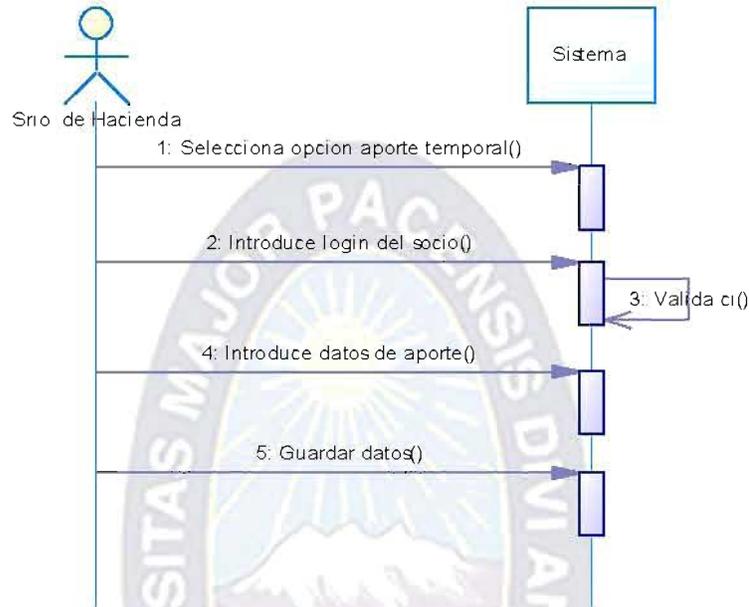
Tipo: Primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El resp. administrativo o el Strio. de hacienda ingresa al modulo de reg. de hacienda y selecciona la opción de aporte temporal.	2. Muestra la cuenta del vehículo registrado del socio.
3. Introduce el nro. de placa del vehículo de operación del socio.	4. Verifica y valida el número de placa. 5. Despliega el formulario de aporte temporal.
6. Introduce los datos necesarios para el aporte temporal (semanal o mensual) y lo adiciona.	7. Registra y guarda los datos del aporte realizado en la base de datos.

Diagrama de secuencia:

Fig. 3.15 Diagrama de secuencia: Registrar aportes temporales



Fuente: Elaboración Propia

Contrato de operaciones:

Nombre: Ingresa opción aporte temporal

Responsabilidad: Muestra el módulo de aporte temporal por cada vehículo registrado

Precondición: Realizan el aporte temporal los socios propietarios.

Postcondición: Se muestra la cuenta del socio y vehículo registrado.

Nombre: Introduce número de placa del vehículo.

Responsabilidad: Verifica si el vehículo se halla registrado en la base de datos.

Precondición: El vehículo deberá estar registrado en la base de datos.

Postcondición: Se muestra el formulario de aporte temporal.

Nombre: Introduce datos de aportes temporales

Responsabilidad: Registrar los datos de aporte temporal del vehículo

Precondición: El aporte será de forma semanal o mensual dependiendo de las necesidades del sindicato.

Postcondición: Se guardan los datos de aporte por cada vehículo de propiedad del socio en la base de datos.

CASO DE USO: Registrar Aportes de sanciones o accidentes

Actor: Responsable administrativo, Secretario de hacienda.

Propósito: Registrar los aportes del socio según el tipo de sanción o accidente cometido.

Resumen: El responsable administrativo o el Strio. de hacienda registran los datos de aporte del socio, según las infracciones o accidentes cometidos en el grupo designado.

Tipo: Primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El responsable Administrativo o Strio. de hacienda ingresan al módulo de reg. de hacienda seleccionan la opción de aporte de sanciones o accidentes.	2. Muestra la cuenta del socio afiliado.
3. Introduce el c.i. del socio.	4. Verifica y valida el c.i. del socio. 5. Despliega el formulario de aporte de sanciones o accidentes.
6. El responsable adm. o el strio. de hacienda introduce los datos necesarios para el aporte de sanciones y lo adiciona.	7. Registra y guarda los datos del aporte realizado en la base de datos.

Diagrama de secuencia:

Fig. 3.16 Diagrama de secuencia: Registrar aportes de sanciones o accidentes



Fuente: Elaboración Propia

Contrato de operaciones:

Nombre: Ingresa opción aporte (sanción/accidente)

Responsabilidad: Muestra el módulo de aporte seleccionado según el tipo de sanción.

Precondición: Realizan el aporte de los socios propietarios.

Postcondición: Se muestra la cuenta del socio registrado.

Nombre: Introduce login del socio

Responsabilidad: Verifica si el socio se halla registrado en la base de datos.

Precondición: El socio deberá estar registrado en la base de datos.

Postcondición: Se muestra el formulario de aporte de sanción/accidente.

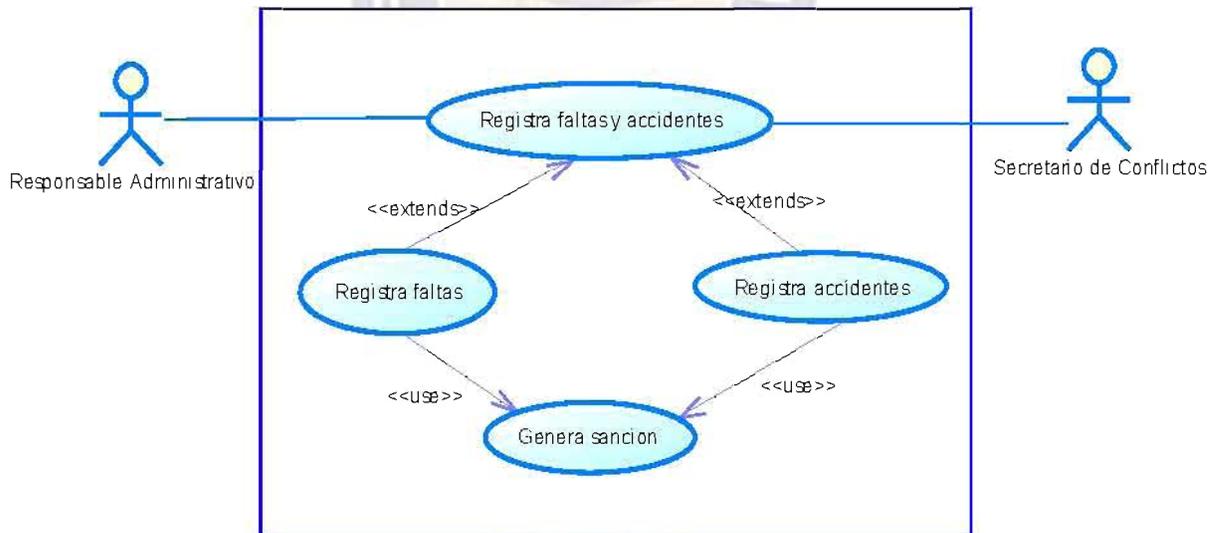
Nombre: Introduce datos de aportes

Responsabilidad: Registrar los datos de aporte de sanción/accidente

Precondición: El aporte será según el tipo de falta cometida por el socio y el vehículo de operación.

Postcondición: Se guardan los datos de aporte en la base de datos.

Fig. 3.17 Diagrama de Casos de Uso: Registrar faltas y accidentes



Fuente: Elaboración Propia

CASO DE USO: Registra faltas

Actor: Responsable administrativo, Secretario de conflictos.

Propósito: Registrar las faltas del socio infractor afiliado al sindicato

Resumen: El responsable administrativo o el Strio. de conflictos registra los datos necesarios de faltas cometidas por el socio según el grado de falta, para su posterior sanción.

