

43



**UNIVERSIDAD NACIONAL AUTONOMA  
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
CAMPUS ARAGÓN**

**“ SISTEMA AUTOMATIZADO DE  
INFORMACIÓN PARA LA GESTIÓN DEL  
TRANSPORTE PÚBLICO EN LA DGT”**

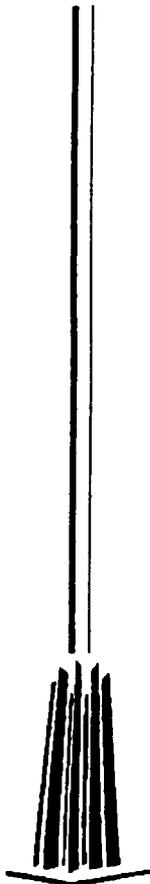
**T E S I S**  
QUE PARA OBTENER EL TITULO DE:  
**INGENIERO EN COMPUTACION**  
**P R E S E N T A:**  
**LLUVIA ANGÉLICA NIETO HERNÁNDEZ**

**ASESOR DE TESIS: ING. SILVIA VEGA MUYTOY**

**MÉXICO**

**TESIS CON  
FALLA DE ORIGEN**

**2002**





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# PAGINACION DISCONTINUA

---

# **AGRADECIMIENTOS**

A la Universidad Nacional Autónoma de México,  
por permitirme ser parte de ella y brindarme la oportunidad de superarme.

A mi mamá  
por ser parte fundamental de mi vida y por no dejarme ser mediocre.

Al Dr. Alejandro Antonio Ayala  
por estar siempre conmigo y darme confianza.

A mi asesora Ing. Silvia Vega Muytoy  
por su apoyo y paciencia en la realización de este nuevo reto en mi vida.

A mi revisores:  
Ing. Gabriela González Hernández,  
Mat. Luis Ramírez Flores,  
Ing. Marcelo Pérez Medel,  
Ing. Martín Ordóñez Rosales,  
Ing. Ernesto Ramírez Montalvo  
por su colaboración y apoyo en la realización de este trabajo.

A mis compañeros y amigos:  
Adriana, Arturo, Charo, Erick, Flor, Gerardo, Juan, Norma, Roberto, Roy,

**Gracias por brindarme su apoyo.**

---

TESIS CON  
FALLA DE ORIGEN

---

# ÍNDICE

TEMA	PÁGINA
Introducción .....	I
CAPITULO I. ANTECEDENTES .....	1
I.1 Importancia de la Información .....	2
I.2 Sistema de Información .....	3
I.2.1 Definición .....	3
I.2.2 Método Tradicional para el Desarrollo de Sistemas .....	5
I.2.3 Responsabilidades durante el Desarrollo de un Sistema ....	6
I.2.4 Calidad en el Desarrollo de un Sistema .....	7
I.2.5 ¿A qué se debe el atraso durante el desarrollo de un sistema de información?.....	9
I.3 Base de Datos .....	10
I.3.1 Definición .....	10
I.3.2 Ventajas y Desventajas de las Bases de Datos .....	10
I.3.3 Modelos de Bases de Datos .....	11
I.3.3.1 Modelo Relacional .....	12
I.3.3.1.1 Antecedentes .....	12
I.3.3.1.2 Características .....	15
I.3.3.1.3 Conceptos del Modelo Relacional.....	18
I.3.4 Administración de la Base de Datos .....	19
I.3.5 Bases de Datos Centralizadas .....	19
CAPITULO II. PLANEACIÓN Y ANÁLISIS .....	21
II.1 Planeación del Sistema .....	22
II.1.1 Perspectiva de la Dependencia .....	23
II.1.1.1 Marco de Referencia .....	23
II.1.1.2 Problemática Actual .....	24
II.1.1.3 Objetivo General .....	24
II.1.1.4 Objetivos Particulares .....	24
II.1.1.5 Beneficios Esperados .....	24
II.1.2 Arquitectura de la Información .....	25
II.1.2.1 Proceso de Desarrollo Análisis y Diseño Orientado a Objetos.....	25
II.1.2.2 Arquitectura propuesta del Sistema. Arquitectura de 3 capas ó DNA (Aplicaciones Distribuidas de Red).....	27
II.1.2.2.1 Capa de Presentación .....	28
II.1.2.2.2 Capa de Negocios .....	28
II.1.2.2.3 Capa de Datos .....	29
II.1.2.3 Documentación UML (Unified Modeling Language ) ..	29

---

II.1.3 Análisis de Áreas de la Empresa.....	31
II.2 Análisis del Sistema .....	33
II.2.1 Estudio de Factibilidad .....	35
II.2.1.1 Factibilidad Operacional .....	35
II.2.1.2 Factibilidad Técnica .....	35
II.2.1.3 Factibilidad Económica-Financiera.....	36
II.2.2 Calendario de Actividades .....	36
II.2.3 Requerimientos de Información .....	37
II.2.3.1 Panorama General .....	37
II.2.3.2 Clientes .....	37
II.2.3.3 Metas .....	37
II.2.3.4 Funciones del Sistema .....	38
II.2.3.5 Atributos del Sistema .....	39
II.2.4 Casos de Uso .....	39
II.2.4.1 Identificación de los Casos de Uso.....	41
II.2.4.2 Determinación de Actores.....	41
II.2.4.3 Determinación de Casos de Uso en el formato de Alto Nivel.....	42
II.2.5 Diagrama de Casos de Uso .....	47
<b>CAPITULO III. DISEÑO Y CONSTRUCCIÓN .....</b>	<b>54</b>
III.1 Diseño y Construcción del Sistema .....	55
III.1.2 Justificación del Hardware y Software propuesto para la Construcción del Sistema.....	55
III.1.2.1 Selección del Sistema Operativo .....	56
III.1.2.2 Selección del DBMS .....	57
III.1.2.3 Selección de la Herramienta de Desarrollo .....	59
III.1.2.4 ¿Por qué es importante el uso de una herramienta de desarrollo en conjunto con un DBMS ? .....	60
III.2 Diseño Conceptual .....	60
III.2.1 Modelo Conceptual .....	60
III.2.1.1 Conceptos.....	61
III.2.1.2Asociaciones .....	63
III.2.1.3Atributos .....	63
III.3 Diseño Lógico .....	66
III.3.1 Diagrama de Clases .....	68
III.3.2 Diccionario de Datos .....	71
III.3.3 Modelos de Comportamiento del Sistema .....	77
III.3.3.1 Diagramas de Secuencia del Sistema .....	77
III.3.3.2 Contratos Threads).....	80
III.4 Diseño Físico y Construcción del Sistema .....	88
III.4.1 Arquitectura Cliente/ Servidor .....	88

TESIS CON  
 FALLA DE ORIGEN

---

III.4.2 Diagramas de Secuencia .....	90
III.4.3 Ciclo de Vida de Desarrollo del Software .....	113
III.4.4 ¿Cómo se lleva a cabo la implementación de la Arquitectura de 3 Capas? .....	115
III.4.4.1 ¿Cómo se lleva a cabo la comunicación entre las capas (com) de nuestro sistema? .....	119
III.4.5 Pruebas del Sistema .....	119
III.4.6 Evaluación de la calidad del sistema.....	121
CAPITULO IV. IMPLEMENTACION Y SOPORTE .....	122
IV.1 Implementación del Sistema .....	123
IV.1.1 Instalar y probar los nuevos paquetes de software comerciales .....	123
IV.1.2 Instalar el Sistema .....	124
IV.1.3 Capacitación y Documentación a Usuarios .....	124
IV.1.4 Realizar un Plan de Conversión .....	125
IV.1.5 Evaluación del Sistema .....	126
IV.2 Soporte de Sistemas .....	126
CONCLUSIONES .....	129
BIBLIOGRAFÍA .....	130
ANEXO A Manual de Instalación	
ANEXO B Manual de Usuario	

---

TESIS CON  
FALLA DE ORIGEN

---

# INTRODUCCIÓN

Debido al gran auge tecnológico en el país, actualmente se tiene la oportunidad de contar con sistemas que faciliten y agilicen el manejo de información en una empresa o institución.

Es por ello que surge la necesidad de que la Dirección General de Transporte del Distrito Federal cuente con un sistema automatizado de información que sustituya al modelo tradicional.

La presente tesis esta enfocada al desarrollo de un sistema de información, que contribuya al mejor desempeño de la dependencia.

A continuación se hará una breve descripción sobre el contenido de esta tesis:

## Capitulo I. Antecedentes.

Se trata de un breve resumen de los aspectos que deben conocerse para entender mejor el desarrollo del sistema de información.

## Capitulo II. Planeación y Análisis.

En la etapa de Planeación, se inicia la investigación acerca de la problemática que enfrenta, los objetivos que persigue.

En la etapa de Análisis se presenta el estudio de factibilidad, así como también los requerimientos de información del sistema.

## Capitulo III. Diseño y Construcción.

Como su nombre lo indica se tratará el tema del diseño y desarrollo del sistema.

## Capitulo IV. Implementación y Soporte

Donde se describirá como se realizo la instalación y puesta en marcha, así como el mantenimiento del sistema.

Por último, se agregan dos anexos, el anexo A contiene un manual de instalación y el anexo B un manual de usuario, ambos tienen la finalidad de facilitar el manejo de este sistema.

---

TESIS CON  
FALLA DE ORIGEN

# ***CAPITULO I***

## ***ANTECEDENTES***

---

El ambiente rápidamente cambiante y el aumento de tamaño y complejidad de los sistemas ha incrementado las necesidades de información, por lo tanto hacen falta buenos sistemas para suministrar información oportuna, adecuada y concisa en todo tipo de empresa u organización.

## **1.1 IMPORTANCIA DE LA INFORMACIÓN**

Las sociedades modernas están orientadas a la tecnología y empiezan a modificar sus características a ritmo acelerado.

Algunos acontecimientos relacionados con la información repercuten de manera directa en la sociedad y en las instituciones, como:

- Mayor cultura en computación entre los líderes y la población en general.
- Los progresos en las telecomunicaciones a nivel internacional.
- La aparición, transformación y proliferación de las computadoras.
- Las computadoras en el hogar y la oficina conectadas a grandes servidores y a extensas bases de datos y entre sí.
- Los delitos por computadora y las medidas para combatirlos.

A medida que aumenta la cantidad y la velocidad con que la información se pone al servicio de los usuarios, el manejo de ella debe ser cada vez más selectivo. Los modernos sistemas de información suministran información generada internamente y en el ambiente.

La información es un signo o conjunto de signos que impulsan a la acción. Se distingue de los datos porque éstos no son estímulos de acción, sino simplemente cadenas de caracteres o patrones sin interpretar.

La toma de decisiones, conduce a la acción basándose en la información. La incertidumbre ante el futuro y la falta de conocimiento sobre la situación actual hacen que se busque información; ésta puede definirse como datos organizados que reducen la incertidumbre en el momento de tomar decisiones.

TESIS CON  
FALTA DE ORIGEN

---

## 1.2 SISTEMA DE INFORMACIÓN

Los Sistemas de Información sistematizan los procesos operativos de las empresas o instituciones, proporcionan información de apoyo al proceso de toma de decisiones y facilitan el logro de ventajas competitivas a través de su implementación en las empresas.

### 1.2.1 DEFINICIÓN

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o institución; estos elementos incluyen:

- ❖ *Equipo de cómputo*, es decir el hardware necesario para que el sistema de información pueda operar. Lo constituyen las computadoras, equipo de comunicación y periféricos que pueden conectarse a ellas.
- ❖ *Recursos Humanos* que interactúan con el sistema, el cual esta formado por las personas que utilizan el sistema, alimentándolo con datos o utilizando los resultados que genere.
- ❖ *Datos o información fuente* introducida en el sistema; son todas las entradas que necesita el sistema de información para generar como resultado la información que se desea.
- ❖ *Programas* que son procesados y producen diferentes tipos de resultados. Los programas son la parte del software del sistema de información que hará que los datos de entrada introducidos sean procesados correctamente y generen los resultados que se esperan.

Un Sistema de Información consta de cuatro actividades básicas:

- **Entrada de Información.** La entrada es el proceso mediante el cual el sistema toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que proviene o es tomada de otros sistemas o de sí mismo. Esto se denomina *interfases automáticas*.

Los dispositivos de entrada de datos a las computadoras son: el teclado, el mouse, discos flexibles, códigos de barras, escáners, monitores sensibles al tacto, etc.

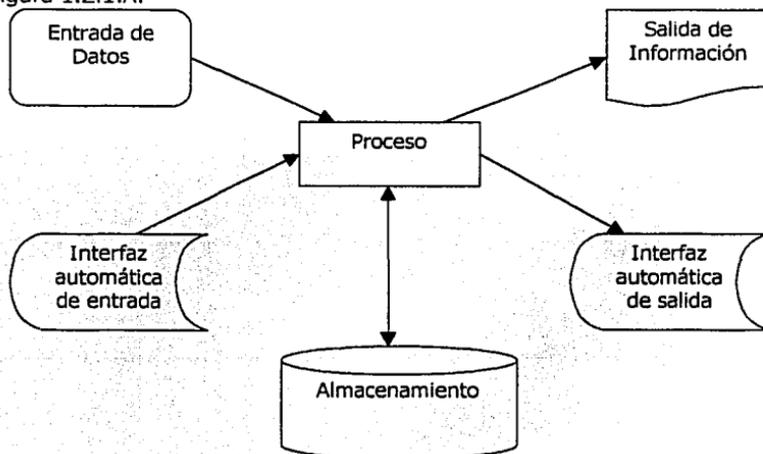
➤ **Almacenamiento de Información.** El almacenamiento es una de las actividades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar el proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

La dispositivos de almacenamiento son los discos duros, discos flexibles, cintas magnéticas y discos compactos (CD-ROM).

➤ **Procesamiento de Información.** Es la capacidad del Sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema ó bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.

➤ **Salida de Información.** Es la capacidad de un sistema para obtener la información procesada o bien datos de entrada al exterior. Los dispositivos de salida son: el monitor, impresoras, discos flexibles, graficadores, entre otros. Es importante tomar en cuenta que la salida de un sistema de información puede constituir la entrada a otro sistema de información o al mismo sistema.

Las actividades que integran un sistema de información se muestran en la figura 1.2.1.A.



TESIS CON  
FALLA DE ORIGEN

FIG 1.2.1.A ACTIVIDADES QUE INTEGRAN UN SISTEMA DE INFORMACIÓN.

Los sistemas de Información cumplen tres objetivos básicos:

- ✓ Sistematización de procesos operativos.
- ✓ Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
- ✓ Lograr ventajas competitivas a través de su implantación y uso.
- ✓ Optimización de recursos humanos, costos y tiempo.

### 1.2.2 MÉTODO TRADICIONAL PARA EL DESARROLLO DE SISTEMAS

Consiste en una serie de etapas consecutivas que inician con la planeación y terminan con el soporte del sistema. A este método se le conoce como *cascada o caída de agua* debido a que las fases son consecutivas, a pesar de que se sigue un orden en la realización de cada una de las fases, es posible regresar a la fase anterior para hacer correcciones en caso de ser necesario (figura 1.2.2.A).

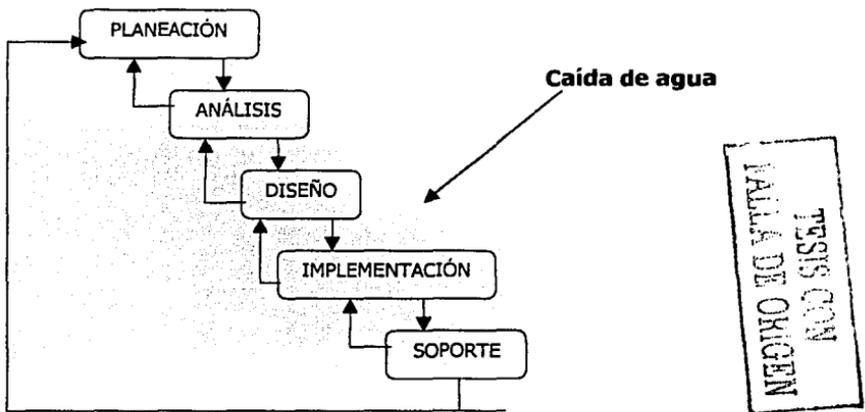


FIG 1.2.2.A MÉTODO TRADICIONAL PARA EL DESARROLLO DE SISTEMAS

- ❖ **Planeación.** Su propósito es establecer las prioridades sobre aquellas aplicaciones cuyo desarrollo reporte máximos beneficios para la empresa. Se identifica la perspectiva de la dependencia, marco de referencia,

problemática actual, objetivos generales y particulares, beneficios esperados.

- ❖ **Análisis.** Se estudia la situación actual de la empresa y se definen sus necesidades de información. Se elabora un estudio de factibilidad, para saber que tan favorable es la creación del sistema.
- ❖ **Diseño.** Especificación detallada de la solución de tipo informático que satisfaga las necesidades planteadas en el análisis. Diseñar e integrar el nuevo sistema.
- ❖ **Implementación.** Se integran los componentes técnicos, se lleva a cabo la instalación, plan de conversión, evaluación y documentación del sistema, así como la capacitación del personal.
- ❖ **Soporte.** Se encarga del mantenimiento y la mejora permanente del sistema para adaptarlo a nuevas necesidades, así como brindar apoyo a los usuarios.

En el presente trabajo, se profundiza en el estudio de las fases antes mencionadas.

### 1.2.3 RESPONSABILIDADES DURANTE EL DESARROLLO DE UN SISTEMA

Tanto usuarios, gerencia y desarrolladores interactúan de diversas formas durante el desarrollo de un sistema de información. En la tabla 1.2.3.A se muestran las actividades sugeridas para cada uno de ellos.

Entiéndase al desarrollador como líder de proyecto, analista, diseñador, programador.