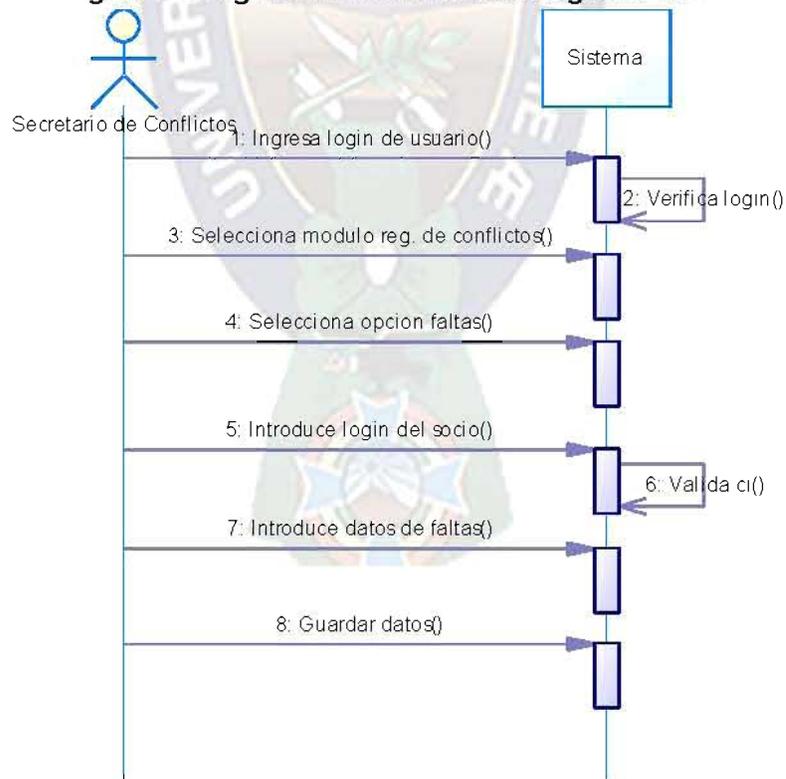
Tipo: Primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El responsable adm. o el strio. de conflictos ingresa al modulo de reg. de conflictos para registrar los datos de faltas cometidas previa autenticación de usuario.	2. Muestra la cuenta del socio infractor.
3. Introduce el ci. del socio.	4. Verifica y valida el ci. del socio 5. Despliega el formulario de faltas cometidas.
6. Introduce los datos necesarios en el formulario.	7. Registra y guarda los datos en la base de datos.

Diagrama de secuencia:

Fig. 3.18 Diagrama de secuencia: Registra faltas



Fuente: Elaboración Propia

CASO DE USO: Registrar Accidentes

Actor: Responsable Administrativo, Secretario de Conflictos.

Propósito: Registrar el tipo de accidente cometido por el socio y el vehículo correspondiente de operación.

Resumen: El resp. adm. o el strio. de conflictos registran los datos necesarios del accidente cometido por el socio para su posterior sanción.

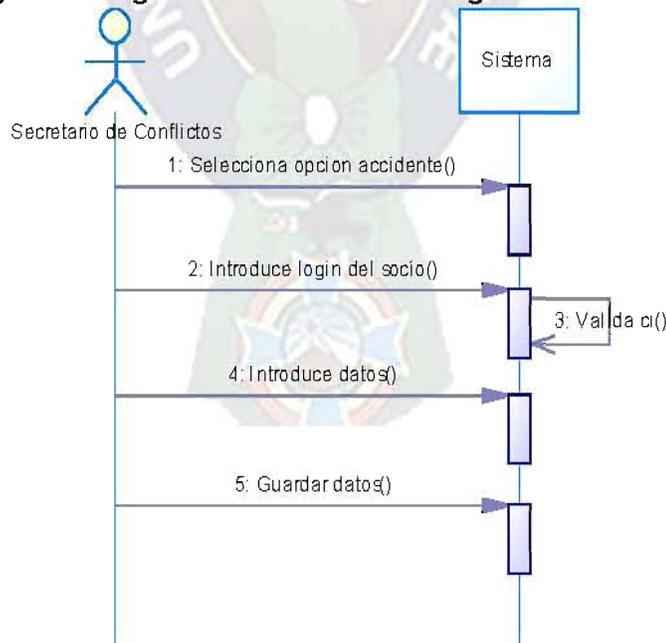
Tipo: Primario esencial

Curso normal de eventos:

Acción del actor	Respuesta del sistema
1. El responsable administrativo o el strio. de conflictos ingresa al modulo de reg. de conflictos para registrar los datos de accidentes cometidos previa autenticación del socio infractor.	2. Muestra la cuenta del socio.
3. Introduce ci. del socio.	4. Verifica y valida ci. del socio. 5. Verifica el formulario de accidentes cometidos
6. Introduce los datos necesarios en el formulario.	7. Registra y guarda los datos en la base de datos

Diagrama de secuencia:

Fig. 3.19 Diagrama de secuencia: Registra Accidentes

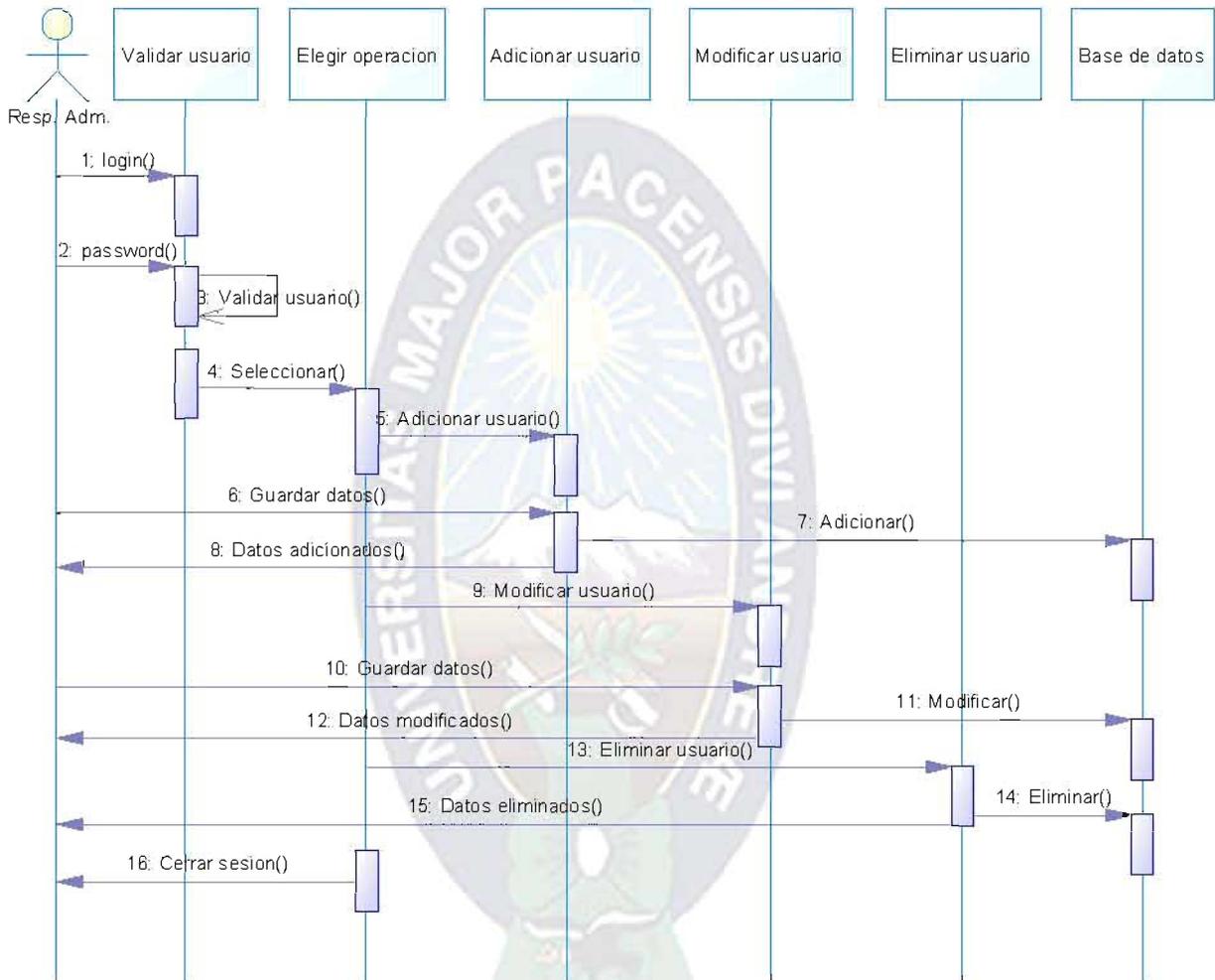


Fuente: Elaboración Propia

3.5 ETAPA DE CONSTRUCCION

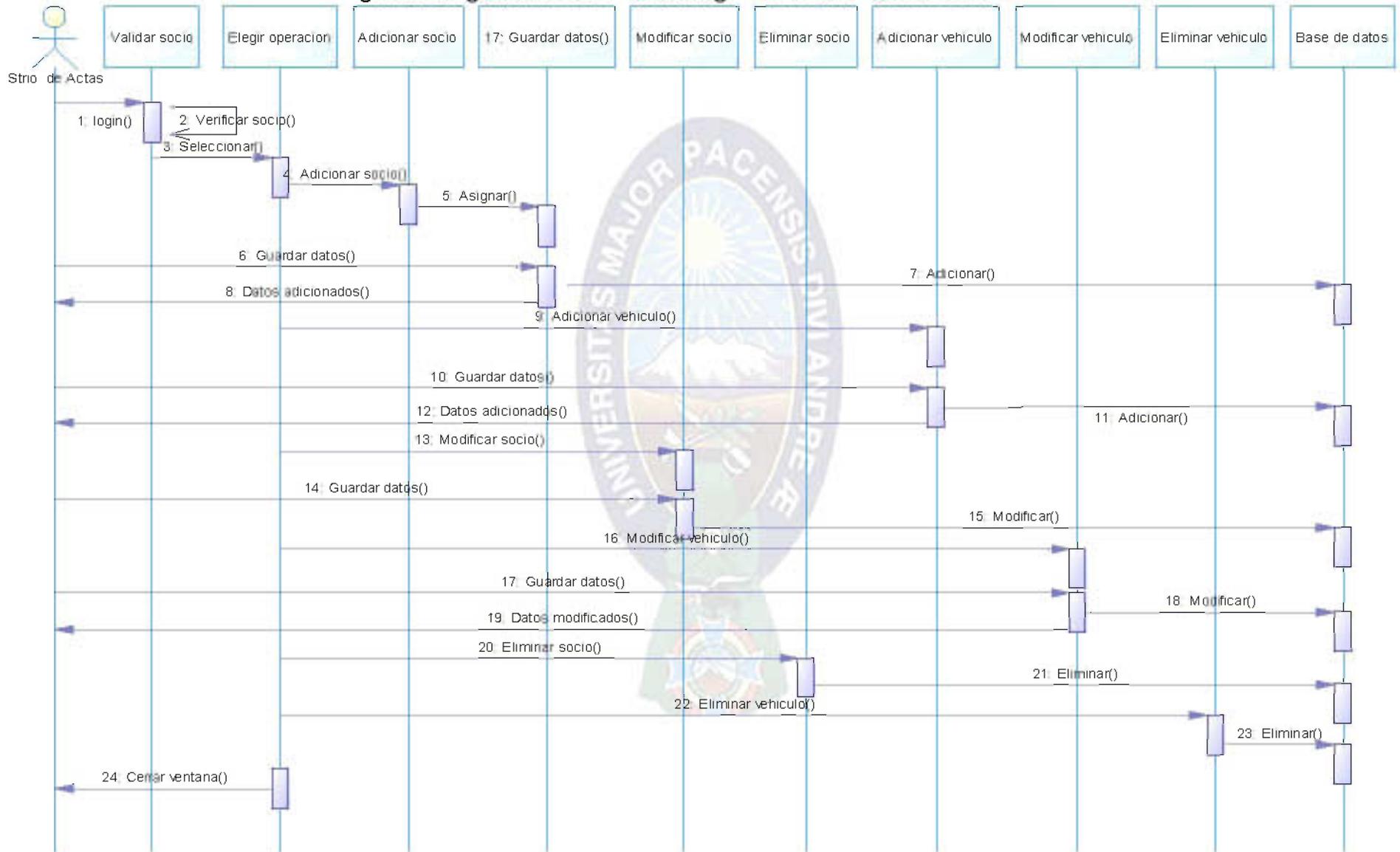
3.5.1 DIAGRAMAS DE SECUENCIAS PARA EL DISEÑO