Etapa	Usuarios	Gerencia	Desarrollador
Planeación	Sugieren aplicaciones, especifican necesidades de información.	Aprueba el área de aplicación y establece objetivos.	Escucha las necesidades, propone opciones, prepara encuesta preliminar.

TESIS CON  
FALLA DE ORIGEN

Etapa	Usuarios	Gerencia	Desarrollador
Análisis	Ayudan a describir el sistema existente. Recolectan y analizan datos.	Revisa factibilidad, proporciona recursos, lleva a cabo revisiones.	Lleva a cabo el análisis, recolección de datos, documenta hallazgos.
Diseño	Diseñan la entrada, salida y lógica de proceso.	Revisa el diseño de los usuarios.	Presenta opciones a los usuarios en relación con sus decisiones.
Programación		Proporciona recursos extras si son necesarios	Organiza la programación, diseña módulos, codifica programas.
Pruebas	Generan datos de prueba y evalúan resultados.	Revisa los datos de prueba, hace correcciones y evalúa resultados.	Prueba los módulos de programa individualmente, así como el sistema completo.
Instalación	Realizan una revisión posterior a la instalación junto con los desarrolladores y reportan resultados.	Controla recursos y verifica que los resultados obtenidos, sean los esperados.	Coordina y lleva a cabo los procedimientos de instalación, como creación de archivos iniciales para el sistema, etc.
Capacitación	Asisten a sesiones de entrenamiento, aprenden el uso del sistema.	Revisa manuales, acepta documentación.	Realización de materiales de apoyo y entrena al grupo operativo.
Soporte	Proporcionan datos y utilizan la salida, sugieren modificaciones y mejoras.	Controla recursos, analiza beneficios.	Procesa los datos para producir una salida confiable, responde a las peticiones de mejoramiento, sugiere mejoras y controla el servicio.

TABLA 1.2.3.A RESPONSABILIDADES EN EL DESARROLLO DE UN SISTEMA

### 1.2.4 CALIDAD EN EL DESARROLLO DE UN SISTEMA.

El desarrollo de sistemas involucra muchos riesgos, sobre todo en las fases iniciales, en las cuales debe quedar bien definido, es por ello que los desarrolladores deben asegurar desde estas etapas la *calidad del sistema*.

Existen dos conceptos relacionados con el proceso de aseguramiento de la calidad durante el desarrollo del sistema:

- ⊙ El usuario del producto desarrollado es el factor más importante en el establecimiento y evaluación de la calidad, es decir, el usuario es quien determina si el sistema satisface sus requerimientos.
- ⊙ Es mucho menos costoso corregir un problema de calidad en sus primeras etapas antes de que el problema se convierta en quejas y genere desconfianza en el usuario.

En la figura 1.2.4.A, se relaciona la naturaleza del error, es decir la fase en que se generó el error contra la etapa en que se detectó el error. El menor costo está indicado en el modelo óptimo, seguido del 1, 2...5, donde el 5 es el costo mayor.

Si el error es detectado rápidamente, será menos costoso corregirlo; aunque lo ideal sería que los errores se detecten oportunamente, en la fase en que se generaron.

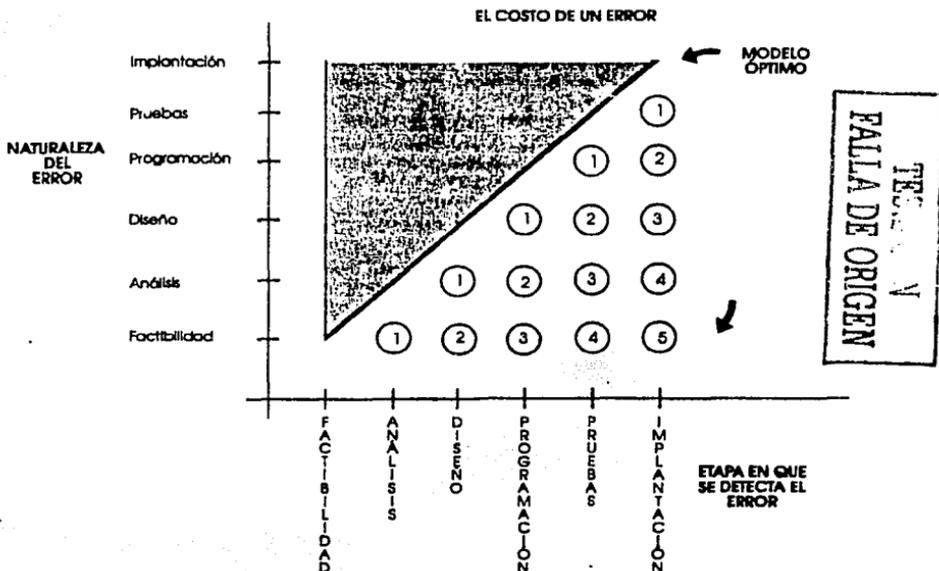


Fig 1.2.4.A COSTO DE LOS ERRORES EN EL DESARROLLO DE SISTEMAS.

Se pueden llevar a cabo revisiones rutinarias, con el fin de monitorear todo el proceso, detectar problemas y considerar soluciones para su corrección, con el fin de evaluar el sistema conforme se va desarrollando y no esperar a que se concluya para determinar su calidad.

### 1.2.5 ¿A QUE SE DEBE EL ATRASO DURANTE EL DESARROLLO DE UN SISTEMA DE INFORMACIÓN ?

Existen diferentes enfoques, desde el punto de vista de:

#### PROCESO

- Ausencia de control de calidad.
- Mala planeación
- Abandono de método o planeación bajo presión.
- Eliminación de la fase de análisis o diseño.
- Control directivo insuficiente.
- Omitir tareas necesarias en la planificación.

#### PRODUCTO

- Exceso de requerimientos.
- Desarrolladores meticulosos.
- Desarrollo orientado a la investigación.

#### TECNOLOGÍA

- Falta de control de código
- Cambio de herramientas a medio desarrollo.

Para complementar lo anterior, se analiza la tabla comparativa 1.2.5.A entre el Ayer y Hoy de los Sistemas de información.

AYER	HOY
Datos y procesamiento separados	Datos y procesos integrados: Encapsulado.
Código redundante	Código NO redundante (en un sólo código se tienen 3 invocaciones).
Alta dependencia de datos	Independencia de datos (lógica encapsulada).
Alta dependencia tecnológica	Mayor independencia tecnológica.
Difícil mantenimiento	Fácil mantenimiento.

TABLA 1.2.5.A AYER Y HOY DE LOS SISTEMAS DE INFORMACIÓN

---

## 1.3 BASES DE DATOS

El uso de sistemas de información por parte de las organizaciones requiere el almacenamiento de grandes cantidades de información, ya sea para el uso mismo del sistema, para generar resultados o para compartir dicha información con otros sistemas. Es por ello, que actualmente se están utilizando las bases de datos para satisfacer estos requerimientos.

### 1.3.1 DEFINICIÓN

Una base de datos es una colección de datos interrelacionados que pueden ser procesados por uno o más sistemas de aplicación. Un sistema de base de datos consta de al menos una base de datos, un *software* de propósito general, llamado sistema administrador de bases de datos (DBMS), que manipula la base de datos, así como por el *hardware* y el personal apropiados.

Los datos se integran en un depósito común que permite evitar la duplicidad y redundancias y da lugar a una mejor utilización de los recursos informáticos, además de simplificar la inserción, la eliminación y la modificación de los registros.

En la base de datos (BD), cada usuario tiene una vista de aquellos datos que son necesarios para la aplicación que esta ejecutando, siendo el administrador de la BD el único que tiene acceso a todos los datos.

### 1.3.2 VENTAJAS Y DESVENTAJAS DE LAS BASES DE DATOS

Las ventajas para los datos:

- ✓ Mayor Disponibilidad (Los datos son compartidos entre los diferentes usuarios y aplicaciones, siempre y cuando estén autorizados).
- ✓ Eficiencia en la Entrada de los Datos al Sistema (Al no existir redundancias, se captura sólo una vez).

Las ventajas para los Resultados:

- ✓ Coherencia (Los datos son almacenados una sola vez. La información es utilizada en muchos procesos, por lo tanto debe ser la misma).
- ✓ Mejor Documentación de la Base de Datos.

- 
- ✓ Mayor Valor Informativo (Una base es un conjunto de elementos interrelacionados, su valor informático es mayor que cada uno de los elementos que lo integran).

Las ventajas para los usuarios:

- ✓ Flexibilidad (Proporcionada por la independencia de datos, porque se puede adaptar el sistema según los nuevos requerimientos de la empresa a bajo costo).
- ✓ Facilidad de compartir datos entre usuarios.

La aplicación de la metodología de las bases de datos ayuda a obtener estructuras de información coherentes, que proporcionan el debido soporte para la toma de decisiones, permitiendo un mejor rendimiento de los datos existentes así como del equipo empleado.

Entre las principales desventajas tenemos:

- × Implementación costosa: si se toma en cuenta el equipo físico (hardware e instalaciones adecuadas), como el lógico (software).
- × Se requiere de personal especializado.
- × Falta de rentabilidad a corto plazo, ya que sólo se aprecian resultados después de iniciado el proyecto, es decir después de varios meses de trabajo.

### **1.3.3 MODELOS DE BASES DE DATOS**

Existen fundamentalmente cuatro alternativas disponibles para diseñar las bases de datos: el modelo jerárquico, de red, el relacional y el orientado a objetos.

Las bases de datos jerárquicas y de red son dependientes de los datos que procesan y si se modifican, se tiene que modificar también la aplicación; en cambio, las bases de datos relacionales ofrecen independencia de datos, es decir es posible hacer cambios a la información que contiene la base o tener acceso a ésta, sin necesidad de modificar la aplicación.

Debido a que los modelos jerárquico y de red, han caído en desuso por las desventajas que presentan, la tendencia de las compañías actualmente ha sido

---

migrar a bases de datos relacionales, por lo tanto se profundizará en el modelo Relacional.

### 1.3.3.1 MODELO RELACIONAL

#### 1.3.3.1.1 Antecedentes

La utilización de punteros físicos suponía ventajas y desventajas para los sistemas de bases de datos jerárquicos y en red.

La ventaja estaba en que los punteros permitían la recuperación rápida de los datos que tuvieran interrelaciones predeterminadas.

La desventaja estaba en el hecho de que estas interrelaciones tenían que definirse antes de que el sistema fuera puesto en marcha. Era difícil recuperar datos basados en otras interrelaciones. En la medida en que los usuarios se familiarizaron con los sistemas de base de datos y con su potencia para manipular los datos rápidamente, encontraron estas limitaciones inaceptables.

En 1970, E. F. Codd publicó un artículo<sup>1</sup> revolucionario que cambió la filosofía de las bases de datos. Codd argumentó que los datos deberían relacionarse mediante interrelaciones naturales, lógicas, inherentes a los datos, y no mediante punteros físicos. Es decir, debía existir la posibilidad de combinar los datos de fuentes diferentes, siempre que la información lógica necesaria para efectuar dicha combinación estuviera presente en los datos. Esto abrió una nueva perspectiva para los sistemas de administración de información, ya que las consultas a las bases de datos no necesitarían, en adelante, limitarse a las interrelaciones indicadas por los punteros físicos.

Codd propuso un modelo simple de datos en el que todos ellos se representarían en tablas constituidas por filas y columnas. A estas tablas se les dio el nombre matemático de *relaciones*, y por eso el modelo se denominó *modelo relacional*.

Codd también propuso dos lenguajes relacionales para manipular los datos en las tablas: *el álgebra relacional* (es un lenguaje procedimental, ya que cuando se escribe una expresión en álgebra relacional, se proporciona una secuencia de operaciones que generan la respuesta a la consulta) y *el cálculo relacional* (es un lenguaje no procedimental donde se da una descripción formal de la información deseada sin especificar como obtenerla). Ambos lenguajes soportan la manipulación de los datos sobre la base de operadores lógicos en lugar de los punteros físicos utilizados en los modelos jerárquico y en red.

---

<sup>1</sup> "A relational model for large shared data banks" Communications of the ACM. Vol 13, Nº 6, pp. 377 - 387. Junio 1970.

---

Al manipular los datos sobre una base conceptual en vez de una base física, Codd introdujo otra innovación en los sistemas de base de datos relacionales: los archivos completos de datos se pueden procesar con instrucciones sencillas; mientras que los sistemas tradicionales requieren que los datos se procesen de registro en registro. El enfoque de Codd mejoró enormemente la eficiencia conceptual de la programación de la base de datos.

La manipulación lógica de los datos también hace factible la creación de lenguajes de interrogación más accesibles al usuario no especialista en computación. Aunque es bastante difícil crear un lenguaje que pueda ser utilizado por todas las personas sin considerar su experiencia previa en computación, los lenguajes relacionales de consulta hacen posible el acceso a las bases de datos para un grupo de usuarios cada vez mayor.

La publicación de los artículos de Codd, a principios de los años setenta, provocó una conmoción en la actividad de las comunidades de desarrollo de sistemas de investigación y de sistemas comerciales, en la medida en que trabajaban para producir un sistema administrador de bases de datos relacional.

El resultado fue la aparición de sistemas relacionales durante la última mitad de los setenta que soportaban lenguajes como el Structured Query Language (SQL), el Query Language (QUEL) y el Query-by-Example (QBE).

El 14 de Octubre y 21 de Octubre de 1985, Codd publicó un artículo en dos partes en la revista *Computerworld*, en donde menciona 12 reglas para determinar si un DBMS es relacional. Estas reglas proveen un patrón útil para la evaluación de sistemas relacionales:

**Regla 1: La regla de la información.**

Toda la información en una base de datos relacional es representada explícitamente a nivel lógico en exactamente una manera, a través de valores en las tablas.

**Regla 2: Regla de garantía de acceso.**

Todos y cada uno de los valores atómicos (datos) en una base de datos relacional es garantizado para ser lógicamente accesible por una combinación de nombres de la tabla, valor de la clave primaria y nombre de la columna.

**Regla 3: Tratamiento sistemático de valores nulos.**

Los valores nulos (no se refiere a caracteres vacíos, string vacíos, valores con cero o cualquier otro número) son soportados en un DBMS completamente relacional, para representar información faltante de una forma sistemática, independencia de tipo de datos.

---

**Regla 4: Catálogo dinámico en línea basado en el modelo relacional.**

La descripción de la base de datos es representada en un nivel lógico, de la misma manera que los datos comunes; así, los usuarios autorizados pueden aplicar el mismo lenguaje relacional para consultar esta descripción, tal como lo hacen con los datos regulares.

**Regla 5: Regla para un sublenguaje de datos.**

Un sistema relacional puede soportar varios lenguajes y modelos de uso terminal (por ejemplo, el modo llenar-los-blancos). Sin embargo, debe existir al menos un lenguaje con sentencias que sean expresables, por alguna sintaxis bien definida, como strings de caracteres y cuya habilidad para soportar todos los siguientes ítems sea comprensivo: definición de datos, definición de vista, manipulación de datos (interactivo y por programas), restricciones de integridad y límites de transacciones (begin, commit y rollback).

**Regla 6: Regla de actualización de vistas.**

Todas las vistas que son teóricamente actualizables, son también actualizables por el sistema.

**Regla 7: Inserción, actualización y eliminación de alto nivel.**

La capacidad de manejo de una relación base o derivada como un único operando, se aplica no sólo para la recuperación de los datos, sino también para la inserción, actualización y eliminación de los datos.

**Regla 8: Independencia física de los datos.**

Los programas de aplicación y actividades terminales permanecen lógicamente intactos, cuando se hace cualquier cambio, ya sea en la representación del almacenamiento o método de acceso.

**Regla 9: Independencia lógica de los datos.**

Se pueden alterar las tablas bases sin alterar los programas de aplicación o las actividades terminales.

**Regla 10: Independencia de integridad.**

Restricciones de integridad específicas para una base de datos relacional particular, deben ser definibles en el sublenguaje de datos relacional y almacenables en el catálogo, no en los programas de aplicación. Debe soportar, como mínimo, estas dos restricciones de integridad:

**1. Integridad de entidades:** No se permiten que componentes de una clave primaria tengan valores nulos.

---

---

**2. Integridad referencial:** Para cada valor distinto y no nulo de la clave foránea en una base de datos relacional, debe coincidir con un valor de una clave primaria en el mismo dominio.

**Regla 11: Independencia de la distribución.**

Un DBMS relacional tiene independencia de la distribución. Esto implica que los usuarios no tienen que estar en conocimiento que una base de datos está distribuida.

**Regla 12: Regla de no subversión.**

Si un sistema relacional tiene un lenguaje de bajo nivel (con capacidad de manejo de un único registro a la vez), ese lenguaje de bajo nivel no puede ser usado para pasar a llevar o no tomar en cuenta las reglas de integridad o las restricciones expresadas en el lenguaje relacional de alto nivel (con capacidad de manejo de múltiples registros a la vez).

Existe una regla agregada a estas 12 reglas, conocida como la **Regla Cero**: "Para cualquier sistema que asegure ser un sistema administrador de bases de datos relacional, ese sistema debe ser capaz de administrar los datos completamente, a través de sus capacidades relacionales".

Codd señala, que de acuerdo con estas reglas, aún no hay disponible un sistema totalmente relacional. En particular por las reglas 6, 9, 10, 11 y 12 que son difíciles de cumplir.

En 1986, el SQL se adoptó como la norma ANSI para los lenguajes relacionales de bases de datos. Esta norma se actualizó en 1989 y en 1992.

A medida que las computadoras se hicieron populares durante los años ochenta, los sistemas relacionales también estuvieron disponibles para ellas.

### 1.3.3.1.2 Características

El modelo relacional es un modelo muy simple con fundamentos matemáticos, basado en la teoría de conjuntos, es por ello que es fácil de manejar por usuarios poco especializados.

Ventajas:

- ✓ Independencia física y lógica.
- ✓ Eliminación de redundancias.
- ✓ Flexibilidad.

- 
- ✓ Uniformidad.
  - ✓ Sencillez.
  - ✓ Sólido fundamento teórico.

Desventajas:

- × Dificultades de instrumentación inicialmente.
- × Escaso rendimiento en sus primeras versiones.

La diferencia de una base de datos respecto a otro sistema de almacenamiento, es que en la base se almacenan los datos de forma que cumplan tres requisitos básicos:

- ❖ *No redundancia.* Los datos se almacenan una sola vez. Si varias aplicaciones necesitan los mismos datos, no crearán cada una su propia copia sino que todas accederán a la misma.
- ❖ *Independencia.* Los datos se almacenan teniendo en cuenta la estructura inherente a los propios datos y no la de la aplicación que los crea. Esta forma de trabajar es la que permite que varias aplicaciones puedan utilizar los mismos datos.
- ❖ *Concurrencia.* Varios usuarios, ejecutando la misma o diferente aplicación, podrán acceder simultáneamente a los datos.

En la figura 1.3.3.1.2.A se observa el modelo relacional dentro del marco definido por la arquitectura ANSI, mostrando además ciertas sentencias de lenguaje SQL que ayudan a definir los elementos correspondientes en los tres niveles. Las sentencias a nivel interno varían de un producto a otro y no están estandarizadas.

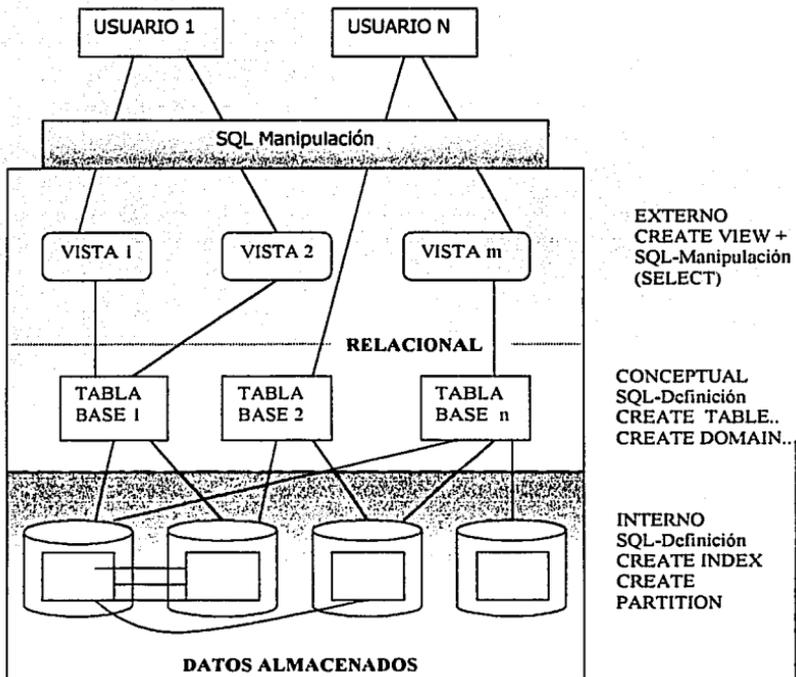


FIG 1.3.3.1.2.A ARQUITECTURA A TRES NIVELES EN UN SGBDR

FALLA DE ORIGEN  
 TRADUCCIÓN

En cuanto el diseño de bases de datos relacionales se puede decir que existen dos formas:

La primera, asume la creación de un modelo de datos conceptual durante el diseño conceptual. Este modelo es transformado a un modelo entidad-relación.



FIG 1.3.3.1.2.B DISEÑO DE UNA BASE DE DATOS RELACIONAL

En el segundo enfoque el diseño conceptual no parte del modelo de datos conceptual, sino que procede directamente a la creación de un esquema de bases de datos relacional definiendo las tablas relacionales a partir de la observación directa del mundo real. El diseño se completa mediante la normalización de estas definiciones de tablas de acuerdo a un proceso bien definido.



FIG 1.3.3.1.2.C DISEÑO DIRECTO DE UNA BASE DE DATOS RELACIONAL.

El segundo enfoque fue usado tradicionalmente antes que el modelo conceptual fuera conocido. Todavía se utiliza en situaciones que requieren un esquema de bases de datos simple.

El primer enfoque, usando modelos conceptuales, es válido en el diseño de grandes y complejos esquemas de bases de datos necesarios para los sistemas de bases de datos corporativas.

### 1.3.3.1.3 Conceptos del Modelo Relacional

**Entidad.** Es un objeto concreto o abstracto que va a ser representado en un sistema de bases de datos.

**Relación.** Es un subconjunto del producto cartesiano de  $n$  dominios no necesariamente distintos, caracterizado por un nombre. Se representa mediante una tabla o matriz de dos dimensiones, consistente en filas y columnas de datos.

**Atributo.** Es una columna de una relación o tabla, que representa una propiedad de la misma que está caracterizada por un nombre.

**Tupla.** Es cada una de las filas de una relación o tabla, y representa una ocurrencia del tipo o información que se almacena en la relación. No hay un orden preestablecido de las filas o tuplas de la relación y dos tuplas no pueden contener idénticos conjuntos de valores.

NOMBRE	APELLIDO	CLAVE	TOTAL (\$)	MES
Alejandro	Correa	12A	12456	junio
Angélica	Hernández	56G	45000	octubre
Sofía	Ortiz	34W	17000	noviembre

FIG 1.3.3.1.3.A ELEMENTOS DE UNA RELACIÓN

### 1.3.4 ADMINISTRACIÓN DE LA BASE DE DATOS

La gestión y control de la base de datos, es una función cuya responsabilidad recae en una persona o grupo de personas que recibe el nombre de *Administrador de la Base de Datos* (DBA, DataBase Administrator), el cual coordina:

- ◆ Los medios para asegurar la integridad de los valores de los datos y las medidas a tomar para que el rendimiento del sistema sea satisfactorio.

### 1.3.5 BASES DE DATOS CENTRALIZADAS

TESIS CON  
FALLA DE ORIGEN

Es una base de datos que está físicamente situada en un único lugar, controlada por una sola computadora. En principio parece obvio que la mayoría de las funciones de una base de datos se realizan más fácilmente si la base de datos está centralizada; es decir, es más fácil de actualizar, hacer copias de seguridad, consultar y controlar el acceso a la base de datos si se sabe exactamente dónde está y cuál es el software que controla la base de datos.

El tamaño de las bases de datos y la computadora en que éstas son almacenadas son independientes del lugar donde la base de datos está centralmente localizada.

Una compañía pequeña con su base de datos sobre una computadora personal tiene una base de datos centralizada, de una forma similar a una gran compañía con muchos puestos de trabajo, pero cuya base de datos está completamente controlada por un servidor.

El desarrollo de sistemas de información eficaces requiere de una administración adecuada, que garantice una orientación acorde con los objetivos y estrategias de la empresa.

# ***CAPITULO II***

## ***PLANEACIÓN Y ANÁLISIS***

Muchos de los proyectos de desarrollo de sistemas, se inician sólo como respuesta a solicitudes de los usuarios, éstos proponen proyectos que se llevan a cabo, sin tener en cuenta su impacto global sobre la empresa.

En esto radica la importancia de la planeación, en evitar que los sistemas sólo sean creados de manera aleatoria.

Además es necesario tomar en cuenta que muchos de los programadores no realizan un análisis muy profundo sobre el tema a resolver. Con frecuencia escriben el programa desde el principio y el código necesario lo agregan conforme se requiere, lo cual en la actualidad es inapropiado en los negocios de alto riesgo.

## **II.1 PLANEACIÓN DEL SISTEMA**

La **planeación de sistemas** pretende señalar y establecer prioridades sobre aquellas tecnologías y aplicaciones que producirán un beneficio máximo para la empresa.

Inicia a partir del acuerdo entre directivos de la empresa y los responsables del desarrollo del sistema, para que pueda llegarse a un acuerdo sobre el sistema idóneo a sus necesidades.

Se puede decir que la planeación se enfoca en el conocimiento y funcionamiento de la empresa.

En la figura II.1.A se pueden observar las fases de planeación de un sistema.

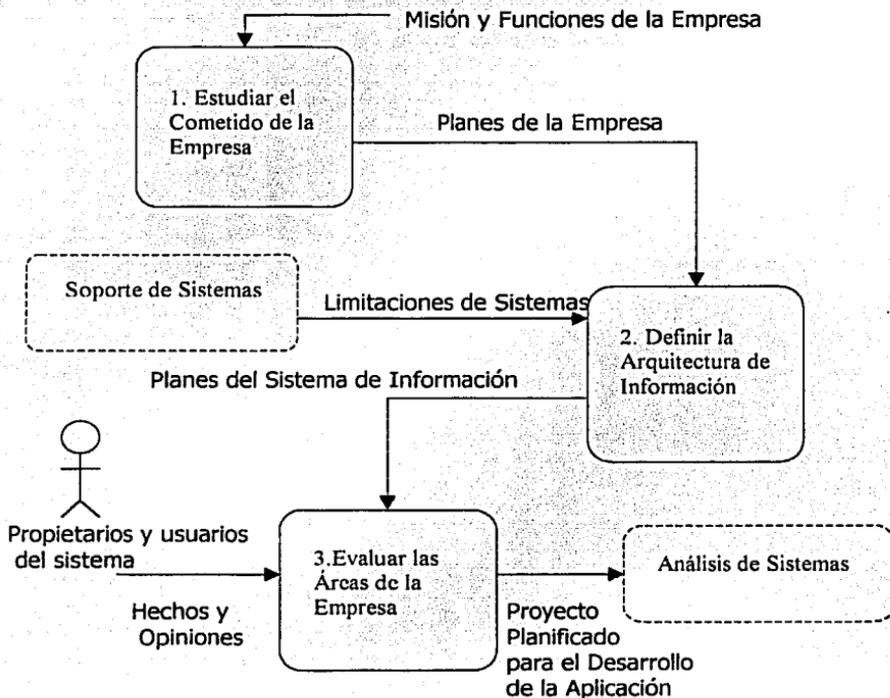


FIG II.1.A FASES DE PLANEACIÓN DE SISTEMAS.

## II.1.1 PERSPECTIVA DE LA DEPENDENCIA

TESIS CON  
FALLA DE ORIGEN

### II.1.1.1 Marco de Referencia.

En el ámbito de la Secretaria de Transporte y Vialidad del Gobierno del Distrito Federal, específicamente en la Dirección General de Transporte, los problemas relacionados al sistema de transporte público han aumentado conforme al crecimiento de la ciudad, por lo que existen necesidades que justifican el desarrollo de un SISTEMA AUTOMATIZADO DE INFORMACIÓN PARA LA GESTIÓN DEL TRANSPORTE PÚBLICO EN LA DGT.

### **II.1.1.2 Problemática Actual**

- ✕ En la DGT se maneja gran cantidad de información de manera manual, por medio de oficios y otro tipo de documentación, el principal problema es que no se tiene un control sobre la información, no esta actualizada, además cuando se hacen cambios de directivos, se pierde mucho tiempo en conocer la estructura de información de la directiva pasada; por lo tanto no es posible tomar decisiones óptimas.

### **II.1.1.3 Objetivo General**

- ⊙ Desarrollar un sistema automatizado en la DGT, que permita el manejo de gran cantidad de información de manera rápida y eficiente, como apoyo a la toma de decisiones, así como para tener un mayor control sobre dicha información y así brindar un mejor servicio al público.

### **II.1.1.4 Objetivos Particulares**

- ⊖ Mantener actualizada la información de Transporte Colectivo, teniendo así un mayor control sobre las rutas, documentar quejas y apercibimientos (llamadas de atención), con la finalidad de mejorar el servicio a los usuarios.
- ⊖ Realizar consultas y emitir reportes, acerca de información específica, cuya finalidad sea el ayudar a tomar acciones correctivas respecto al servicio que se otorga a la ciudadanía.
- ⊖ El sistema deberá ser usado por múltiples usuarios a través de una Intranet, es decir se compartirá y difundirá información a través de una red interna.

### **II.1.1.5 Beneficios Esperados**

- ✓ Mejorar la operatividad de la dependencia, dotándole de información oportuna y confiable.
- ✓ Manejo oportuno de sanciones, reincidencias y otros aspectos legales, así como la localización de áreas clave de preocupación para el usuario, gracias a la elaboración de reportes confiables.

- ✓ Será usado por múltiples usuarios a través de una intranet, permitiendo el manejo de datos globales, parciales y específicos, para la elaboración de reportes de diversa índole.
- ✓ Disminuir el papeleo excesivo y posibles irregularidades que se presenten en el manejo de la información.
- ✓ Localización y acceso a la información en menos tiempo y de manera confiable.

## II.1.2 ARQUITECTURA DE LA INFORMACIÓN

Es un plan de selección de la tecnología de información para elegir la opción que más convenga al desarrollo del sistema. Para esta fase es importante tomar en cuenta los objetivos de la empresa.

Se considera importante tomar en cuenta para este sistema, los siguientes estándares de desarrollo:

### II.1.2.1 Proceso de Desarrollo

#### ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS.

En el **análisis orientado a objetos** se identifica y describe conceptos dentro del dominio del problema.

En el **diseño orientado a objetos** se definen los componentes lógicos del software que serán implementados en un lenguaje de programación.

Es importante mencionar la siguiente comparación entre el Análisis y Diseño Orientado a Objetos y el Análisis y Diseño Estructurado como fundamento de esta propuesta.

Anteriormente el método usual era el de Análisis y Diseño Estructurado, donde se descomponía el problema en funciones o procesos, lo cual originaba una división jerárquica de procesos formados por subprocesos. El problema se presentaba al realizar sistemas complejos o muy extensos.

Actualmente el Análisis y Diseño Orientado a Objetos busca descomponer el problema por objetos y no por funciones, es decir dividirlo en unidades

maneables e independientes unas de otras; para que quede más claro, se muestra el siguiente esquema:

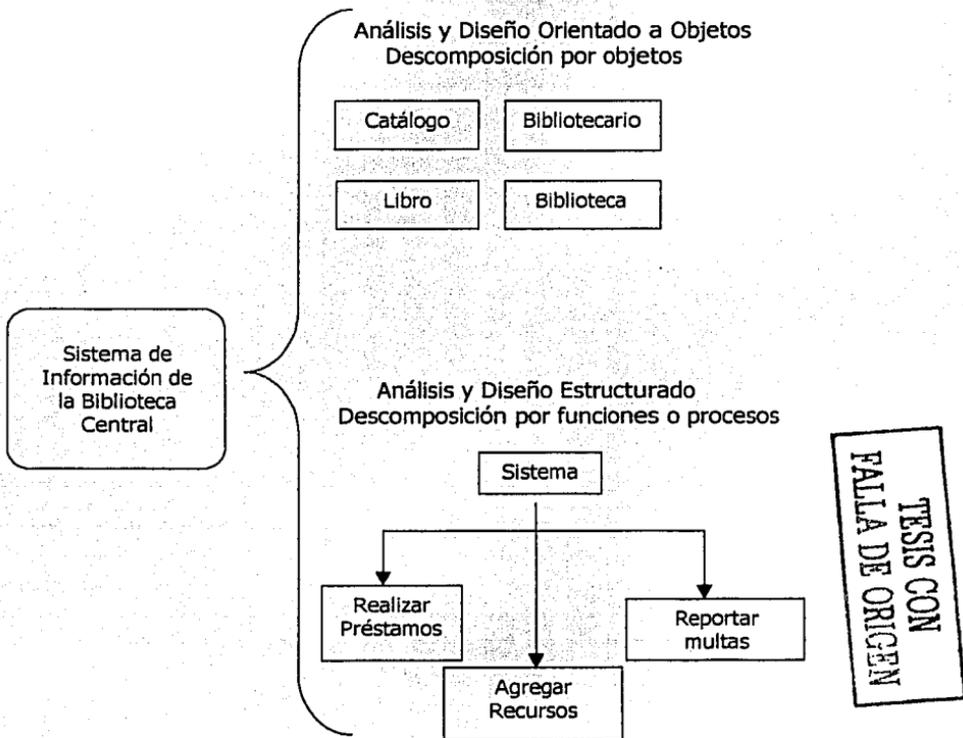


FIG II.1.2.1.A COMPARACIÓN ENTRE EL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS Y EL ANÁLISIS Y DISEÑO ESTRUCTURADO.

Al tener esta perspectiva, sobresale la metodología orientada a objetos por la facilidad de manejo en la información, además que permite la creación de sistemas robustos y de fácil mantenimiento ya que todo esta organizado y bien definido y se tiene una mejor estructura de trabajo.

### II.1.2.2 Arquitectura propuesta del sistema

#### ARQUITECTURA DE 3 CAPAS o DNA ( APLICACIONES DISTRIBUIDAS DE RED).

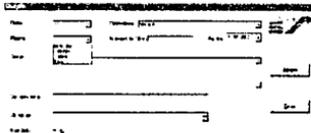
La arquitectura de aplicaciones de tres capas esta basada en componentes. Su función es unificar las aplicaciones para PC, aplicaciones cliente / servidor y aplicaciones basadas en la Web, lo cual es posible para aplicaciones de cualquier tamaño.

El objetivo de implementar soluciones basadas en DNA involucra crear aplicaciones divididas en capas funcionales que se comunican entre sí.

La arquitectura consta de las siguientes capas:

- Capa de Presentación
- Capa de Negocios
- Capa de Datos

#### Capa de Presentación



#### Capa de Negocios



#### Capa de Datos

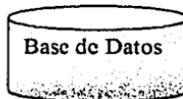


FIG II.1.2.2.A ARQUITECTURA DE 3 CAPAS

TESIS CON  
 FALTA DE ORIGEN

### **II.1.2.2.1 Capa de Presentación**

Los servicios de presentación son identificados con la interfaz de usuario, y normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final.