Fig. 3.20 Diagrama de Secuencia: Autenticación de Cuentas



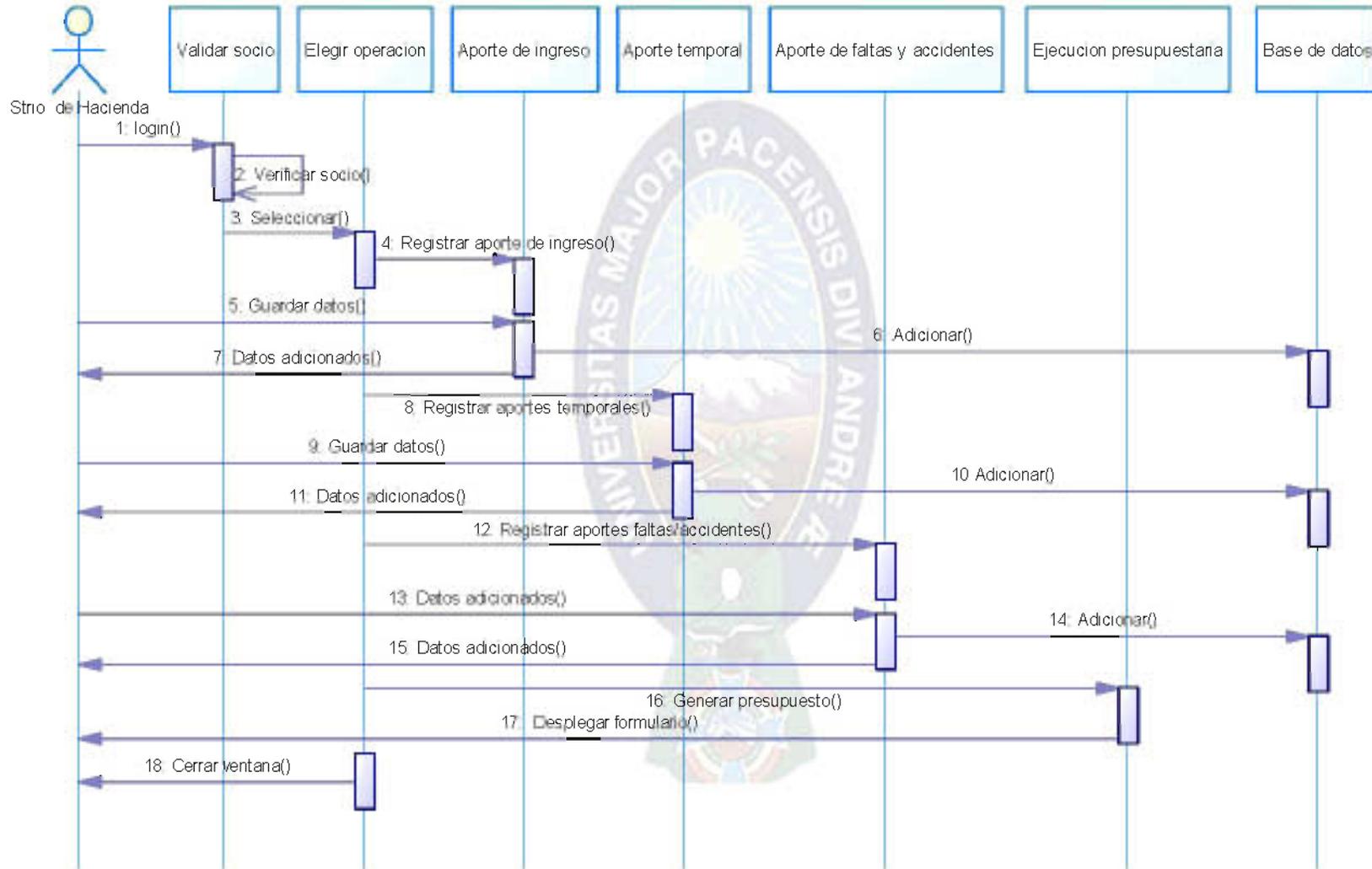
Fuente: Elaboración Propia

Fig. 3.21 Diagrama de Secuencia. Registro de Afiliación de Socios



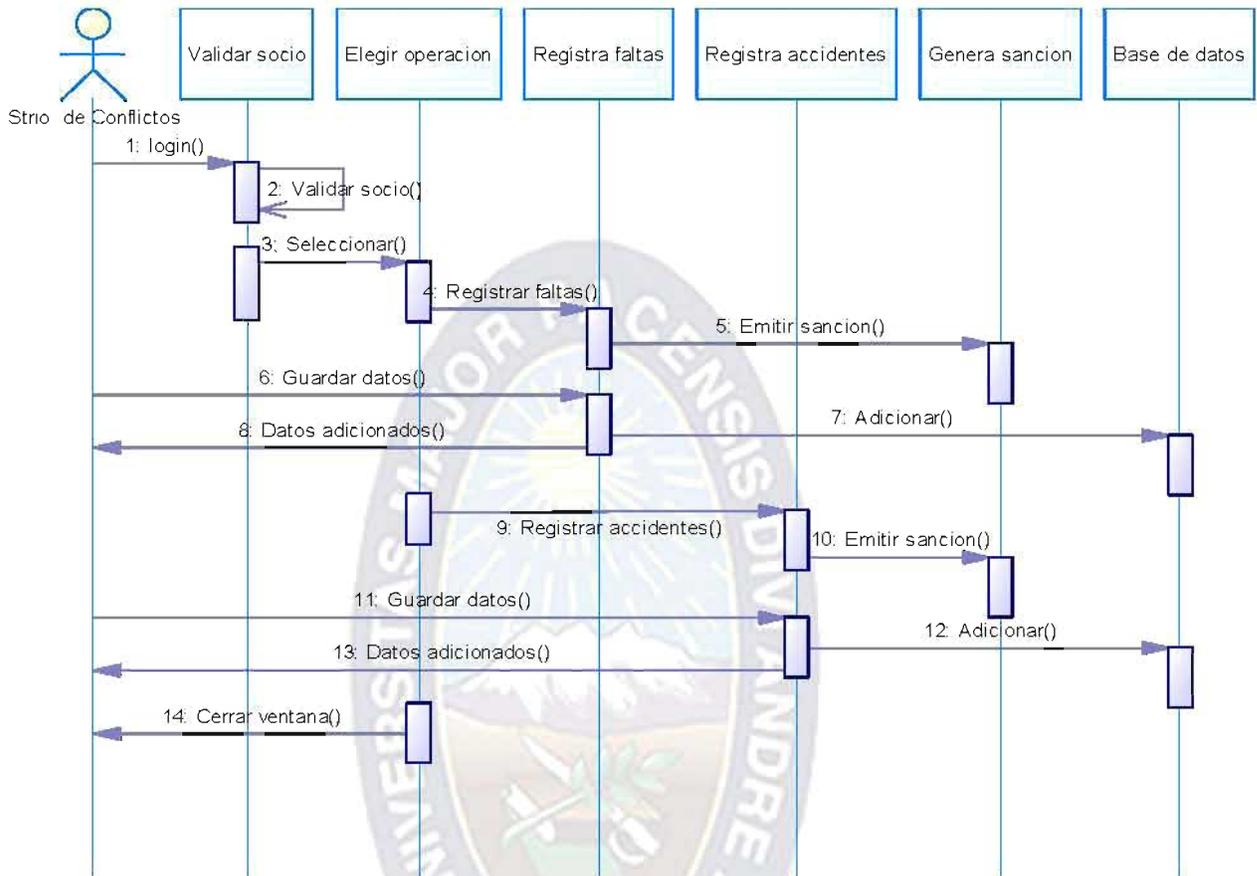
Fuente: Elaboración Propia

Fig. 3.22 Diagrama de Secuencia. Registro de Aportes de Afiliación



Fuente: Elaboración Propia

Fig. 3.23 Diagrama de Secuencia. Registro de faltas y accidentes



Fuente: Elaboración Propia

3.5.2 DIAGRAMA DE ESTADO

Cada diagrama de estado constituye una entrada de valor para la implementación de su correspondiente clase de diseño. Se describe en función a la realización de casos de uso.

Fig. 3.24 Diagrama de Estado: Autenticación de Cuentas



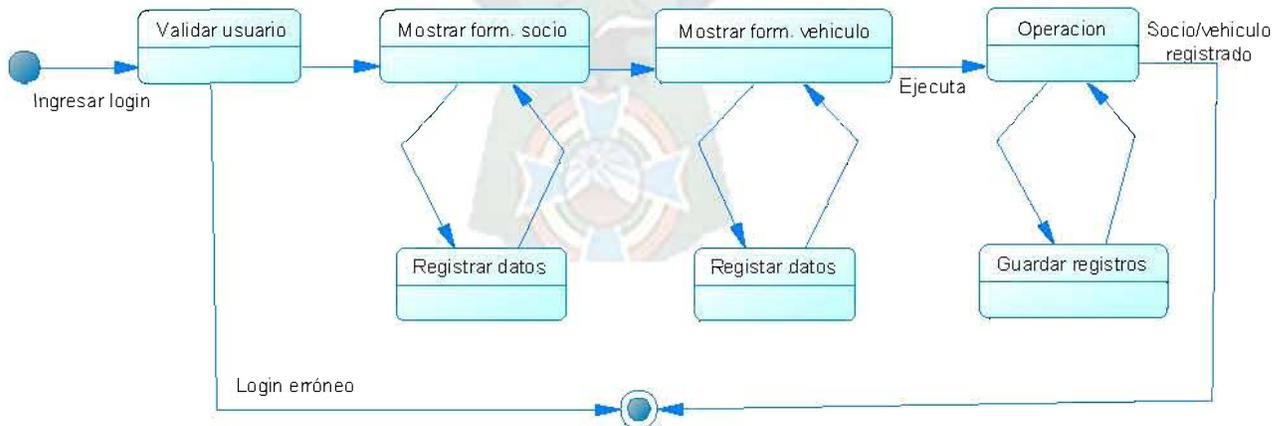
Fuente: Elaboración Propia

Fig. 3.25 Diagrama de Estado: Registrar Aportes



Fuente: Elaboración Propia

Fig. 3.26 Diagrama de Estado: Registrar Afiliación



Fuente: Elaboración Propia

3.5.3 DIAGRAMA DE COMPONENTES

Representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

Nos muestra la estructura física del código en términos de los componentes de código. Las dependencias entre estos, son afectadas por un cambio en uno de los componentes del sistema.

Fig. 3.27 Diagrama de Componentes del Sistema



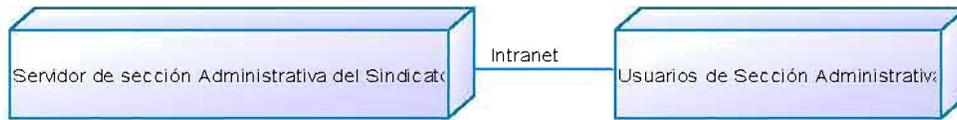
Fuente: Elaboración Propia

3.5.4 DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue nos mostrará la configuración que se realizará para el sistema propuesto, formado por dos nodos mediante red: el servidor del sistema quien está a cargo la sección administrativa del sindicato de grúas, montacargas y equipo pesado, y estarán comunicados a través de un intranet.

En el servidor se instalará el sistema de información y control para el sindicato de grúas, montacargas y equipo pesado La Paz, se registraran a usuarios del sistema en la base de datos, los administradores se conectaran directamente al servidor actualizando su cuenta de usuario, para después generar los reportes requeridos.