La capa de servicios de presentación es responsable de:

- Obtener información del usuario.
- Enviar la información del usuario a los servicios de negocios para su procesamiento.
- Recibir los resultados de los servicios de negocios.
- Presentar estos resultados al usuario.

### **II.1.2.2.2 Capa de Negocios**

Los servicios de negocios son la conexión entre un usuario y los servicios de datos. Una tarea de negocios es una operación definida por los requerimientos de la aplicación, como agregar una ruta, cambiar el padrón vehicular, etc.

Las reglas de negocio controlan el flujo de las tareas, éstas tienden a cambiar más frecuentemente que las operaciones específicas del sistema, es por ello que se pueden encapsular en componentes que están lógicamente separados de la lógica de la aplicación en sí.

El nivel de servicios de negocios es responsable de:

- Recibir la entrada del nivel de presentación.
- Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar.
- Enviar el resultado procesado al nivel de presentación.

### **II.1.2.2.3 Capa de Datos**

El nivel de servicios de datos es responsable de:

- Almacenar los datos.
- Recuperar los datos.
- Mantener los datos.
- Integridad de los datos.

### **II.1.2.3 Documentación**

#### **UML (UNIFIED MODELING LANGUAGE )**

UML (Lenguaje Unificado para la Construcción de Modelos) es un lenguaje que permite especificar, visualizar y construir modelos de los sistemas de información. Es un sistema notacional destinado al modelado de sistemas que utiliza conceptos orientados a objetos.

Surge por iniciativa de Grady Booch y Jim Rumbaugh por combinar el método de Booch y el OMT(Object Modeling Technique, Técnica de Modelado de Objetos) en 1994, posteriormente se les unió el creador del método OOSE(Object-Oriented Software Engineering, Ingeniería de Software Orientada a Objetos) Ivar Jacobson.

Propusieron al UML como notación estándar del lenguaje de construcción de modelos en 1997 a la OMG (Object Management Group, asociación que determina estándares de la industria), pero aunque no se había aceptado aun, ya era empleado por muchas organizaciones dedicadas al desarrollo de software, ya que representa métodos característicos de la primera generación de análisis y diseño orientado a objetos.

UML esta formado por diversos elementos gráficos que se combinan para integrar diagramas; la finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo.

---

Los diagramas más comunes de UML son:

- ◇ Diagrama de Casos de Uso
- ◇ Diagrama de Clases
- ◇ Diagrama de Objetos
- ◇ Diagrama de Estados
- ◇ Diagrama de Actividades
- ◇ Diagrama de Comportamiento
  - ⇒ Diagrama de Secuencias
  - ⇒ Diagrama de Colaboraciones
- ◇ Diagrama de Implementación
  - ⇒ Diagrama de Componentes
  - ⇒ Diagrama de Distribución

Es importante mencionar que en un modelo UML no es necesario que aparezcan todos los diagramas, la mayoría de modelos contienen un subconjunto de los diagramas anteriores, que son necesarios dependiendo de las necesidades del sistema que estemos desarrollando.

Un sistema cuenta con diversas personas implicadas, las cuales tienen enfoques particulares en diferentes aspectos del sistema, es por ello que son importantes los diagramas UML porque nos muestran una forma de incorporar una perspectiva en particular.

Ventajas:

- ❖ Fácil manejo de la complejidad.
- ❖ Implantación del control de calidad desde las primeras etapas de desarrollo.
- ❖ Buen control de avances y un mejor control directivo.

- ❖ Es independiente del lenguaje de programación o herramienta de implementación del sistema.
- ❖ 90% de las herramientas CASE lo soportan.
- ❖ Microsoft y el OMG lo toman como estándar.
- ❖ Lenguaje gráfico recomendable para ilustrar el diseño de aplicaciones en tres capas.

### **II.1.3 ANÁLISIS DE ÁREAS DE LA EMPRESA.**

Las áreas de empresa, son grupos de funciones y actividades de la empresa relacionadas lógicamente entre sí y dependientes de la estructura organizacional.

Las prioridades de las áreas de empresa se establecen normalmente conforme a su importancia y valor percibidos en el contexto de la empresa.

Esta fase consiste en evaluar las áreas de empresa que han de identificarse y establecer prioridades sobre proyectos de desarrollo específicos.

Los resultados de esta fase son proyectos planeados para el desarrollo de aplicaciones que serán posteriormente enviados al análisis del sistema.

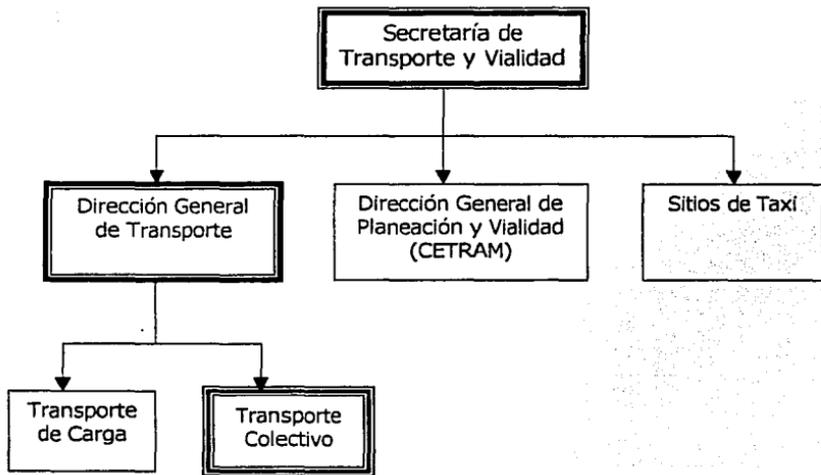
El análisis de áreas de empresa puede ser una fase muy larga, es por ello que en la mayoría de las empresas analizan una o dos áreas al mismo tiempo, preferiblemente aquellas que se identificaron como más importantes.

Este sistema debe integrar a nivel informático a la Dirección General de Transporte perteneciente a la Secretaría de Transporte y Vialidad del Gobierno del Distrito Federal.

La Dirección General de Transporte (DGT) a su vez se divide en unidades operacionales distintas: Transporte Colectivo y Transporte de Carga, que son manejados como unidades independientes.

De estas subdivisiones, *Transporte Colectivo*, es el elemento con mayor manejo de información, es por ello que será el primero en desarrollarse y en el cual se basa este trabajo.

En el organigrama II.1.3.A se puede observar claramente.



II.1.3.A ORGANIGRAMA

TESIS CON  
FALLA DE ORIGEN

## II.2 ANÁLISIS DEL SISTEMA

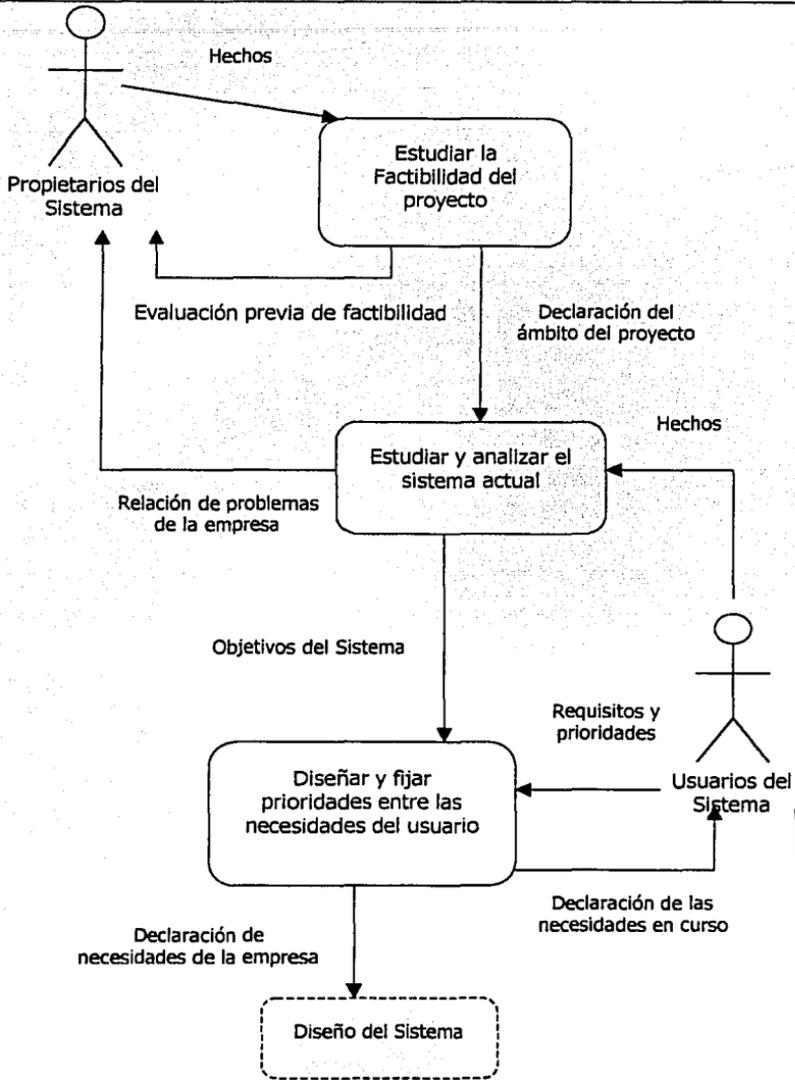
Su propósito es analizar el problema o situación de la empresa para poder definir las necesidades y prioridades manifestadas por los usuarios, para la creación de un sistema de información. En gran parte, el análisis se lleva a cabo a través de la cooperación de los usuarios.

En el cuadro II.2.A se puede ver el modelo de Análisis, que se sigue en la realización de este sistema:

II.2.1	Estudio de Factibilidad
II.2.2	Calendario de Actividades
II.2.3	Requerimientos de Información
II.2.4	Casos de Uso Diagramas de Casos de Uso

CUADRO II.2.A MODELO DE ANÁLISIS

En la figura II.2.B están las fases de un análisis típico de sistemas.



TESIS CON  
FALLA DE ORIGEN

FIG II.2.B FASES DEL ANÁLISIS DE SISTEMAS

## **II.2.1 ESTUDIO DE FACTIBILIDAD**

Se toma en cuenta la información obtenida en las entrevistas preliminares y se determina la posibilidad de que el sistema sea viable para la DGT; es decir la factibilidad del proyecto. Para ello es necesario tomar en cuenta lo siguiente:

### **II.2.1.1 Factibilidad Operacional**

Nos indica la viabilidad de que la aplicación sea operada por los usuarios.

Es necesario tomar en cuenta el gran apoyo al proyecto por parte de directivos de la DGT; cabe mencionar que también por parte de los usuarios existe interés y apoyo en el desarrollo del sistema, esto es importante ya que el sistema estará basado en sus necesidades.

### **II.2.1.2 Factibilidad Técnica**

Tiene que ver con la tecnología que se necesita para el manejo y desarrollo del sistema; si existe algún problema en esta etapa y no se detecta a tiempo, se afecta el desarrollo e implementación del sistema.

En este caso, se cuenta con un convenio con una empresa que se dedica a la venta de equipo, que además de proporcionar el equipo de cómputo necesario, instaló una red local en la dependencia.

El sistema podrá ser accesado por  $n$  usuarios a la vez, ya que tendrá una arquitectura cliente / servidor (siempre y cuando se tenga instalado el sistema y se cuenta con la dirección IP del servidor al que se accederá).

Otro punto importante, es que si se requiere agregar alguna nueva pantalla o módulo no habría ningún problema, ya que su construcción es en forma modular, lo mismo sucedería si se desea modificar algún elemento, sólo éste y sus dependencias (si es que las hay) sufrirán cambios, se tendrá un sistema flexible, que se podrá expandir de acuerdo a nuevas necesidades del usuario o de la empresa.

Por último cabe mencionar que se debe contar con un control estricto en el acceso al sistema, ya que por seguridad se designarán claves y contraseñas específicas dependiendo del área a la que se desee acceder.

### II.2.1.3 Factibilidad Económica-Financiera.

El proyecto representa beneficios económicos, sin embargo las cifras son confidenciales, siendo importante señalar que más allá de los beneficios económicos resultan de mayor trascendencia los beneficios sociales del proyecto, que redundará en un mejor servicio al público usuario, eficientando los servicios de transporte público para la ciudadanía y propiciando la disminución de actos de corrupción o desviación de recursos.

Los gastos se recuperarán con el tiempo, ya que se obtendrán beneficios, como los siguientes:

- ✓ Se tendrá una disminución en costos de operación, ya que como los tareas son realizadas de manera manual, los reportes, oficios, etc, que se tardaban días en realizarse, ahora se tendrían en minutos, con información veraz y oportuna;
- ✓ Disminuirían los gastos del área de papelería ya que son muchas las formas y oficios que deben llenarse;

Considerando los aspectos operativos, técnicos y económicos se puede decir que la realización del sistema es FACTIBLE.

### II.2.2 CALENDARIO DE ACTIVIDADES

El calendario de actividades propuesto fue el siguiente:

TESIS CON  
FALLA DE ORIGEN

ACTIVIDAD	TIEMPO (MESES)					
	1	2	3	4	5	6
Planeación del Sistema	⊗	⊗				
Análisis del Sistema	⊗	⊗	⊗			
Diseño del Sistema		⊗	⊗	⊗		
Construcción del Sistema		⊗	⊗	⊗	⊗	
Implementación del Sistema						⊗
Elaboró	Supervisó			VoBo		

## **II.2.3 REQUERIMIENTOS DE INFORMACIÓN**

Los requerimientos son una descripción de las necesidades o deseos de un producto. El objetivo principal de la fase de requerimientos es identificar y documentar lo que en realidad se necesita, en una forma que claramente se lo comunique al cliente y a los miembros del equipo de desarrollo.

El reto consiste en definirlos de manera inequívoca, de modo que se detecten los riesgos y no se presenten problemas al momento de entregar el producto.

Se toman en cuenta los siguientes elementos:

### **II.2.3.1 Panorama General**

Desarrollo de un sistema informático integral para el manejo de información que permita la gestión y análisis de datos, que sean útiles para realizar consultas, así como para tener un mayor control del Transporte Colectivo.

### **II.2.3.2 Clientes**

Dirección General de Transporte, específicamente en el área de Transporte Colectivo.

### **II.2.3.3 Metas**

- ✓ Organización de los datos para la administración de:
  - Aproximadamente poco más de 100 rutas locales.
  - 100 rutas del Estado de México.
  - Aproximadamente 1400 derroteros.
- ✓ Elaboración de 3 procedimientos (Agregar, Cambiar y Eliminar).
- ✓ Realización de 4 tipos de reportes.
- ✓ Disminución considerable del tiempo de consulta de la información.

### II.2.3.4 Funciones del Sistema

Las funciones del sistema son lo que éste habrá de hacer, se clasifican por prioridad como se ve a continuación:

<b>Categoría de la función</b>	<b>Significado</b>
Evidente	Debe realizarse y el usuario debe saber que se ha realizado.
Ocultas	Debe realizarse, aunque no es visible para los usuarios. Las funciones ocultas a menudo se omiten durante el proceso de obtención de requerimientos.
Superflua	Opcionales: su inclusión no repercute significativamente en el costo ni en otras funciones.

TABLA II.2.3.4.A PRIORIDAD DE LAS FUNCIONES DEL SISTEMA

A continuación se mencionan las funciones básicas del sistema:

<b>Ref #</b>	<b>Función</b>	<b>Categoría</b>
R1.1	Validar la clave y contraseña que ingrese el usuario en el sistema.	Evidente
R1.2	Agregar información.	Evidente
R1.3	Cambio de información, si es que se requiere alguna modificación posterior.	Evidente
R1.4	Eliminación de información que ya no es requerida.	Evidente
R1.5	Cambio en el estado de la información, es decir si esta vigente o no.	Evidente
R1.6	Guardar la información.	Ocultas
R1.7	Realizar reportes de información actualizada.	Evidente
R1.8	Imprimir dichos reportes	Superflua
R1.9	Ofrecer mecanismos de comunicación entre los procesos y entre los formularios.	Ocultas
R1.10	Realizar consultas específicas.	Evidente

### II.2.3.5 Atributos del Sistema

Son sus características o cualidades; los atributos pueden abarcar todas las funciones o ser específicos de una sola.

Los atributos tienen un conjunto de detalles de atributos los cuales tienden a ser valores discretos, confusos o simbólicos o pueden tener restricciones de frontera del atributo, que son condiciones obligatorias de frontera generalmente en un rango numérico.

Atributo	Detalle y Restricción de Frontera
Interfaz gráfica	(Detalle) Formularios gráficos y cuadros de diálogo, con un color estandarizado según el área que corresponda.  (Detalle) Fácil navegación dentro de los formularios por medio del teclado.
Amigable	(Detalle) Fácil de entender y manejar.
Tiempo de respuesta	(Restricción de Frontera) Al dar de alta, cambiar o eliminar datos se actualiza la información en un mínimo de segundos.
Plataforma del Sistema Operativo	(Detalle) Windows 2000 Advanced Server
Confiable	(Detalle) Sólo se podrán realizar las operaciones permitidas según el usuario.

### II.2.4 CASOS DE USO

Una técnica excelente de UML (Lenguaje Unificado para la Construcción de Modelos) que permite mejorar la comprensión de los requerimientos es la creación de casos de uso, que es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso.

Los casos de uso requieren tener al menos un conocimiento parcial de los requerimientos del sistema.

Existen 2 formatos para los casos de uso:

- Un **caso expandido de uso** muestra más detalles que uno de alto nivel; este tipo de casos suelen ser útiles para alcanzar un conocimiento más profundo de los procesos y de los requerimientos. Durante la fase de

---

especificación de requerimientos, conviene escribir en el formato expandido los casos más importantes y de mayor influencia, en cambio los menos importantes pueden posponerse hasta el diseño.

- Un **caso de uso de alto nivel** describe un proceso muy brevemente en 2 ó 3 enunciados. Es empleado en la etapa de requerimientos a fin de entender rápidamente el grado de complejidad y funcionalidad del sistema.

Para establecer prioridades en el desarrollo de los casos de uso, se pueden clasificar en:

- **Casos primarios de uso** representan los procesos comunes más importantes.
- **Casos secundarios de uso** representan procesos menores o raros.
- **Casos opcionales de uso** representan procesos que pueden no abordarse.

Cabe mencionar otra clasificación, tomando en cuenta la extensión del caso:

- **Casos esenciales de uso:** Son casos expandidos que se expresan en forma teórica que contiene poca tecnología y pocos detalles de implementación; las decisiones de diseño se posponen y se abstraen de la realidad, como las referentes a la interfaz de usuario.

Describe el proceso a partir de sus actividades y motivos esenciales.

Los casos de alto nivel siempre son de carácter esencial, debido a su brevedad y abstracción.

Es recomendable usarlos al comenzar a investigar los requerimientos, con el propósito de entender mejor el alcance del problema y las funciones necesarias; permiten captar la esencia del proceso y sus motivos fundamentales. No se alteran con el paso del tiempo, por ejemplo, una empresa puede recuperar y releer los casos esenciales de uso tiempo después y aplicarlos a un nuevo proyecto, ya que excluye los detalles de diseño.

- **Casos reales de uso:** Describe el proceso a partir de su diseño actual, sujeto a las tecnologías específicas de entrada y salida, etc. Cuando se trata de la interfaz para el usuario, ofrece la presentación de pantallas, botones, etc, y explica la interacción entre ellas.

En teoría, los casos reales de uso se crean durante la fase de diseño; en algunos proyectos se prevén decisiones de diseño concernientes a la interfaz para el usuario, de ahí la necesidad de crear casos reales en la fase inicial de elaboración, para que se pueda aceptar el diseño prematuramente.

#### **II.2.4.1 Identificación de los Casos de Uso.**

Para identificar los casos de uso se requiere de la utilización de una técnica de análisis, como lo es la lluvia de ideas y de revisar los documentos actuales sobre la especificación de requerimientos, así como la información recabada en las entrevistas realizadas.

Existen 2 métodos para identificar los casos de uso:

- ❖ Un método con que se identifican los casos de uso se basa en los actores.
  1. Se identifican los actores relacionados con un sistema o empresa.
  2. En cada actor, se identifican los procesos que inicia o en que participa.
  
- ❖ Un segundo método de identificación de los casos de uso se basa en eventos.
  1. Se identifican los eventos externos a los que un sistema ha de responder.
  2. Se relacionan los eventos con los actores y con los casos de uso.

#### **II.2.4.2 Determinación de Actores.**

Una vez identificadas las operaciones de la empresa, es el momento de determinar quien las hará cumplir; es necesario definir las responsabilidades de los empleados para poder distribuir las tareas necesarias para el correcto desempeño de dichas operaciones, así como definir de que manera colaboran o comparten el trabajo.

En este caso se detectan 3 actores principales: el administrador que tendrá control absoluto de la información, el supervisor que tendrá algunas restricciones y el capturista que sólo se dedica a esta actividad.

### II.2.4.3 Determinación de Casos de Uso en el Formato de Alto Nivel.

A continuación se muestran los casos de uso para la DGT en el segmento de Transporte Colectivo:

Caso de Uso	<b>Verificando Usuario</b>
Actor	Capturista o Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra a la ventana principal de acceso al sistema donde ingresa una clave y contraseña, proporcionados por el administrador, para permitirle el acceso a las que partes del sistema que le corresponden. Si los datos son incorrectos no permite la entrada al sistema.

Caso de Uso	<b>Agregando Ruta</b>
Actor	Capturista o Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere agregar una ruta, captura la información requerida y posteriormente se agrega al sistema.

Caso de Uso	<b>Agregando Derrotero</b>
Actor	Capturista o Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere agregar un derrotero, verifica que exista la ruta deseada y comienza a capturar la información requerida, posteriormente se agrega al sistema.

Caso de Uso	<b>Agregando RecorridoDerrotero</b>
Actor	Capturista o Supervisor o Administrador
Tipo	Primario, esencial
Descripción	Después de haber agregado un derrotero, se captura la información requerida que indica el recorrido que tiene éste, posteriormente se agrega al sistema.

Caso de Uso	<b>Agregando PadronVehicularRuta</b>
Actor	Capturista o Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere agregar un vehículo, elige la ruta y derrotero correspondiente y comienza a capturar la información requerida, posteriormente se agrega al sistema.

Caso de Uso	<b>Agregando AccidenteRuta</b>
Actor	Capturista o Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere agregar un accidente, elige la ruta, derrotero y placas, comienza a capturar la información requerida y posteriormente se agrega al sistema.

Caso de Uso	<b>Agregando Apercibimiento</b>
Actor	Capturista o Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere agregar un apercibimiento, elige la ruta y derrotero correspondiente, comienza a capturar la información requerida y posteriormente se agrega al sistema.

Caso de Uso	<b>Agregando QuejaRuta</b>
Actor	Capturista o Supervisor o Administrador
Tipo	Primario, esencial

Descripción	El actor entra al sistema, especifica que quiere presentar una queja, elige la ruta y derrotero correspondiente, comienza a capturar la información requerida y posteriormente se agrega al sistema.
-------------	--

Caso de Uso	<b>Agregando EstadoCuentaRuta</b>
Actor	Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere agregar un estado de cuenta, verifica la existencia de la ruta y comienza a capturar la información requerida, posteriormente se agrega al sistema.

Caso de Uso	<b>Cambiando Ruta</b>
Actor	Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere cambiar información de una ruta y comienza a modificarla, posteriormente se guarda en el sistema.

Caso de Uso	<b>Cambiando Derrotero</b>
Actor	Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere cambiar información de un derrotero y comienza a modificarla, posteriormente se guarda en el sistema.

Caso de Uso	<b>Cambiando Recorrido_Derrotero</b>
Actor	Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere cambiar información del recorrido de un derrotero, comienza a modificarla y posteriormente se guarda en el sistema.

Caso de Uso	<b>Cambiando PadronVehicularRuta</b>
Actor	Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere cambiar información de un vehículo en especial, comienza a modificarla, posteriormente se guarda en el sistema.

Caso de Uso	<b>Cambiando AccidenteRuta</b>
Actor	Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere cambiar información de un accidente específico y comienza a modificarla, posteriormente se guarda en el sistema.

Caso de Uso	<b>Cambiando Apercibimiento</b>
Actor	Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere cambiar información de un apercibimiento y comienza a modificarla, posteriormente se guarda en el sistema.

Caso de Uso	<b>Cambiando QuejaRuta</b>
Actor	Supervisor o Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere cambiar información de una queja y comienza a modificarla, posteriormente se guarda en el sistema.

Caso de Uso	<b>Cambiando EstadoCuentaRuta</b>
Actor	Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere cambiar información de un estado de cuenta y comienza a modificarla, posteriormente se guarda en el sistema.

Caso de Uso	<b>Obteniendo Reportes</b>
Actor	Supervisor o Administrador
Tipo	Secundario, esencial
Descripción	El actor entra al sistema, especifica que quiere obtener un reporte con determinada información.

Caso de Uso	<b>Eliminando Ruta</b>
Actor	Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere eliminar una ruta, automáticamente se elimina del sistema no de la base, es decir sólo se cambia su estado a NoVigente.

Caso de Uso	<b>Eliminando Derrotero</b>
Actor	Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere eliminar un derrotero, automáticamente se elimina del sistema no de la base, es decir sólo se cambia su estado a NoVigente.

Caso de Uso	<b>Eliminando PadronVehicularRuta</b>
Actor	Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere eliminar un vehículo, automáticamente se elimina del sistema no de la base, es decir sólo se cambia su estado a NoVigente.

Caso de Uso	<b>Eliminando AccidenteRuta</b>
Actor	Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere eliminar un accidente, automáticamente se elimina de la base.

Caso de Uso	<b>Eliminando Apercibimiento</b>
Actor	Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere eliminar un apercibimiento, automáticamente se elimina de la base.

Caso de Uso	<b>Eliminando QuejaRuta</b>
Actor	Administrador
Tipo	Primario, esencial
Descripción	El actor entra al sistema, especifica que quiere eliminar una queja, automáticamente se elimina de la base.

## II.2.5 DIAGRAMA DE CASOS DE USO

El diagrama de casos de uso representa gráficamente la forma en como un cliente (actor) trabaja con el sistema de desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones).

Un diagrama de casos de uso consta de:

- ❖ *Actor*: es un rol que un usuario juega con respecto al sistema, un actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.
- ❖ *Casos de uso*: es una operación/ tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.
- ❖ *Relaciones de Uso, Herencia y Comunicación*

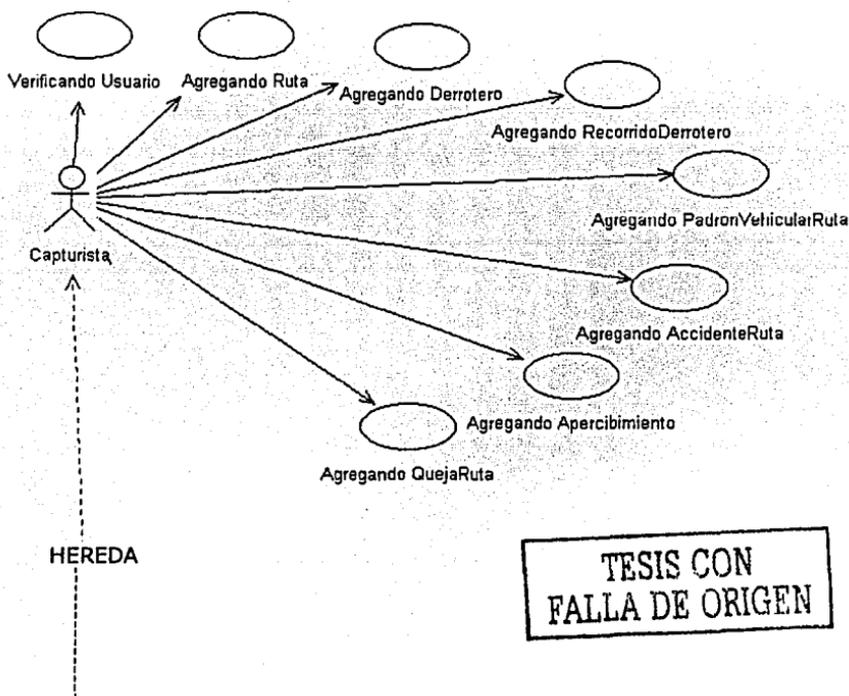
En general se tienen casos de uso para agregar, cambiar y eliminar elementos. Tomando en cuenta que una función del sistema, es un caso de uso si se debe indicar explícitamente al sistema que se quiere acceder a esta función.

Existe una *jerarquización* entre actores, donde un actor puede ejecutar todos los casos que ejecute otro actor; en este caso, el administrador puede hacer todo lo que hace el supervisor, pero además puede eliminar; a su vez el supervisor puede hacer todo lo que hace el capturista, además de poder realizar cambios.

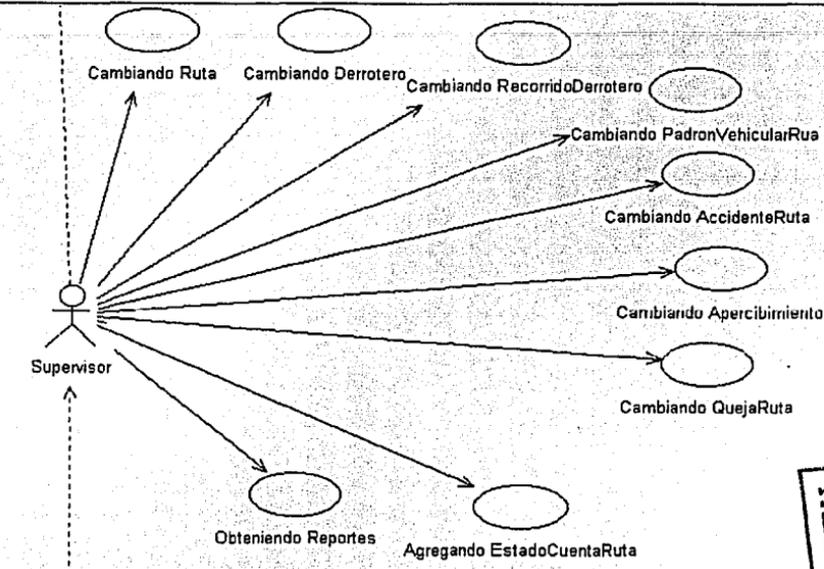
El administrador hereda al supervisor y este hereda al capturista. (La flecha simple representa la asociación entre un actor o caso de uso y otra operación (caso de uso).)

Es importante recordar que los casos de uso están expresados desde el punto de vista del actor.

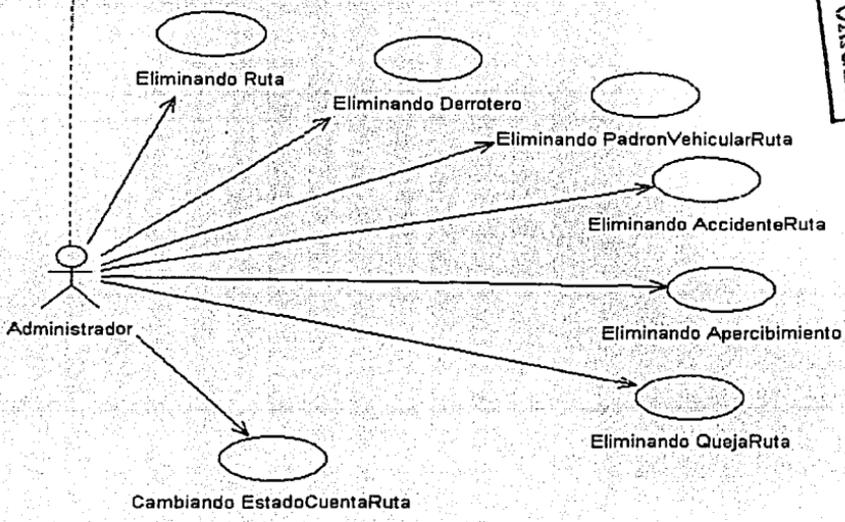
Al relacionar los casos de uso y los actores definidos anteriormente se obtiene el siguiente Diagrama de Casos de Uso.



TESIS CON  
FALLA DE ORIGEN



HEREDA



TESIS CON  
FALLA DE ORIGEN

A continuación se escriben en *formato esencial expandido* los casos de uso más importantes, a fin de entender y estimar mejor la naturaleza y las dimensiones del problema; se posponen los casos de uso menos importantes hasta el ciclo de diseño.

Caso de Uso	<b>Agregando Ruta</b>
Actores	Capturista o Supervisor o Administrador
Propósito	Dar de alta una ruta.
Tipo	Primario, esencial
Resumen	El actor especifica que quiere agregar una ruta y comienza a capturar la información requerida, posteriormente se agrega al sistema.
Referencias cruzadas	Funciones: R1.1, R1.2, R1.6 (Pág.33) Casos de uso: El actor debe haber concluido el caso de uso: <i>Verificando Usuario</i> .

### Curso Normal de los Eventos

<b>Acción de los Actores</b>	<b>Respuesta del Sistema</b>
<p>1. Para llevar a cabo este caso de uso, el actor elige la opción de Alta Ruta.</p> <p>3.El actor comienza a capturar la información de la ruta que desee, al terminar acepta la captura.</p> <p>5.El actor cierra la ventana de captura.</p>	<p>2.Le muestra el formulario para iniciar la captura.</p> <p>4.Al obtener respuesta del actor (aceptar captura) automáticamente agrega al sistema el nuevo registro.</p> <p>6.El sistema queda en espera de otra selección.</p>

- **Cursos Alternos**

- Línea 3. El actor puede cancelar para no guardar la información capturada, por lo tanto se suspende la operación de agregar.

Caso de Uso	<b>Agregando Derrotero</b>
Actores	Capturista o Supervisor o Administrador
Propósito	Dar de alta un derrotero.
Tipo	Primario, esencial
Resumen	El actor entra al sistema, especifica que quiere agregar un derrotero y comienza a capturar la información requerida, posteriormente se agrega al sistema.
Referencias cruzadas	Funciones: R1.1, R1.2, R1.6 (Pág.33) Casos de uso: El actor debe haber concluido el caso de uso: <i>Verificando Usuario</i> .

*Curso Normal de los Eventos*

<b>Acción de los Actores</b>	<b>Respuesta del Sistema</b>
<p>1.El actor elige la opción de Alta Derrotero.</p> <p>3.El actor comienza a capturar la información del derrotero que desee, seleccionando primero la ruta que le corresponde; al terminar acepta la captura de información.</p> <p>5.El actor cierra la ventana de captura.</p>	<p>2.Le muestra el formulario para iniciar la captura.</p> <p>4.Al obtener la respuesta aceptable del actor, verifica que los datos capturados sean válidos y automáticamente agrega al sistema el nuevo registro.</p> <p>6.El sistema queda en espera de otra selección.</p>

- *Cursos Alternos*

- Línea 3. Si no existe la ruta, deberá ser dada de alta primero. El actor puede cancelar para no guardar la información capturada.
- Línea 4. Si existen datos incorrectos, manda un mensaje de error, el sistema no permite agregar un registro al sistema si éste tiene algún error detectado.