Fig. 3.28 Diagrama de Despliegue: Base de Datos - Sistema



Fuente: Elaboración Propia

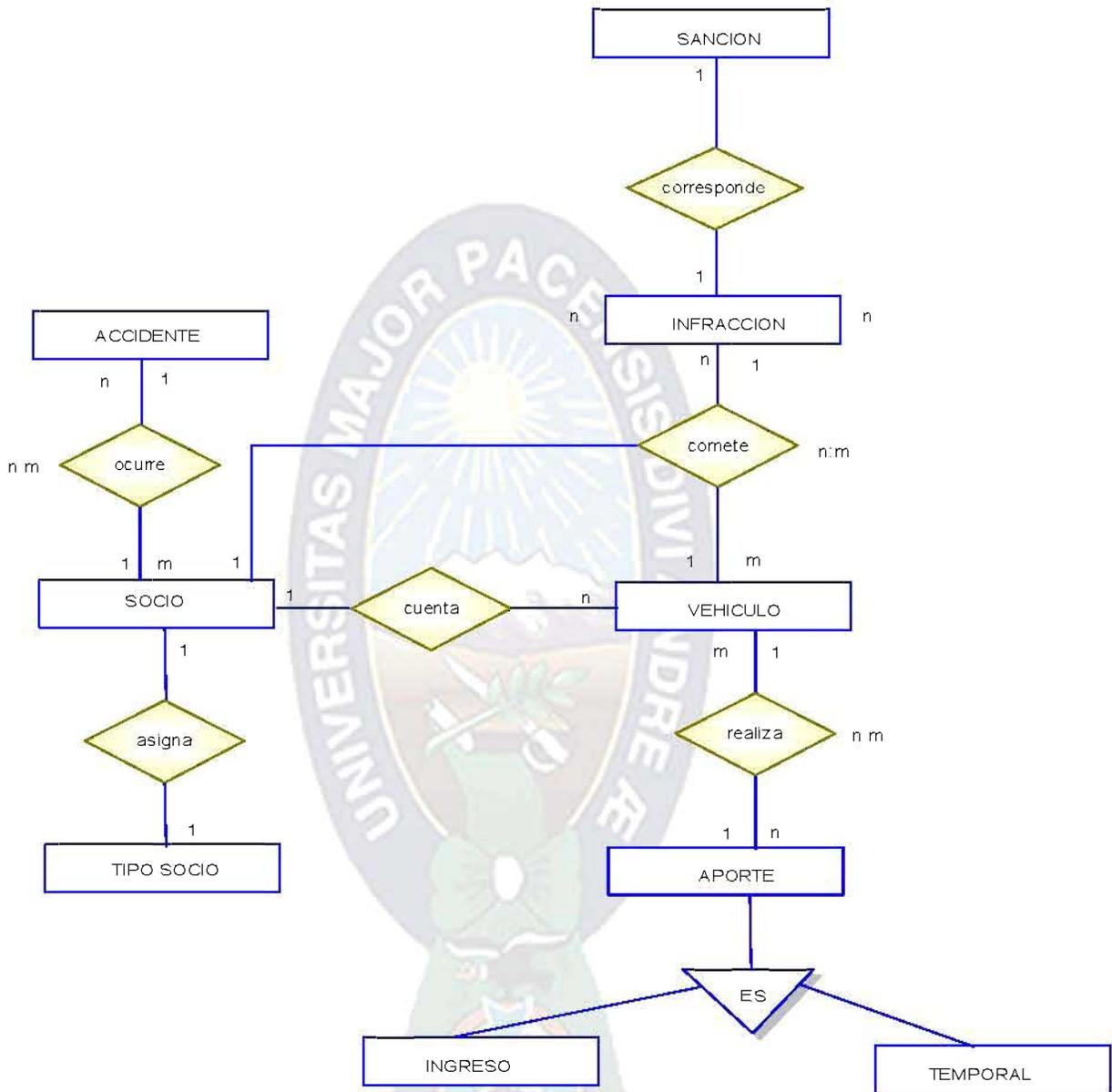
3.5.5 MODELO ENTIDAD-RELACION

El modelo entidad-relación es una herramienta para el modelado de datos de un sistema de información, mediante el cual se pretende mostrar los objetos que pertenecen a la base de datos, éstas son las entidades. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

En la siguiente figura se muestra las entidades relevantes del sistema con la base de datos.



Fig. 3.29 Modelo Entidad – Relación



Fuente: Elaboración Propia

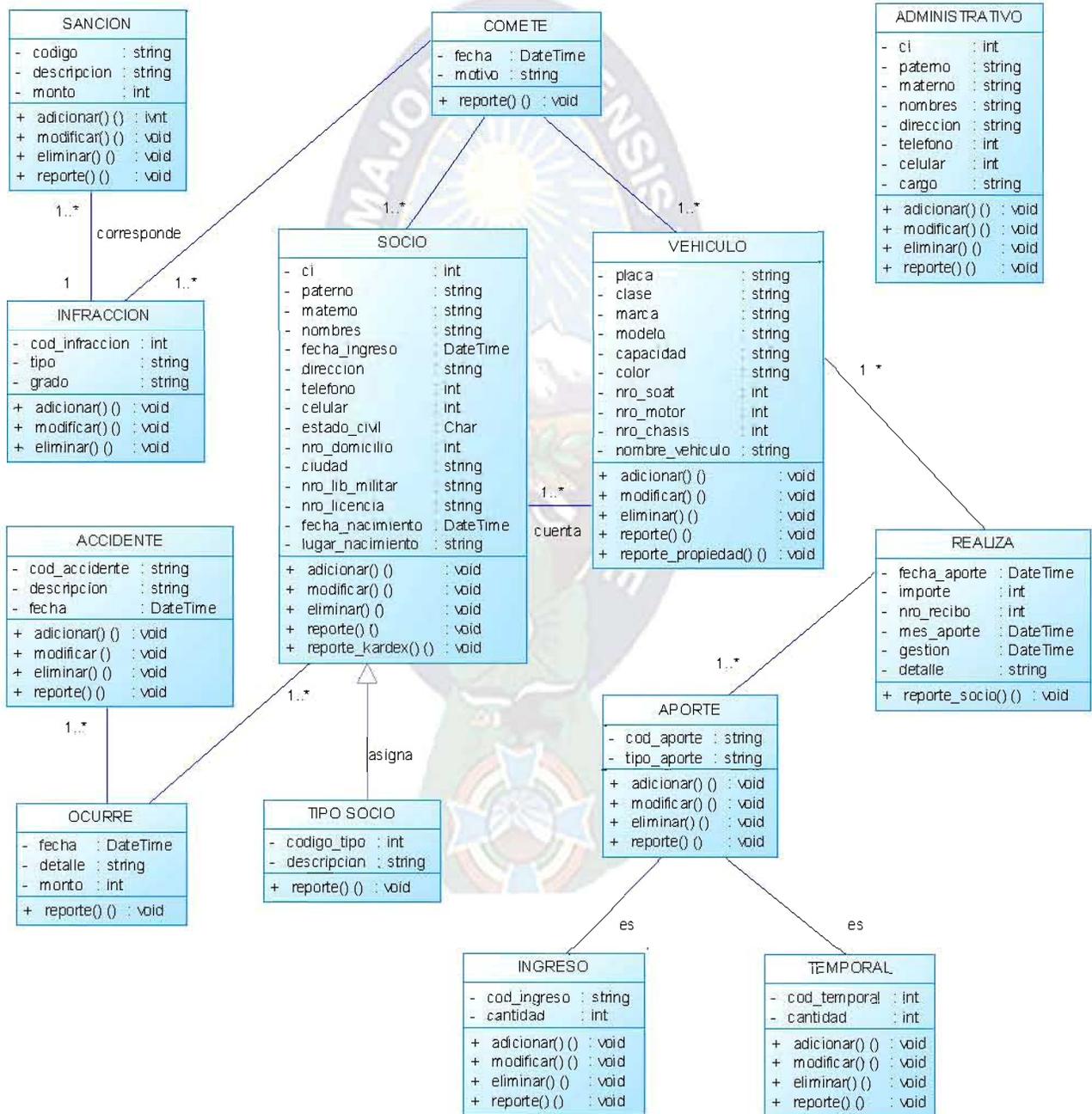
3.5.6 DIAGRAMA DE CLASES

Un diagrama de clases es estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se

manejará en el sistema, los componentes se encargaran del funcionamiento, la relación entre uno y otro.

Estos diagramas señalan los tipos de objetos que existen en el sistema junto a sus diversas relaciones estáticas y se podrá observar los métodos y atributos correspondientes.