<b>Caso de Uso</b>	<b>Agregando Padron Vehicular Ruta</b>
Actores	Capturista o Supervisor o Administrador
Propósito	Dar de alta un vehículo.
Tipo	Primario, esencial
Resumen	El actor entra al sistema, especifica que quiere agregar un vehículo y comienza a capturar la información requerida, posteriormente se agrega al sistema.
Referencias cruzadas	Funciones: R1.1, R1.2, R1.6 (Pág.33)  Casos de uso: El actor debe haber concluido el caso de uso: <i>Verificando Usuario</i> .

*Curso Normal de los Eventos*

<b>Acción de los Actores</b>	<b>Respuesta del Sistema</b>
<p>1.El actor elige la opción de Alta Padron Vehicular Ruta.</p> <p>3.El actor comienza a capturar la información del vehículo que desee, seleccionando antes la ruta y el derrotero correspondiente, al terminar acepta agregar la información al sistema.</p> <p>5.El actor cierra la ventana de captura.</p>	<p>2.Le muestra el formulario para iniciar la captura.</p> <p>4.Al obtener la respuesta aprobatoria del actor, verifica que los datos capturados sean válidos, automáticamente agrega al sistema el nuevo registro.</p>

	6.El sistema queda en espera de otra selección.
--	---

- *Cursos Alternos*

- Línea 3. Si no existe la ruta, debe ser dada de alta primero; si se desconoce el nombre del derrotero se elige 'NINGUNO'. El actor puede cancelar para no guardar la información capturada.
- Línea 4. Si existen datos incorrectos o falta agregar el derrotero, manda un mensaje de error; el sistema no permite agregar un registro al sistema si éste tiene algún error detectado.

Hoy en día, es necesario contar con sistemas bien planeados y analizados, ya que un cliente tiene que comprender qué es lo que se hará, además tiene que ser capaz de señalar cambios si no se han captado claramente sus necesidades o si cambia de opinión durante el proceso.

# ***CAPITULO III***

## ***DISEÑO Y CONSTRUCCIÓN DEL SISTEMA***

Después de llevar a cabo la planeación y el análisis, se prosigue con la fase de diseño y construcción, en el cual se traduce el análisis en forma de pasos o algoritmos que constituyen la base para la programación.

En esta etapa se diseñan los procedimientos que sirven para cumplir con el objetivo del sistema y la forma en como ingresan los datos al sistema. Además se especifica el proceso para producir los resultados deseados.

### **III.1 DISEÑO Y CONSTRUCCIÓN DEL SISTEMA**

El Diseño se ocupa de la especificación y construcción de la estructura del sistema de información, se divide en: *diseño de bases de datos* y el *diseño de aplicaciones*; ambas actividades de diseño se mencionan conjuntamente. La fase de diseño se dividió en 3 etapas: el diseño conceptual, lógico y físico (construcción del sistema).

#### **III.1.2 JUSTIFICACIÓN DEL HARDWARE Y SOFTWARE PROPUESTO PARA LA CONSTRUCCIÓN DEL SISTEMA.**

Existen diferentes causas para elegir determinado hardware y software para un sistema dado. Sin embargo en este caso la dependencia ya había adquirido un servidor ProLiant ML350 de COMPAQ con procesador Intel Pentium III y suficiente equipo de cómputo de acuerdo a sus necesidades, además cuenta con una red de área local, por lo que sólo nos enfocaremos al desarrollo del sistema.

#### **JUSTIFICACIÓN DE LA ARQUITECTURA DE TRES CAPAS (DNA).**

Existen diferentes servicios de datos, incluyendo los sistemas administradores de bases de datos relacionales (DBMS) como Microsoft SQL Server.

Lo que propone DNA para el acceso a datos es el *Universal Data Access* (UDA), en vez de requerir que todos los datos se encuentren en un almacenamiento común, UDA provee una interfaz programable común a virtualmente cualquier tipo de almacenamiento.

UDA especifica interfaces a nivel de sistema llamadas OLE DB. Proveedores de datos como son SQL Server implementan interfaces OLE DB para proveer acceso a sus almacenamientos específicos.

La arquitectura DNA de tres capas cuenta con servicios específicos en cada capa que se comunican entre sí mediante COM (Component Object Model, Modelo de Objeto Componente).

Las aplicaciones DNA mejoran el flujo de información dentro y fuera de la organización, son flexibles al cambio en la medida en que la empresa lo requiera, se integran fácilmente con los sistemas y datos existentes.

La arquitectura DNA está diseñada para maximizar:

- ❖ **Autonomía.** Se refiere a la habilidad de la aplicación para operar sus recursos críticos (son los recursos requeridos por una aplicación para funcionar confiablemente como una entidad independiente).
- ❖ **Confiabilidad.** Habilidad de la aplicación para proporcionar resultados exactos.
- ❖ **Disponibilidad.** Se refiere a la cantidad de tiempo que una aplicación es capaz de dar servicio confiablemente a las peticiones del cliente. Es importante debido a que una aplicación solamente es útil cuando está disponible para dar servicio a las peticiones de los clientes. La disponibilidad de la aplicación es dependiente de la disponibilidad del hardware, software y de la red.
- ❖ **Escalabilidad.** Es poder agregar o quitar recursos adicionales, que permitan a una aplicación aumentar su potencial.
- ❖ **Interoperabilidad.** Es la habilidad de una aplicación para acceder a aplicaciones, datos o recursos en otras plataformas.

Por estas razones, es que se propone a la Arquitectura de 3 Capas como una buena opción para elaborar este sistema.

### III.1.2.1 Selección del Sistema Operativo

Los equipos clientes funcionan en plataforma Windows98, que es la que tienen actualmente .

El servidor que se tiene, puede funcionar en cualquier plataforma de Windows Server. En este caso, la dependencia ya cuenta con **Windows 2000 Advanced Server**:

☐ *Windows 2000 Advanced Server* El sistema operativo de servidor diseñado para aplicaciones escalables de comercio electrónico y de línea de empresa con más demanda de recursos.

Se trata de una versión mejorada de Windows 2000 Server y cuenta con las siguientes características:

- ❖ Es un sistema multi-hilo que te permite manejar fácilmente el tráfico en la web.
- ❖ Soporta aplicaciones CGI tales como Perl o PHP, para aplicaciones futuras.
- ❖ El servicio de organización por clústeres de dos nodos y alta disponibilidad asegura que las aplicaciones críticas funcionen correctamente y se ejecuten cuando se les solicite.
- ❖ El equilibrio de carga en la red redistribuye la carga de trabajo del Web entre los servidores que no presentan errores; la arquitectura de núcleo mejorada proporciona mejoras de diseño para aumentar el tiempo de funcionamiento del sistema.
- ❖ Compatibilidad con actualizaciones continuas para reducir el tiempo de inactividad planeado para realizar servicios de mantenimiento o actualizaciones.
- ❖ Se manejan los clústeres de manera remota con el administrador de clústeres y la interfaz de secuencias de línea de comandos NLB desde cualquier sistema Windows 2000 conectado a la red.
- ❖ Se puede contar con 8 procesadores y compatibilidad con una cantidad total de 8 GB de RAM (memoria de acceso aleatorio). Ideal para ejecutar las aplicaciones actuales con mayor demanda de recursos.

Este fue adquirido por su funcionalidad y escalabilidad, pensando en aplicaciones futuras, con un manejo mayor de información y usuarios, así como consultas en línea y aplicaciones vía Internet.

### III.1.2.2 Selección del DBMS

Debe elegirse el DBMS más adecuado para las necesidades del sistema, por lo que investigando algunos de los productos existentes en el mercado como *Oracle* o *Sybase* se puede decir que son motores demasiado grandes para los propósitos de

la aplicación, consumen grandes recursos del hardware, licencias de altos costos y actualmente se dispone de poca documentación al respecto. Access podría ser otra opción, pero sólo nos sirve para programas pequeños, poco robustos y sin seguridad; por lo que es deficiente para los recursos a manejarse.

Es por ello, que se elige a **Microsoft SQL Server 2000**, que cuenta con las siguientes características:

- ❖ Administración remota de los recursos del sistema
- ❖ Generación automática de copias de seguridad
- ❖ Soporte de tolerancia a fallas
- ❖ Monitoreo gráfico del rendimiento del sistema
- ❖ Arquitectura de un sólo proceso pero multitareas, asegura un desempeño del sistema eficiente para un gran número de usuarios con poco consumo de recursos.

Ventajas:

- ✓ Menor costo de adquisición.
- ✓ Consume menores recursos de hardware.
- ✓ Se ejecuta sobre ambientes gráficos que ayudan tanto al cliente para el desarrollo de aplicaciones, como al administrador para tener mejor control del sistema.
- ✓ Permite enviar, modificar y guardar información fácilmente desde cualquier origen en tablas relacionales de SQL.
- ✓ Se pueden crear índices en vistas para mejorar el rendimiento de las consultas existentes sin necesidad de volver a programar todo.
- ✓ Cuenta con restricciones de integridad referencial en cascada para controlar como se realizan modificaciones en las tablas cuando se actualizan las llaves.

### III.1.2.3 Selección de la Herramienta de Desarrollo

Se considero a **Visual Basic 6.0** ya que permite generar aplicaciones que realizan diversas acciones por medio de ventanas y controles gráficos, para la manipulación de información permitiendo que un usuario trabaje con un sistema con ambiente Windows, lo que permite que sea amigable y compatible con una gran cantidad de software y hardware comerciales, sin que sea necesario el empleo de recursos muy sofisticados o costosos.

Características:

- ❖ Soporta la creación de aplicaciones cliente/servidor mediante el uso de ODBC (Open DataBase Connectivity, Conectividad de Bases de Datos Abiertas), ya que permite que una misma aplicación pueda interactuar con distintos manejadores de bases de datos.
- ❖ Utiliza programación orientada a eventos, es decir cuando suceden ciertos eventos (mover, clic, doble clic) se realizan acciones que se encuentran programadas en éstos.
- ❖ Permite crear aplicaciones ejecutables que pueden ser distribuidas a todos los usuarios, sin necesidad de que éstos tengan Visual Basic instalado.
- ❖ Cuenta con un diseñador de entorno de datos para diseñar visualmente bases de datos de Microsoft SQL Server, así como para crear consultas reutilizables de acceso a datos.
- ❖ Permite generar aplicaciones gráficas que pueden acceder y manipular información de bases de datos relacional.

Entre sus ventajas tenemos:

- ✓ Proporciona un conjunto de herramientas para integrar bases de datos con cualquier aplicación.
- ✓ Menores costos de implementación y mantenimiento.
- ✓ Sencillez para programar aplicaciones de cierta complejidad para Windows.
- ✓ Se facilita el código para enlazar controles a los orígenes de datos.
- ✓ Enlace a múltiples orígenes de datos para agregar y manipular los datos.

Además de las ventajas anteriormente expuestas, se debe tomar en cuenta que para su mantenimiento posterior, será más fácil contratar a personal especializado en el manejo de Visual Basic, ya que es una herramienta que ha tenido gran difusión comercial por su facilidad de uso así como su bajo costo.

La principal desventaja es la necesidad de archivos adicionales además del ejecutable.

#### **III.1.2.4 ¿Por qué es importante el uso de una herramienta de desarrollo en conjunto con un DBMS ?**

Los DBMS trabajan de una forma general con las bases de datos y cuando se trata de resolver problemas concretos no son muy eficaces si el usuario no tiene conocimientos profundos de la aplicación y raras veces los tienen. En cambio, mediante una aplicación diseñada para dicho fin, en este caso en Visual Basic podemos manipular los datos de la base a nuestro gusto, de forma que se resuelvan problemas concretos mediante aplicaciones que sean fáciles de usar por cualquier persona que no tenga ningún conocimiento de bases de datos ni de programación.

### **III.2 DISEÑO CONCEPTUAL**

Se obtiene un modelo conceptual, el cual es una descripción de la estructura del sistema, independiente del software que se use.

La función del diseño conceptual es describir el contenido de información de la base, más que las estructuras de almacenamiento que se requerirán para el manejo de dicha información.

Toma como base los requerimientos del sistema.

#### **III.2.1 MODELO CONCEPTUAL**

Un modelo conceptual explica los conceptos significativos en un dominio del problema real, define y modela los aspectos más importantes acerca de la información que la empresa necesita obtener y las relaciones entre dicha información; no es una descripción del software.

En UML (Lenguaje Unificado para la Construcción de Modelos) se representa con un grupo de *diagramas de estructura estática* donde no se define ninguna operación.

Estos diagramas están formados por:

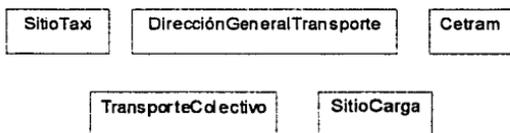
- Conceptos (Objetos)
- Asociaciones entre objetos
- Atributos de los conceptos

### III.2.1.1 CONCEPTOS

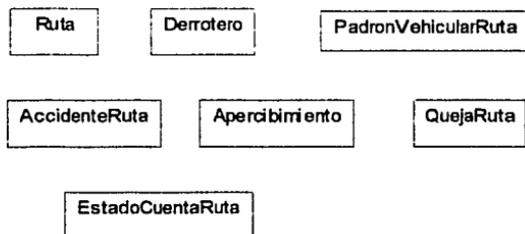
Los conceptos son una idea, cosa u objeto y podemos representarlo de manera formal por medio de:

- Símbolo: palabras o imágenes que representan un concepto.
- Intención: definición del concepto.
- Extensión: el conjunto de ejemplos a que se aplica el concepto.

El modelo conceptual inicial del dominio SETRAVI sería:



El modelo conceptual inicial del dominio Transporte Colectivo es:



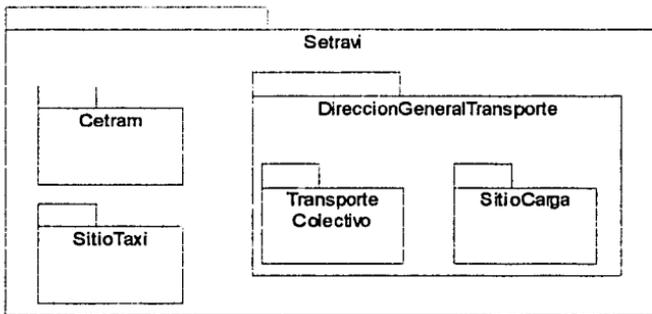
TESIS CON  
 FALTA DE ORIGEN

Cabe mencionar que un elemento importante es el de *Reportes*, que nos muestra la información obtenida al realizar consultas, pero en este caso no se anexa como concepto, ya que toda su información proviene de otras fuentes.

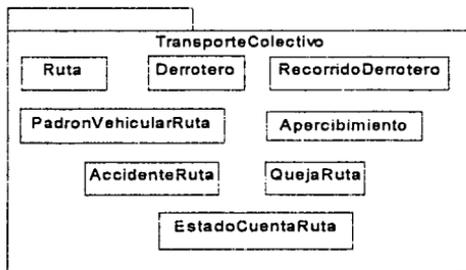
En algunas ocasiones se tiene la necesidad de distribuir el modelo conceptual en *paquetes*, con la finalidad de contemplar el modelo en agrupamientos más simples, representados por una carpeta tabular.

Un elemento, tipo o clase es una *propiedad* del paquete dentro del cual esta definido.

A continuación se crea un paquete del dominio Setravi que contiene a su vez tres paquetes más: Cetram, SitioTaxi y DirecciónGeneralTransporte, este último tiene 2 paquetes: TransporteColectivo y SitioCarga. Cada uno de estos paquetes esta formado por un conjunto de propiedades específicas.



Por ejemplo, dentro del paquete Transporte Colectivo, se tienen las siguientes propiedades:



TESIS CON  
FALTA DE ORIGEN

### III.2.1.2 Asociaciones

La asociación es una relación entre dos conceptos que indica alguna conexión entre ellos, se representa como una línea entre conceptos con el nombre de la asociación, ésta es bidireccional, es decir es posible una relación entre los objetos de un tipo y los de otro.

En UML, **asociación** tiene el mismo significado que la **relación** en el modelo E-R.

La expresión de multiplicidad es parte de una asociación (se asignan en sus extremos) e indica cuantas instancias de la clase A se asocian a la clase B.

Existen los siguientes valores de multiplicidad:

De Cero a Muchos	0 .. *
De Uno a Muchos	1 .. *
Muchos	*
Exactamente n	N
Exactamente a,b,c	a,b,c
De Cero a Uno	0 .. 1

TABLA III.2.1.1 MULTIPLICIDAD

Los nombres de las asociaciones comienzan con mayúsculas, una frase se debe construir con guiones. La dirección por omisión en que debe leerse el nombre de una asociación es de izquierda a derecha o de arriba hacia abajo.

Cuando se crea un modelo conceptual, se pueden definir asociaciones que no se necesitan en la construcción, y viceversa, ya que se pueden implementar asociaciones que se omitieron durante la fase de análisis.

### III.2.1.3 Atributos

Atributo: es un valor lógico de un dato perteneciente a un objeto. Para obtener los *Atributos* de los conceptos se necesita conocer los requerimientos de información establecidos en los casos de uso.

Entre los tipos de *atributos simples o valores puros de datos* tenemos: Boolean, fecha, cadena, carácter, hora, número; en el diagrama es opcional establecer su tipo.