Fig. 3.30 Diagrama de Clases

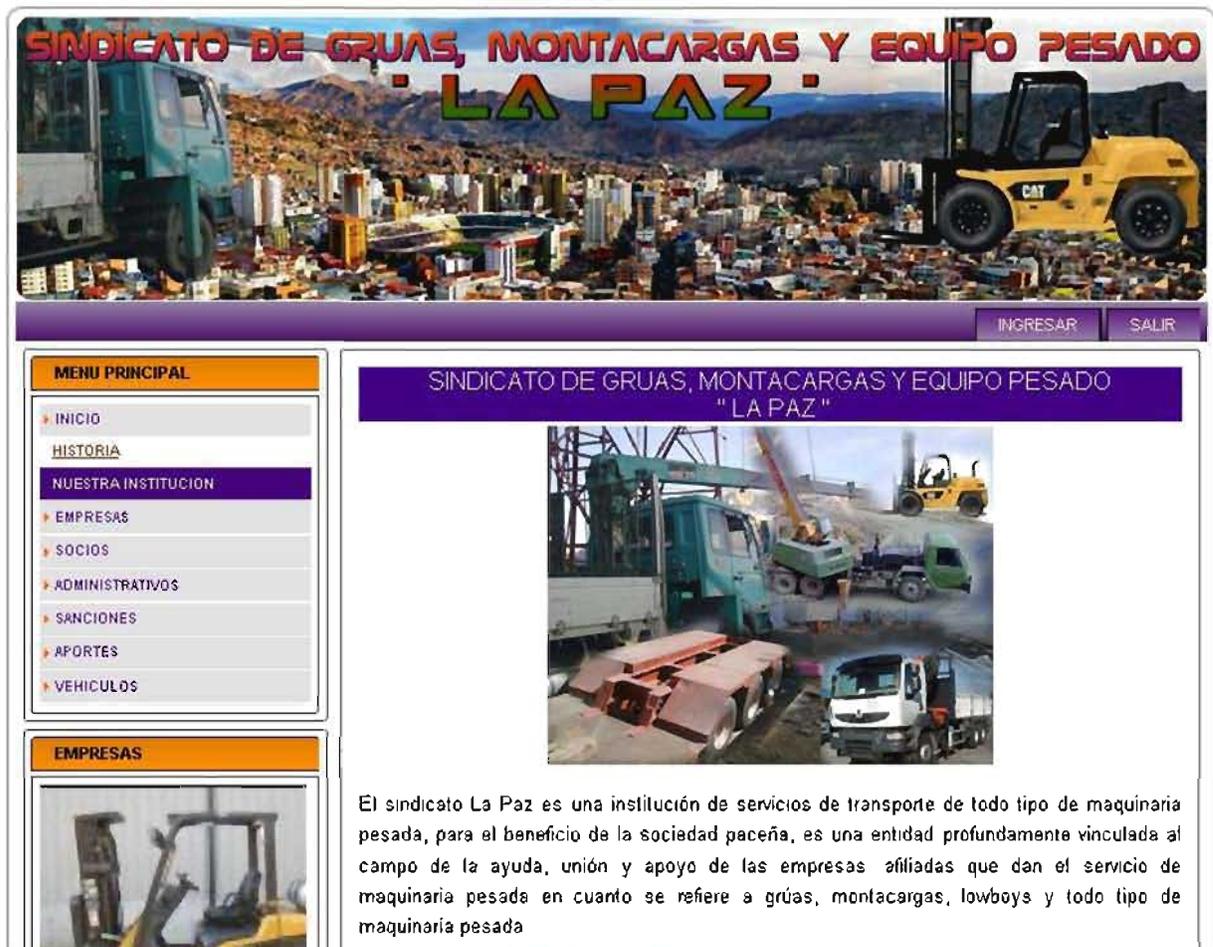


Fuente: Elaboración Propia

3.6 FASE DE IMPLEMENTACION

Ahora se mostrará algunas capturas de pantallas de la implementación del sistema de acuerdo al diseño elaborado. Esta fase nos lleva a exponer las interfaces gráficas y el esquema de codificación de ventanas para obtener una descripción fluida, como se muestran en las siguientes figuras elaborados a lo largo de esta fase.

Fig. 3.31 Página de Inicio de Interfaz con el sistema



Fuente: Elaboración Propia

Fig. Registrar de Nuevo Socio



MENU PRINCIPAL	
▶ INICIO	
▶ EMPRESAS	
▶ SOCIOS	
<u>BUSCAR</u>	
<u>REGISTRAR</u>	
<u>ACTUALIZAR</u>	
<u>ELIMINAR</u>	
▶ ADMINISTRATIVOS	
▶ SANCIONES	
▶ APORTES	
▶ VEHICULOS	

EMPRESAS	
	

REGISTRO DE SOCIOS

CI	<input type="text" value="4335566"/>
APELLIDO PATERNO	<input type="text" value="CHOQUE"/>
APELLIDO MATERNO	<input type="text" value="TARQUI"/>
NOMBRES	<input type="text" value="JUAN"/>
DIRECCION	<input type="text" value="CALLE/AVENIDA BUENOS AIRES"/>
	Nº <input type="text" value="1233"/> ZONA <input type="text" value="3 DE MAYO"/>
FECHA DE NACIMIENTO	<input type="text" value="7/16/1978"/> 
TELEFONO	<input type="text" value="2323344"/>
CELULAR	<input type="text" value="77566477"/>
ESTADO CIVIL	<input type="text" value="CASADO"/> ▼
CIUDAD	<input type="text" value="POTOSI"/> ▼
NRO LIBRETA MILITAR	<input type="text" value="2334433444"/>
NRO LICENCIA	<input type="text" value="34455667"/>

Fuente: Elaboración Propia



Fig. Eliminar Socios



MENU PRINCIPAL

- ▶ INICIO
- ▶ EMPRESAS
- ▶ SOCIOS
- BUSCAR
- REGISTRAR
- ACTUALIZAR
- ELIMINAR
- ▶ ADMINISTRATIVOS
- ▶ SANCIONES
- ▶ APORTES
- ▶ VEHICULOS

ELIMINAR SOCIOS

CI

C.I.	Ap. Paterno	Ap. Materno	Nombre	Fecha Ingreso	Telefono	Celular	Edad
4335566	CHOQUE	TARQUI	JUAN	07/11/2011 08:59:04 p m	CHOQUE	77566477	31
4929774	ARCE	ALARCON	MARIA SOLEDAD	06/11/2011 06:47:26 p m	ARCE	77594334	31

Fuente: Elaboración Propia

Fig. Actualizar Socio

SINDICATO DE GRUAS, MONTACARGAS Y EQUIPO PESADO "LA PAZ"

INGRESAR SALIR

MENU PRINCIPAL

- ▶ INICIO
- ▶ EMPRESAS
- ▶ SOCIOS
- ◀ BUSCAR
- ◀ REGISTRAR
- ▶ ACTUALIZAR**
- ◀ ELIMINAR
- ▶ ADMINISTRATIVOS
- ▶ SANCIONES
- ▶ APORTES
- ▶ VEHICULOS

EMPRESAS

ACTUALIZAR SOCIOS

CI: 4929774

APELLIDO PATERNO: ARCE

APELLIDO MATERNO: ALARCON

NOMBRES: MARIA SOLEDAD

DIRECCION: CALLE/AVENIDA ABDON ONDARZA

N° 1633 ZONA SAN PEDRO

FECHA DE INGRESO: 06/11/2011 06:47:26 p.m.

FECHA DE NACIMIENTO:

TELEFONO: ARCE

CELULAR: 77594334

ESTADO CIVIL: CASADO

CIUDAD: LA PAZ

NRO LIBRETA MILITAR: NO TIENE

NRO LICENCIA: NO TIENE

Fuente: Elaboración Propia

3.7 PRUEBAS

Se llevaron a cabo dos tipos de pruebas en el sistema. Las primeras pruebas funcionales se hicieron para estar seguros que el sistema está funcionando como se espera o como fue diseñado. Estas pruebas se llevaron a cabo creando un grupo de casos de prueba con datos de prueba. Las pruebas incluyeron la validación de campos, reglas de negocio, integración de procesos y la entrada de datos. Las pruebas fueron realizadas en ambos ambientes presentados de cliente/servidor.

Las segundas pruebas fueron preparadas para asegurarse que el sistema era capaz de manejar el volumen de datos y el tiempo de respuesta al usuario era el esperado. Estas

pruebas fueron llevadas a cabo poniendo atención especial a los procesos de creación de los nuevos aportes y demás elementos que componen a la función del sindicato.

3.8 FASE DE TRANSICION

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

En la etapa del análisis, se estableció que el sistema debe apoyar a las funciones del Administrativo y del secretario general del sindicato, mostrando y emitiendo información y documentos que este requiera, una vez refinado y aprobado, el sistema será instalado en la oficina del sindicato de transporte de grúas y montacargas "La Paz".