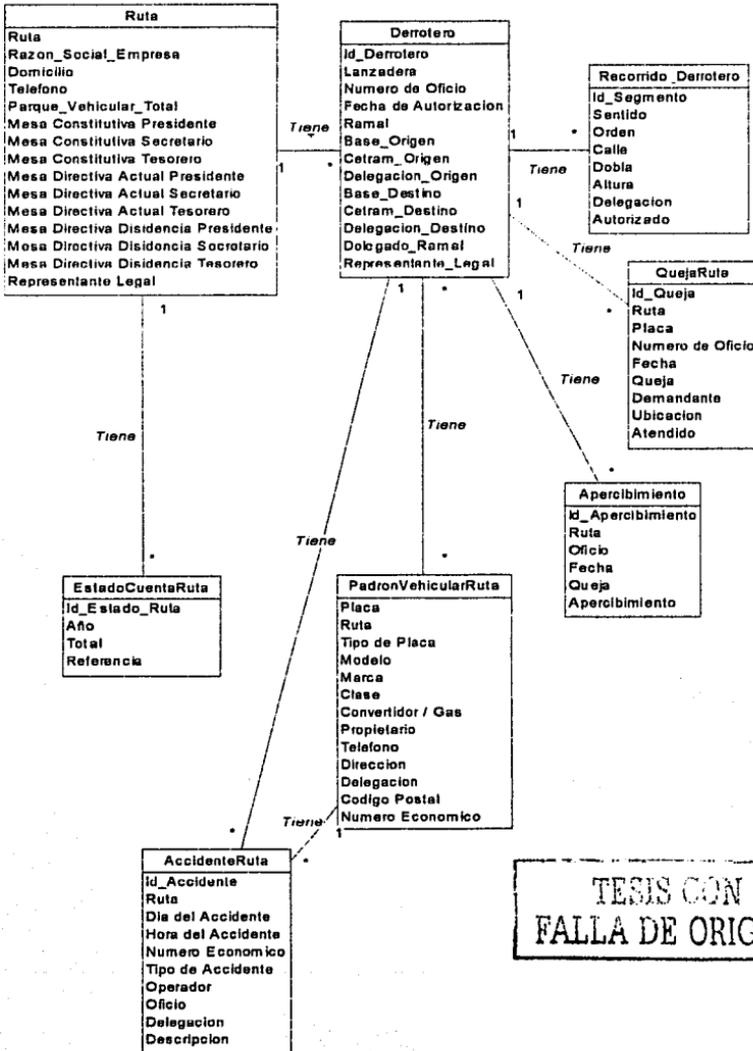
Para poder distinguir entre un concepto y un atributo, podemos emplear la siguiente regla:

*"Si en el mundo real no consideramos algún concepto X como número o texto, probablemente X sea un concepto y no un atributo."*

Lo ideal es relacionar conceptos por medio de una asociación y no por medio de atributos, por ejemplo frecuentemente se agrega un atributo de llave foránea para relacionar dos conceptos, lo cual es un gran error.

Por último cabe resaltar que 'concepto' nos ayuda a designar elementos del mundo real.

En la figura III.2.1.A se tienen las asociaciones y atributos correspondientes a cada concepto de Transporte Colectivo:



TESIS CON FALLA DE ORIGEN

FIG III.2.1.A MODELO CONCEPTUAL

---

### III.3 DISEÑO LÓGICO

Los requerimientos identificados durante la planeación, son el punto principal del diseño lógico; se debe considerar lo siguiente:

- *Flujo de datos:* Entradas y salidas del sistema.
- *Modelo de Datos:* Conjuntos temporales o permanentes de datos (BD).
- *Procesos:* Actividades para aceptar, manejar y suministrar datos e información. Pueden ser manuales o basadas en computadora.
- *Procedimientos:* Métodos y rutinas para utilizar el sistema de información y lograr con ello los resultados esperados.
- *Controles:* Estándares y lineamientos para determinar si las actividades están ocurriendo en la forma anticipada o aceptada, es decir, si se encuentran bajo control. Asimismo especificar las acciones que deben emprenderse cuando ocurren problemas o presentan circunstancias inesperadas. Puede incluirse un reporte sobre las excepciones o procedimientos para la corrección de los problemas.
- *Funciones del personal:* Las responsabilidades de todas las personas que tienen que ver con el nuevo sistemas incluyendo los usuarios, operadores de computadora y personal de apoyo. Abarca todo el espectro de componentes del sistema, incluso desde la entrada de datos hasta la distribución de salidas o resultados. A menudo, las funciones del personal se establecen en forma de procedimiento.

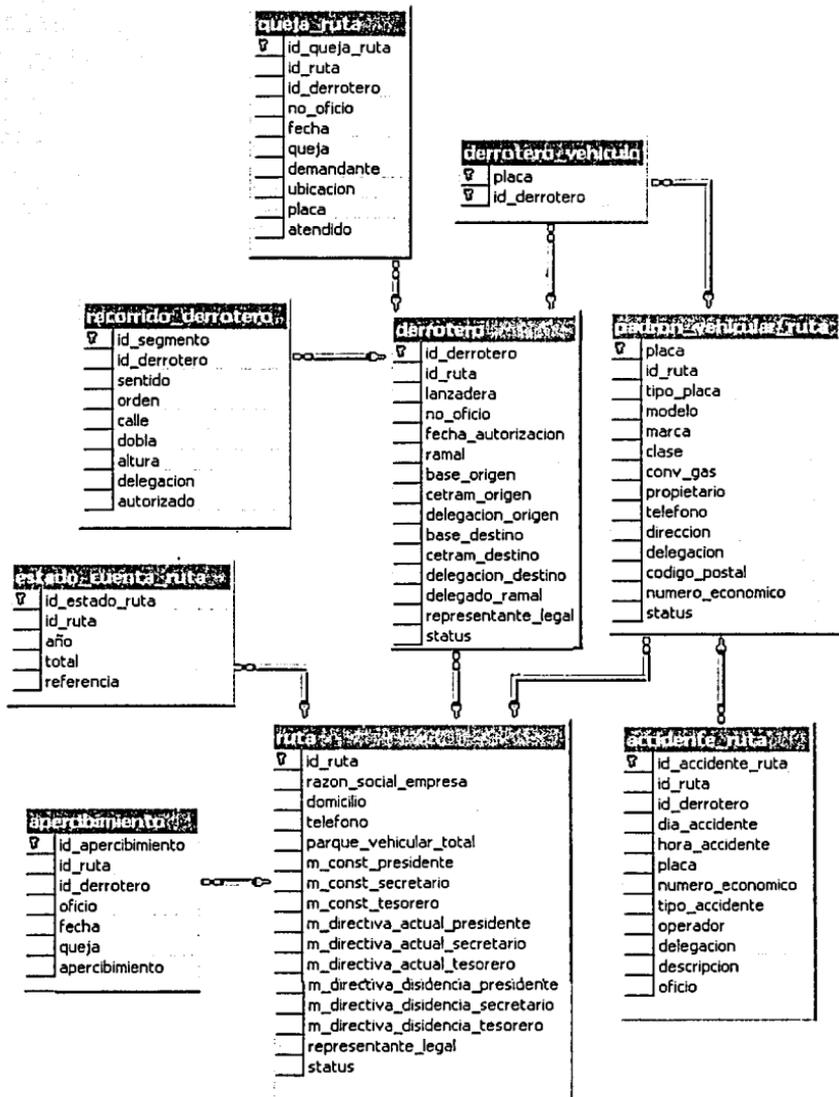
A continuación nos enfocaremos al modelo de datos, obteniendo un esquema lógico, el cual es una descripción de la estructura de la base de datos que usará el software.

Para comenzar se hace un *mapeo* del modelo conceptual, para convertir los conceptos en tablas, los atributos en columnas y las asociaciones a columnas de llaves foráneas.

Es importante señalar, que existe una *asociación de muchos a muchos* entre los conceptos *derrotero* y *padrón\_vehicular\_ruta*, es por ello que se crea una nueva *entidad transitiva* que hereda los identificadores únicos de ambos, construyendo de esta forma una llave compuesta. A la entidad transitiva también se le puede llamar **entidad débil**.

De esta manera se obtiene el **MODELO ENTIDAD-RELACIÓN** que a continuación se presenta:

---



**TESIS CON  
FALLA DE ORIGEN**

FIG III.3.A MODELO ENTIDAD-RELACION

### III.3.1 DIAGRAMA DE CLASES

Este diagrama, describe gráficamente las características de las clases de software en una aplicación, representa la información de asociaciones, atributos, tipos de atributos, métodos, dependencias, etc.

La **clase** es la unidad básica que encapsula toda la información de un objeto, a través de ella podemos modelar el sistema, es representada por un rectángulo dividido en 3 secciones:

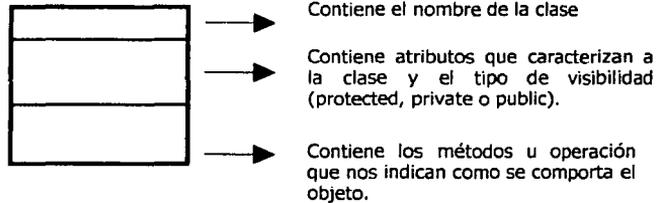


FIG III.3.1.A CLASE

Los **atributos**, dependiendo del grado de comunicación y visibilidad entre ellos y el medio que los rodea, están divididos en tres tipos:

- ◇ *Public*: Es accesible desde todos lados, tanto dentro como fuera de la clase.
- 🔒 *Private*: Sólo es accesible desde dentro de la clase (sólo sus métodos los podrán acceder).
- 🔒 *Protected*: No se puede acceder desde fuera de la clase, pero si por métodos de la clase, así como de subclases.

Los **métodos** son la forma en como interactúa la clase con el medio que la rodea; tienen las siguientes características:

- ◇ *Public*: Indica que el método será visible tanto dentro como fuera de la clase.
- 🔒 *Private*: El método será accesible sólo dentro de la clase.
- 🔒 *Protected*: Puede ser accesado por los métodos de la clase, así como los métodos de las subclases.

TESIS CON  
 FALLA DE ORIGEN

Las **asociaciones** permiten relacionar objetos que colaboran entre sí; en el modelo conceptual ayuda a mejorar la comprensión del dominio del problema, es un modelo analítico; mientras que las asociaciones en el diagrama de clases están orientadas al software.

Por medio de entrevistas y documentos proporcionados, se obtiene la información para agregar a las clases sus *Atributos* y para los *métodos* se toma en cuenta a los diagramas de secuencia y se aplica la siguiente regla, que deberán de cumplir, para ser considerados como métodos: *"el conjunto de mensajes enviados a la clase 1 indican la mayor parte de métodos que deben ser definidos por la clase 1"*.

Se escribe en mayúscula la primer letra de cada nombre de clase, si tal nombre tiene más de una palabra, se une y pone con mayúscula la primera letra de cada palabra, eliminando también los acentos.

Cabe señalar que este diagrama define las clases como entidades de software, por lo tanto es importante para el desarrollo del sistema, en cambio el modelo conceptual representa la abstracción de conceptos del mundo real.

A continuación se puede ver que todos los atributos de las clases son de Tipo Privado ya que sólo serán accesados por su propia clase o en su defecto instanciados por medio de un objeto; los métodos *Agregar*, *Cambiar* y *Eliminar* se especifican como Públicos para ser accesados fuera de la clase, ya que hacen el tratamiento de la información agregando el objeto a la colección o cambiando las propiedades del objeto, y los métodos *Guardar*, *Modificar* y *Borrar* son Privados ya que realizan operaciones con los datos, a nivel base de datos.

El agregar el tipo de atributo, parámetros y valores devueltos del método, depende de a quien se le va a presentar, si el diagrama se esta realizando para la generación automática de código por medio de una Herramienta CASE si será necesario agregar la mayor cantidad de información; si es sólo para el diseño no será necesario saturar el diagrama, por lo que para nuestro caso sólo se mostrarán las clases con los atributos y operaciones correspondientes.



### III.3.2 DICCIONARIO DE DATOS

El **Diccionario de Datos** contiene las características de las entidades y atributos, que definen la estructura de los datos; para que tanto el usuario como el analista tengan una referencia común de todos los elementos del sistema.

Cada entrada en el diccionario consiste de un conjunto de detalles que describen los datos utilizados o producidos por el sistema.

Cada uno está identificado con:

- **Tabla** a la que pertenece dicho elemento.
- **Nombre** para distinguir un dato de otro.
- Si se trata de una **llave Primaria, Secundaria** o sólo es un atributo simple.
- **Descripción** que indica lo que representa en el sistema.
- **Tipo de elemento**, es decir si es un número, cadena de texto, etc.
- **Longitud**, para saber la cantidad de espacio necesario para cada dato.
- **Validación** de los datos, para distinguir que elementos pueden ser omitidos y cuales son obligatorios.

Principales Funciones:

- ✓ Facilitar el control de cada una de las entidades y atributos que forman parte de la base, ya que el sistema puede sufrir cambios continuos.
- ✓ Para documentar las características del sistema, ya que así se tiene una fuente común de información.

Por último, cabe mencionar que no debe existir más de una definición para el mismo dato, así como no incluir elementos que no se utilicen en el modelo, con el propósito de que el diccionario sea completo y consistente.

El Diccionario de Datos para el presente sistema es el siguiente:

Tabla	Nombre	Llave	Descripción	Tipo	Longitud
T_RUTAS	ID_RUTA	LLAVE PRIMARIA	IDENTIFICADOR DE LA RUTA	INT	4
T_RUTAS	RAZON_SOCIAL_EMPRESA		NOMBRE DE LA RAZÓN SOCIAL A LA QUE PERTENECE	VARCHAR	255
T_RUTAS	DOMICILIO		UBICACIÓN DE LA BASE DE LA RUTA	VARCHAR	100
T_RUTAS	TELÉFONO		TELÉFONO DE LOCALIZACIÓN DE LA RUTA	VARCHAR	15
T_RUTAS	PARQUE_VEHICULAR_TOTAL		PARQUE VEHICULAR TOTAL DE LA RUTA	VARCHAR	15
T_RUTAS	M_CONST_PRESIDENTE		NOMBRE DEL PRESIDENTE DE LA MESA CONSTITUTIVA	VARCHAR	40
T_RUTAS	M_CONST_SECRETARIO		NOMBRE DEL SECRETARIO DE LA MESA CONSTITUTIVA	VARCHAR	40
T_RUTAS	M_CONST_TESORERO		NOMBRE DEL TESORERO DE LA MESA CONSTITUTIVA	VARCHAR	40
T_RUTAS	M_DIRECTIVA_ACTUAL_PRESIDENTE		NOMBRE DEL PRESIDENTE DE LA MESA DIRECTIVA ACTUAL	VARCHAR	40
T_RUTAS	M_DIRECTIVA_ACTUAL_SECRETARIO		NOMBRE DEL SECRETARIO DE LA MESA CONSTITUTIVA ACTUAL	VARCHAR	40
T_RUTAS	M_DIRECTIVA_ACTUAL_TESORERO		NOMBRE DEL TESORERO DE LA MESA CONSTITUTIVA ACTUAL	VARCHAR	40
T_RUTAS	M_DIRECTIVA_DISIDENCIA_PRESIDENTE		NOMBRE DEL PRESIDENTE DE LA MESA DIRECTIVA DISIDENCIA	VARCHAR	40
T_RUTAS	M_DIRECTIVA_DISIDENCIA_SECRETARIO		NOMBRE DEL SECRETARIO DE LA MESA DIRECTIVA DISIDENCIA	VARCHAR	40
T_RUTAS	M_DIRECTIVA_DISIDENCIA_TESORERO		NOMBRE DEL TESORERO DE LA MESA DIRECTIVA DISIDENCIA	VARCHAR	40
T_RUTAS	REPRESENTANTE_LEGAL		NOMBRE DEL REPRESENTANTE LEGAL	VARCHAR	40
T_RUTAS	STATUS		ESTADO DE LA RUTA (1 BIT ACTIVO, 0 ARCHIVO HISTÓRICO)	BIT	1
T_PADRON_VEHICULAR_RUTA	PLACA	LLAVE PRIMARIA	PLACAS DEL VEHÍCULO	VARCHAR	10

TESIS CON  
 FALLA DE ORIGEN

T_PADRON_VEHICULAR_RUTA	ID_RUTA	LLAVE FORANEA	RUTAS REGISTRADAS	INT	4
T_PADRON_VEHICULAR_RUTA	TIPO_PLACA		TIPO DE PLACAS	VARCHAR	20
T_PADRON_VEHICULAR_RUTA	MODELO		MODELO DEL VEHICULO REGISTRADO EN EL PADRON	INT	4
T_PADRON_VEHICULAR_RUTA	MARCA		MARCA DEL VEHICULO	VARCHAR	20
T_PADRON_VEHICULAR_RUTA	CLASE		CLASE DEL VEHICULO	VARCHAR	20
T_PADRON_VEHICULAR_RUTA	CONV_GAS		VEHICULOS CONVERTIDOR O GAS	CON VARCHAR	20
T_PADRON_VEHICULAR_RUTA	PROPIETARIO		NOMBRE DEL PROPIETARIO	VARCHAR	40
T_PADRON_VEHICULAR_RUTA	TELEFONO		TELÉFONO LOCALIZACIÓN PROPIETARIO	DE VARCHAR DEL	25
T_PADRON_VEHICULAR_RUTA	DIRECCIÓN		DIRECCIÓN PROPIETARIO	DEL VARCHAR	120
T_PADRON_VEHICULAR_RUTA	DELEGACION		DELEGACIÓN DONDE RESIDE EL PROPIETARIO	VARCHAR	40
T_PADRON_VEHICULAR_RUTA	CÓDIGO_POSTAL		CÓDIGO POSTAL PROPIETARIO	DEL VARCHAR	5
T_PADRON_VEHICULAR_RUTA	NUMERO_ECONOMICO		NÚMERO ECONÓMICO DEL VEHICULO	DEL VARCHAR	10
T_PADRON_VEHICULAR_RUTA	STATUS		ESTADO DEL PADRON VEHICULAR( 1 ACTIVO, 0 ARCHIVO HISTÓRICO)	BIT	1
T_QUEJA_RUTAS	ID_QUEJA_RUTA	LLAVE PRIMARIA	IDENTIFICADOR DE QUEJA	INT/IDENTITY	4
T_QUEJA_RUTAS	ID_RUTA	LLAVE FORANEA	RUTA QUE RECIBE LA QUEJA	INT	4
T_QUEJA_RUTAS	ID_DERROTERO	LLAVE FORANEA	DERROTERO IMPLICADO EN LA QUEJA	INT	4
T_QUEJA_RUTAS	SINO_OFICIO		NÚMERO DEL OFICIO DONDE SE REALIZO LA QUEJA	VARCHAR	20
T_QUEJA_RUTAS	FECHA		FECHA EN QUE SE DIO LA QUEJA	FECHA/HORA	8