3.9 IMPLEMENTACION 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 sino de forma.

3.10 INSTALACION DEL SISTEMA

El nuevo sistema no necesita ser instalado, se ingresará mediante un navegador como ser: Internet Explorer, Mozilla, Google Chrome y otros navegadores existentes, teniendo previamente la dirección de la página web dentro de la empresa. El Administrador o el Secretario General podrá ingresar al sistema para realizar las pruebas finales de funcionamiento, antes de ser entregado como producto final a los usuarios que utilizarán el sistema

CAPITULO IV

CALIDAD DE SOFTWARE

La evolución de un software va surgiendo de acuerdo a la corrección de los errores, se mejora el funcionamiento según los nuevos requerimientos del cliente, aunque no siempre se desarrollaron sistemas de forma controlada, ya que en la actualidad existen algunas aplicaciones que presentan grandes dificultades para su mantenimiento y actualización.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Tener una buena calidad de software es una dedicación con muchos esfuerzos, sin embargo, el software casi nunca es perfecto. El objetivo de tener un buen proyecto de software es tener la mejor calidad posible, que cumpla y supere las expectativas de los usuarios.

La calidad de concordancia es un aspecto centrado principalmente en la implementación; si la implementación sigue al diseño, y el sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta.

Adicionalmente se puede seguir los siguientes aspectos principales para evaluar la calidad del software:

- ✓ Funcionalidad
- ✓ Confiabilidad
- ✓ Eficiencia
- ✓ Facilidad de mantenimiento
- ✓ Portabilidad,

La gestión de la calidad es el aspecto de la función de gestión que determina y aplica la política de la calidad, los objetivos y las responsabilidades y que lo realiza con medios tales como la planificación de la calidad, el control de la calidad, la garantía de calidad y la mejora de la calidad.

La calidad de software no se certifica, lo que se certifica son los procedimientos para construir un software de calidad, los procedimientos deben ser correctos y estar en función de la normalización.

Existen varias métricas de medición que nos permiten medir la calidad del software, uno de ellos es el organismo de normalización ISO (International Standards Organization) que ha definido los requisitos de un sistema de gestión de calidad de software de carácter general que cubre el desarrollo de cualquier producto. Para el presente proyecto se toma en cuenta la métrica estándar de calidad ISO 9126.

4.1 LA NORMA DE CALIDAD ISO 9126

La norma de calidad ISO 9126 fue desarrollada en un intento de identificar los atributos clave de calidad para el software. El estándar identifica seis atributos clave de calidad que a continuación se desarrollan:

4.1.1 CONFIABILIDAD

Este atributo se relaciona con la reducción en la frecuencia de fallas en un intervalo de tiempo dado. Frecuentemente se expresa con la fórmula:

La confiabilidad es la cantidad de tiempo que el software está disponible para su uso. Está referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.

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

Donde: λ Es la tasa constante de fallo y t es el intervalo de tiempo

Durante las pruebas al sistema de información y control para el Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz", se presenta un error mínimo del 0.001 para un intervalo de 4 días con duración de 8 horas diarias. Se reemplazan los valores en la fórmula y se obtiene lo siguiente:

$$R(t) = e^{-\lambda t} = e^{-(0.001 \cdot 32)}$$

$$R(t) = 0.968 \approx 97\%$$

Éste resultado significa que en 4 días de prueba con ocho horas de trabajo diario, la confiabilidad del sistema mediante la distribución exponencial es del 97%.

4.1.2 FUNCIONALIDAD

La funcionalidad del sistema es medida a través de métricas de software orientadas a funciones como es el de Punto de Función, que proporciona una media objetiva y cuantitativa basada en la visión del usuario final de la aplicación.

Se determinan cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada de la tabla. Los valores de los dominios de información se definen de la forma siguiente:

- *Número de entradas de usuario.* Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.
- *Número de salidas de usuario.* Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuenta de forma separada.
- *Número de peticiones de usuario.* Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- *Número de archivos.* Se cuenta cada archivo maestro lógico (grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).

- *Número de interfaces externas.* Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

En la siguiente tabla 4.1 se muestra las cinco características del dominio de la información para su posterior cálculo de punto de función.

Tabla 4.1: Descripción de parámetros de medición

<u>ENTRADAS</u>	<u>SALIDAS</u>
<u>CONSULTAS</u> Informe de kardex personal de socios Información de datos de tipos de aportes Información de datos de administrativos Información de datos de vehículos de propiedad de socios. Datos de aporte de accidentes y beneficencia. Datos de socios/vehículos Datos de infracciones	<u>TABLAS</u> Administrativo Vehículo Socio Cuenta – vehículo Asigna tipo - socio Aporte Aporte temporal Aporte de ingreso Realiza - aporte

Datos de sanciones	Accidente Infracción Sanción Comete - accidente Corresponde – sanción
<u>INTERFACES EXTERNAS</u>	
Disco extraíble CD's Copias de Seguridad Impresora	

Fuente: Elaboración Propia

En la siguiente tabla 4.2 se muestra la sumatoria de los datos anteriores globalizando con los parámetros de medición y asociados a los factores de ponderación. Determinar la complejidad es un tanto subjetivo.

Tabla 4.2: Métricas de Puntos de Función

PARÁMETROS DE MEDICIÓN	CUENTA	SIMPLE	MEDIO	COMPLEJO	TOTAL
Nro. de entradas de usuario	11	3	4	6	44
Nro. de salidas de usuario	9	4	5	7	45
Nro. de peticiones de usuario	8	3	4	6	32
Nro. de archivos	14	7	10	15	140
Nro. de interfaces externas	4	5	7	10	28
CUENTA_TOTAL					289

Fuente: Elaboración propia

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

$$PF = CUENTA_TOTAL * [R(t) + 0.01 * SUM(fi)]$$

Donde:

CUENTA_TOTAL: Es la suma de todas las entradas de PF obtenidas de la tabla anterior.

fi: Donde i puede ser de 1 hasta 14 los valores de ajuste de complejidad basados en las respuestas a las cuestiones señaladas en la siguiente tabla 4.3. Cada pregunta es respondida usando la escala de rangos desde 0 (sin influencia), 1 (Incidental), 2 (Moderado), 3 (Medio), 4 (Significativo) hasta 5 (absolutamente esencial).

Los factores de peso que se aplican al cuestionario, se determinan empíricamente.

Tabla 4.3: Valores de ajuste de complejidad

Fi	FACTOR Fi	VALOR (0 - 5)
1	¿Requiere el sistema copias de seguridad y de recuperación fiable?	5
2	¿Se requiere comunicación de datos?	5
3	¿Existen funciones de procedimiento distribuido?	3
4	¿Es crítico el rendimiento?	4
5	¿Se ejecutará el sistema en un entorno operativo y fuertemente utilizado?	5
6	¿Requiere entrada de datos interactiva?	1
7	¿Requiere la entrada de datos interactiva que las transiciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	2
8	¿Se actualizan los archivos maestros de forma interactiva?	4
9	¿Son complejas las entradas, las salidas, los archivos o las peticiones?	3
10	¿Es complejo el procesamiento interno?	4
11	¿Se ha diseñado el código para ser reutilizable?	4
12	¿Están incluidas en el diseño la conversión y la instalación?	3
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5

Fuente: Elaboración propia

Reemplazando los valores correspondientes en la ecuación de PF dada, se tiene:

$$PF = 289 * [0.97 + (0.01 * 52)] = 430.61$$

Ahora considerando a SUM(fi) como el 100%, es decir la suma total de los 14 valores sería de 70, máximo valor de ajuste de complejidad, entonces la relación de los puntos de función máximo es:

$$PF_{\max} = 289 * [0.97 + (0.01 * 70)] = 482.63$$

Es entonces que:

$$PF/PF_{\max} = 430.61/482.63 = 0.89$$

Por lo tanto el sistema tiene una funcionalidad del 89 % que es aceptable.

5.1.3 USABILIDAD

Para conocer si el sistema satisface los requerimientos establecidos por el usuario, se realiza una evaluación del mismo en base a encuestas planteadas a los usuarios del sistema, los cuales calificaron en una ponderación del 100%, obteniendo los resultados en la tabla 4.4

Tabla 4.4: Evaluación del sistema

Pregunta	Ponderación
¿La interfaz del usuario es amigable?	95%
¿Es entendible y se comprende el manejo del sistema?	90%
¿Las operaciones y la navegación son intuitivas desde la interfaz de usuario?	85%
¿El sistema satisface todos los requerimientos?	97%
¿Los datos de salida son confiables?	100%
¿El tiempo de respuesta de la aplicación es óptimo?	96%

Fuente: Elaboración Propia

De acuerdo a los resultados anteriores y mediante la siguiente fórmula se obtiene:

$$U = \sum X_i / n = 563/6 = 93,8$$

Por lo tanto concluimos que la usabilidad del sistema es de 94%.

4.1.4 FACILIDAD DE MANTENIMIENTO

La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

La característica determinada por el diseño del sistema es usual, determina los procedimientos de mantenimiento y la duración de los tiempos de reparación de los diferentes módulos que comprende el sistema.

- **Índice de madurez del software:** El estándar IEEE 982.1-1988 nos sugiere un índice de madurez del software (IMS) que indica la estabilidad de un producto de software, determinado en la siguiente fórmula:

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

Donde: M_t es el número de módulos en la versión actual.

F_c es el número de módulos en la versión actual que han cambiado.

F_a es el número de módulos en la versión actual que se han añadido.

F_d es el número de módulos en la versión anterior que se han borrado en la versión actual.

Si obtenemos un resultado aproximado a uno, el producto comienza a estabilizarse, entonces con: $M_t = 11$, $F_c = 1$, $F_a = 0$, $F_d = 0$ y se tiene el siguiente resultado:

$$IMS = [10 - (1 + 0 + 0)] / 10 = 0.90$$

Por lo tanto el sistema de información y control para el sindicato de grúas, montacargas y equipo pesado La Paz tiene un índice de estabilidad aceptable del 90%.

4.1.5 EFICIENCIA

La eficiencia es el conjunto de características que determinan la relación entre el nivel de rendimiento del software y el número de recursos usados, bajo ciertas condiciones dadas. Se representa mediante la siguiente ecuación:

$$Eficiencia_{sist} = Eficiencia / LCC$$

Donde: $Eficiencia = disponibilidad * confiabilidad * mantenibilidad * capacidad$

$LCC = Costo de ciclo de vida$

La disponibilidad es una medida frecuente para que el sistema esté listo y en funcionamiento, produciendo resultados efectivos, para ello se considera que la disponibilidad del sistema está alrededor del 95%.

La capacidad está relacionada con la entrega productiva, lo cual es medida en términos del desempeño de actividad productiva en comparación a los datos del sistema. Así la capacidad será de un 95% y el costo del ciclo de vida (LCC) se considerará como un 80%.

Por lo tanto se reemplaza en la fórmula de eficiencia del sistema:

$$Eficiencia_{sist} = (0.95 * 0.97 * 0.90 * 0.95) / 0.80$$

$$Eficiencia_{sist} = 0.98$$

Finalmente se concluye que la eficiencia del sistema alcanza a un 98% de eficiencia.

4.1.6 PORTABILIDAD

La portabilidad es un conjunto de características que determinan la capacidad del software y/o hardware para ser transferido de un entorno de operación a otro, sin necesidad de aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito.

Trasladar el Sistema de Información y Control para el Sindicato de Grúas, Montacargas y equipo pesado La Paz de un sistema hardware y/o software no es una prioridad del proyecto, porque la empresa cuenta con éste sistema vía web.

Además los usuarios pueden acceder al sistema desde cualquier equipo y la Portabilidad del sistema se encuentra centrada en el servidor de la empresa, para lo cual se realiza el análisis siguiente.

Hardware del servidor: el entorno de hardware a nivel de servidor funciona a partir de una Pentium D con memoria RAM de 1 Gigabyte y 4 gigas de disco duro.

Software del servidor: la aplicación puede funcionar en las siguientes plataformas: Windows XP y Windows 7 que es recomendable, debido a la velocidad y bajo consumo de recursos y cuenta con mecanismos para mejorar y optimizar la gestión de memoria RAM.

Es también posible acceder al sistema mediante cualquier navegador como ser: Internet Explorer, Mozilla, Opera y otros, porque la tecnología de desarrollo procesa la petición en el servidor y responde al cliente en HTML estándar que es reconocido por cualquier navegador.

4.2 MODELO CONSTRUCTIVO DE COSTES

El modelo constructivo de costes (COCOMO) es un modelo matemático de base empírica utilizado para estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y de aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Se tiene la siguiente ecuación que describe al modelo COCOMO:

$$E = a_b (KLDC) \exp(b_b)$$

$$D = c_b (E) \exp(d_b)$$

Donde E: es el esfuerzo aplicado en personas-mes

D: es el tiempo de desarrollo en meses cronológicos

KLDC: es el número estimado de líneas de código distribuidas en miles para el proyecto a_b , b_b , c_b , d_b (coeficientes que se muestran en la tabla 4.5)

Proyecto de Software	a_b	b_b	c_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Entonces para el sistema de Información de prestación de servicios, calculando datos se tiene:

$$KLDC = 3 \text{ (miles de líneas de código)}$$

$$E = 3.0 * 3^{1.12} = 10.27 \approx 10 \text{ personas - mes}$$

$$D = 2.5 * 10.27^{0.35} = 5.65 \approx 5 \text{ meses}$$

Por lo tanto es recomendable N personas para el desarrollo:

$$N = E/D$$

$$N = 10.27 / 5.65 = 1.82$$

$$N = 2 \text{ personas}$$

En consecuencia podemos concluir que si asumimos como sueldo mínimo de un programador es de 2500 Bs. Mensuales, se tiene:

$$\text{Costo-programadores} = 5 \text{ meses} * 2 \text{ personas} * 2500 \text{ Bs.}$$

$$\text{Costo-programadores} = 25.000 \text{ Bs.}$$

El costo total del proyecto tendría un precio de Bs. 25.000.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

Finalizando el desarrollo del proyecto con el logro de que cumpla los requisitos el sistema de información y control del Sindicato de grúas, montacargas y equipo pesado La Paz que fueron planteados por los interesados, se especifican las conclusiones y recomendaciones que se tiene para el nuevo sistema.

5.1 CONCLUSIONES

Se realizaron todas las actividades propuestas al inicio del proyecto y las conclusiones a las que se llegaron son las siguientes:

Mediante los requerimientos que se obtuvieron se lograron identificar los problemas de registrar a los afiliados que son los socios del sindicato, de los cuales se propusieron soluciones factibles que permitieron resolver los problemas planteados.

También se pudieron registrar a los secretarios que son los principales organizadores de la dirección del sindicato para el posterior uso del sistema y puedan tener el control directo de todas las actividades del sindicato.

La información recabada y controlada de los vehículos, socios, aportes, infracciones, sanciones y el administrativo se realiza de manera automática.

La información que se maneja es ahora más detallada y rápida al realizar las diferentes consultas que se generan en los reportes de los vehículos, socios, aportes, sanciones e infracciones.

Se realizó una interfaz Web de fácil manejo para los usuarios y de fácil acceso a los módulos para los administrativos que deseen consultar información que se encuentra en la base de datos del sindicato.

Se construyó y desarrolló los módulos necesarios en el sistema para generar consultas, reportes, etc. según los requerimientos encontrados en el sindicato de grúas.

El sistema desarrollado es un aporte para el área de afiliados el cual logro llenar con buenas expectativas al Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz".

5.2 RECOMENDACIONES

Terminando de denotar las conclusiones del proyecto, se muestran las siguientes recomendaciones:

Se recomienda habilitar un módulo de consultas para clientes de cada empresa afiliada al Sindicato, que contenga información histórica de los servicios realizados por cada empresa.

Para brindar mayor información se recomienda implementar una conexión entre las empresas afiliadas que contengan una página Web y el Sistema de Información y Control del Sindicato de Grúas, Montacargas y Equipo Pesado "La Paz", para una mejor comunicación entre empresas.

Se recomienda realizar el mantenimiento de software del sistema y también de la base de datos, depurando y actualizando de manera periódica para su buen funcionamiento.

Es recomendable el cambio constante de claves de acceso al sistema, realizar constantemente copias de seguridad de los registros que se realizaran en la base de datos.

También es necesario e importante implementar un módulo de vialidad y recorrido de cada vehículo que se encuentra afiliado, con el fin de contar con la información gráfica de las calles y avenidas de los recorridos por los vehículos, para facilitar de gran manera las futuras planificaciones de circulación vehicular.