**TESIS CCM**  
**FALTA DE ORIGEN**

T_QUEJA_RUTAS	QUEJA		DESCRIPCIÓN DE LA QUEJA	VARCHAR	255
T_QUEJA_RUTAS	DEMANDANTE		NOMBRE DE LA PERSONA QUE REALIZA LA DEMANDA	VARCHAR	40
T_QUEJA_RUTAS	UBICACIÓN		LUGAR POR LA CUAL SE REALIZA LA QUEJA	VARCHAR	40
T_QUEJA_RUTAS	PLACA		PLACAS DEL VEHÍCULO IMPLICADO EN LA QUEJA	VARCHAR	10
T_QUEJA_RUTAS	ATENDIDO		ESTADO DE ATENCIÓN A LA QUEJA (ATENDIDO , NO ATENDIDO)	BIT	1
T_APERCIBIMIENTO_RUTA	ID_APERCIBIMIENTO	LLAVE PRIMARIA	IDENTIFICADOR APERCIBIMIENTO	DEL INT/IDENTITY	4
T_APERCIBIMIENTO_RUTA	ID_RUTA	LLAVE FORANEA	RUTA A LA QUE SE LE APLICA EL APERCIBIMIENTO	INT	4
T_APERCIBIMIENTO_RUTA	ID_DERROTERO	LLAVE FORANEA	DERROTERO IMPLICADO EN EL APERCIBIMIENTO	INT	4
T_APERCIBIMIENTO_RUTA	OFICIO		OFICIO EXPEDIDO EN EL APERCIBIMIENTO	VARCHAR	20
T_APERCIBIMIENTO_RUTA	FECHA		FECHA EN QUE SE REALIZO EL APERCIBIMIENTO	FECHA/HORA	8
T_APERCIBIMIENTO_RUTA	QUEJA		CAUSA POR LA CUAL SE REALIZO APERCIBIMIENTO	VARCHAR UN	255
T_APERCIBIMIENTO_RUTA	APERCIBIMIENTO		DESCRIPCIÓN APERCIBIMIENTO SANSIÓN	DEL VARCHAR O	255
T_ACCIDENTE_RUTA	ID_ACCIDENTE_RUTA	LLAVE PRIMARIA	IDENTIFICADOR ACCIDENTE	DEL INT/IDENTITY	4
T_ACCIDENTE_RUTA	ID_RUTA		RUTA INVOLUCRADA EN EL ACCIDENTE	INT	4
T_ACCIDENTE_RUTA	ID_DERROTERO	LLAVE FORANEA	DERROTERO INVOLUCRADO EN EL ACCIDENTE	INT	4
T_ACCIDENTE_RUTA	DIA_ACCIDENTE		DÍA EN QUE OCURRE EL ACCIDENTE	FECHA/HORA	8
T_ACCIDENTE_RUTA	HORA_ACCIDENTE		HORA EN LA QUE OCURRE EL ACCIDENTE	VARCHAR	5
T_ACCIDENTE_RUTA	PLACA		PLACAS DEL VEHÍCULO IMPLICADO EN EL ACCIDENTE	VARCHAR	10
T_ACCIDENTE_RUTA	NÚMERO_ECONÓMICO		NÚMERO CON QUE IDENTIFICAN EL VEHÍCULO	VARCHAR	10

**TESIS CON  
FALLA DE ORIGEN**

T_ACCIDENTE_RUTA	TIPO_ACCIDENTE		TIPO DE ACCIDENTE	VARCHAR	3
T_ACCIDENTE_RUTA	OPERADOR		NOMBRE DEL OPERADOR INVOLUCRADO EN EL ACCIDENTE	VARCHAR	40
T_ACCIDENTE_RUTA	DELEGACIÓN		NOMBRE DE LA DELEGACIÓN EN DONDE OCURRE EL ACCIDENTE	VARCHAR	40
T_ACCIDENTE_RUTA	DESCRIPCION		DESCRIPCIÓN DEL ACCIDENTE	VARCHAR	255
T_ACCIDENTE_RUTA	OFICIO		OFICIO EXPEDIDO EN EL ACCIDENTE	VARCHAR	10
T_ESTADO_CUENTA_RUTA	ID_ESTADO_RUTA	LLAVE PRIMARIA	IDENTIFICADOR DEL ESTADO DE CUENTA DE LA RUTA	IDENTITY	4
T_ESTADO_CUENTA_RUTA	ID_RUTA	LLAVE FORANEA	RUTAS EXISTENTES	INT	4
T_ESTADO_CUENTA_RUTA	AÑO		AÑO DEL ESTADO DE CUENTA	INT	4
T_ESTADO_CUENTA_RUTA	TOTAL		TOTAL ADEUDO POR CADA RUTA	MONEY	8
T_ESTADO_CUENTA_RUTA	REFERENCIA		REFERENCIA DEL ESTADO DE CUENTA	VARCHAR	20
T_DERROTERO_VEHÍCULO	PLACA		IDENTIFICADOR DEL VEHÍCULO	VARCHAR	10
T_DERROTERO_VEHÍCULO	ID_DERROTERO		IDENTIFICADOR DEL DERROTERO	INT	4
T_DERROTERO	ID_DERROTERO	LLAVE PRIMARIA	IDENTIFICADOR DEL DERROTERO	IDENTITY	4
T_DERROTERO	ID_RUTA	LLAVE FORANEA	RUTA A LA QUE PERTENECE EL DERROTERO	INT	4
T_DERROTERO	LANZADERA		LUGAR DONDE ESPERA SU SALIDA, EL DERROTERO	VARCHAR	255
T_DERROTERO	NO_OFICIO		NÚMERO OFICIO CUANDO SE REGISTRA EL DERROTERO	VARCHAR	20
T_DERROTERO	FECHA_AUTORIZACION		FECHA EN QUE SE AUTORIZO EL DERROTERO	FECHA/HORA	8
T_DERROTERO	RAMAL		NOMBRE DE LOS LUGARES DE SALIDA Y LLEGADA DEL DERROTERO	VARCHAR	255
T_DERROTERO	BASE ORIGEN		NOMBRE DE LA BASE DONDE SALE EL DERROTERO	VARCHAR	255
T_DERROTERO	CETRAM_ORIGEN		NOMBRE DEL CETRAM DONDE SALE EL DERROTERO	DEBIT	1

**TESIS CON  
FALTA DE ORIGEN**

T_DERROTERO	DELEGACION_ORIGEN		NOMBRE DE LA DELEGACION DONDE SALE EL DERROTERO	VARCHAR	40
T_DERROTERO	BASE DESTINO		NOMBRE DE LA BASE DONDE LLEGA EL DERROTERO	VARCHAR	255
T_DERROTERO	CETRAM_DESTINO		NOMBRE DEL CETRAM DONDE LLEGA EL DERROTERO	ABIT	1
T_DERROTERO	DELEGACION_DESTINO		NOMBRE DE LA DELEGACION A DONDE LLEGA EL DERROTERO	VARCHAR	40
T_DERROTERO	DELEGADO_RAMAL		NOMBRE DEL DELEGADO DEL RAMAL	VARCHAR	60
T_DERROTERO	REPRESENTANTE_LEGAL		NOMBRE DEL REPRESENTANTE LEGAL	VARCHAR	40
T_DERROTERO	STATUS		ESTADO DEL DERROTERO (1 BIT ACTIVO, 0 ARCHIVO HISTORICO)		1
T_RECORRIDO_DERROTERO	ID_SEGMENTS	LLAVE PRIMARIA	IDENTIFICADOR DEL RECORRIDO DERROTERO	INT/IDENTITY	4
T_RECORRIDO_DERROTERO	ID_DERROTERO	LLAVE FORANEA	DERROTERO AL QUE LE PERTENECE EL RECORRIDO	INT	4
T_RECORRIDO_DERROTERO	SENTIDO		TOPOLOGIA USADA PARA DESCRIBIR EL RECORRIDO	VARCHAR	4
T_RECORRIDO_DERROTERO	ORDEN		NUMERO CONSECUTIVO PARA EL ORDEN DEL RECORRIDO	VARCHAR	2
T_RECORRIDO_DERROTERO	CALLE		NOMBRE DE LAS CALLES UTILIZADAS EN EL RECORRIDO DERROTERO	VARCHAR	60
T_RECORRIDO_DERROTERO	DOBLA		DIRECCION EN EL QUE DOBLA EL DERROTERO	VARCHAR	15
T_RECORRIDO_DERROTERO	ALTURA		CALLE REFERENCIA CUANDO SE DA UNA VUELTA EN 'U'	VARCHAR	60
T_RECORRIDO_DERROTERO	DELEGACION		NOMBRE DE LA DELEGACION DONDE SE LLEVA A CABO EL RECORRIDO	VARCHAR	40
T_RECORRIDO_DERROTERO	AUTORIZADO		ESPECIFICACION DE LA AUTORIZACION DEL RECORRIDO DERROTERO (S/N)	VARCHAR	1

TESIS CON  
FALLA DE ORIGEN

### III.3.3 MODELOS DE COMPORTAMIENTO DEL SISTEMA

Los diagramas de secuencia y contratos ayudan a determinar el comportamiento del sistema, el cual se define como la descripción de lo que hace el sistema, sin explicar la manera en que lo hace.

#### III.3.3.1 Diagrama de Secuencia del Sistema

El **diagrama de secuencia** muestra gráficamente los eventos que trascienden las fronteras del sistema y que fluyen de los actores al sistema. Están basados fundamentalmente en los casos de uso, ya que muestra la interacción de los actores con el sistema, durante la interacción un actor genera eventos dirigidos al sistema, solicitando alguna operación a cambio.

El **evento** es un hecho externo que un actor produce, éste da origen a una operación de respuesta; la cual es la acción que el sistema ejecuta en respuesta a dicho evento.

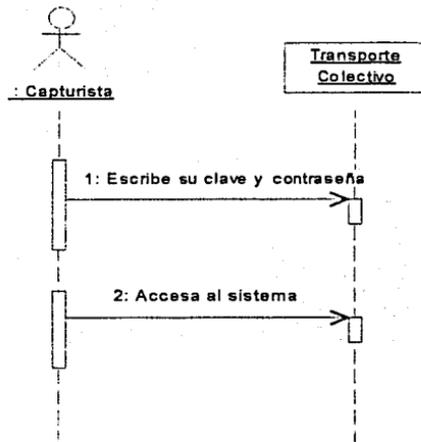
Para mejorar la notación, el nombre del evento debe comenzar preferentemente con un verbo, porque reafirma que los eventos están orientados a comandos.

El diagrama consta de actores que operan directamente sobre el sistema, eventos generados por los actores y el sistema al que van dirigidos (a los sistemas se les trata como caja negra), el tiempo avanza hacia abajo y es representado con una línea vertical.

Los diagramas están formados por:

- **Actor:** Es quien hace la petición para realizar alguna operación.
- **Objeto:** El rectángulo representa una instancia de un objeto en particular y la línea punteada representa las llamadas a métodos del objeto.
- **Mensaje a otro objeto:** Se representa por una flecha entre un objeto y otro, es decir, la llamada de un método (operación) en particular.
- **Mensaje al mismo objeto:** Es posible visualizar llamadas a métodos desde el mismo objeto.

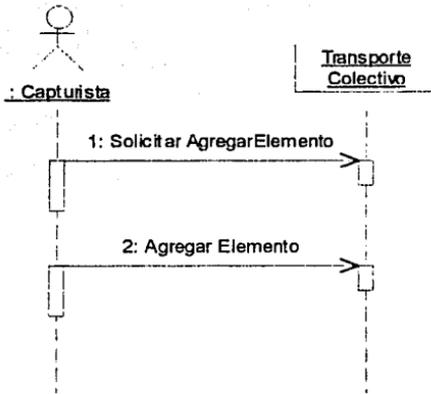
Inicialmente se muestra el diagrama de secuencia del caso de uso *Verificando Usuario*.



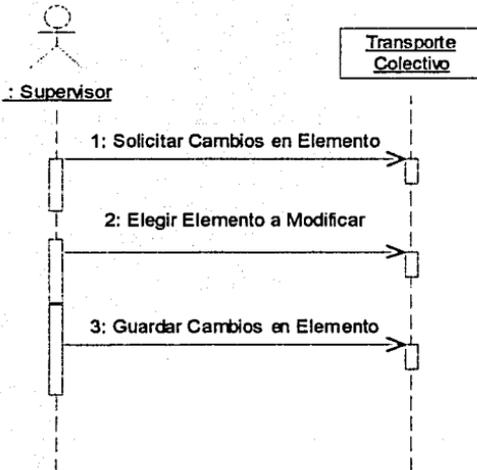
A continuación se muestran los diagramas de secuencia del sistema, para las opciones de Agregar, Cambiar y Eliminar, sólo es necesario sustituir la palabra *Elemento* por cualquiera de las siguientes opciones:



TESIS CON  
FALLA DE ORIGEN



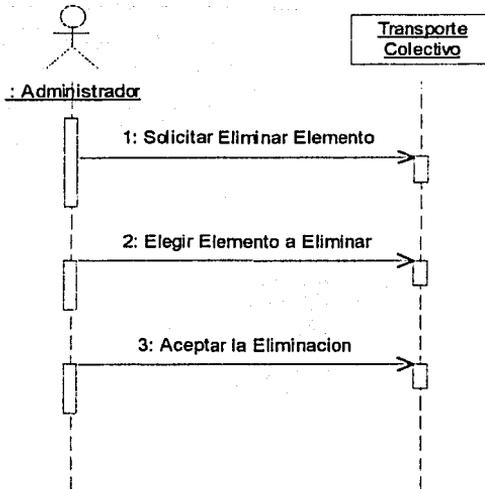
Puede repetirse las veces que sea necesario, obviamente con registros diferentes.



Puede repetirse las veces que sea necesario modificar un Elemento.

**TESIS CON FALLA DE ORIGEN**

ESTA TESIS NO SALE DE LA BIBLIOTECA



Puede repetirse las veces que sea necesario, obviamente con registros diferentes.

### III.3.3.2 Contratos (Threads)

Es un documento que describe lo que la operación se propone lograr; el efecto que sobre el sistema tienen las operaciones, es decir ayuda a definir el comportamiento del sistema, sólo indica lo que hará y no como lo conseguirá.

Su elaboración depende del modelo conceptual, diagramas de secuencia del sistema y la identificación de las operaciones.

A continuación se analizan los contratos realizados para las operaciones de Agregar, Cambiar, Eliminar.

Las **precondiciones** indican el estado en el que se encuentra el sistema antes de ejecutar la operación.

Las **poscondiciones** indican como cambio el sistema después de realizar una operación.

TESIS CON  
FALLA DE ORIGEN

<b>Nombre</b>	<b>AgregarRuta</b>
Responsabilidades	Registrar una ruta y guardarla en el sistema.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.2, R1.6 (Pág 38) Casos de Uso: Agregando Ruta
Precondiciones	Deben estar capturados todos los registros.
Poscondiciones	Fue creada y guardada una nueva ruta en el sistema.

<b>Nombre</b>	<b>AgregarDerroteros Ruta</b>
Responsabilidades	Registrar un derrotero y guardarlo en el sistema.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.2, R1.6 (Pág 38) Casos de Uso: Agregando Derrotero
Excepciones	Si no se especifica la ruta indicar un error.
Precondiciones	Deben estar capturados todos los registros.
Poscondiciones	Fue creado y agregado un nuevo derrotero al sistema. Además fue creada una asociación con Ruta.

<b>Nombre</b>	<b>AgregarRecorridoDerroteros</b>
Responsabilidades	Registrar todo el recorrido de un derrotero y guardarlo en el sistema..
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.2, R1.6 (Pág 38) Casos de Uso: Agregando Recorrido_Derrotero
Excepciones	
Precondiciones	Debe existir el derrotero y estar capturados todos los registros.
Poscondiciones	Fue creado y agregado el recorrido de un derrotero al sistema. También fue creada una asociación con Derrotero.

<b>Nombre</b>	<b>AgregarEstadoCtaRuta</b>
Responsabilidades	Registrar el estado de cuenta de una ruta y guardarla en el sistema.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.2, R1.6 (Pág 38) Casos de Uso: Agregando EstadoCuentaRuta
Excepciones	Si no se especifica la ruta indicar un error. Si las cantidades numéricas no son validas indicar error.
Precondiciones	Deben estar capturados todos los registros.

Poscondiciones	Fue creado y agregado un nuevo estado de cuenta de una ruta en el sistema. Además fue creada una asociación con Ruta.
----------------	--

<b>Nombre</b>	<b>AgregarApercibimiento_Ruta</b>
Responsabilidades	Registrar un apercibimiento y guardarlo en el sistema.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.2, R1.6 (Pág 38)  Casos de Uso: Agregando Apercibimiento
Excepciones	Si no se especifica la ruta y el derrotero indicar un error.
Precondiciones	Deben estar capturados todos los registros.
Poscondiciones	Fue creado y agregado un nuevo apercibimiento al sistema. Fue creada una asociación con Ruta y Derrotero.
<b>Nombre</b>	<b>AgregarPadron_Vehicular_Ruta</b>
Responsabilidades	Registrar un vehículo y guardarlo en el sistema.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.2, R1.6 (Pág 38)  Casos de Uso: Agregando PadronVehicularRuta
Excepciones	Si no se especifica la ruta y el derrotero indicar un error.
Precondiciones	Deben estar capturados todos los registros.
Poscondiciones	Fue creado y agregado un nuevo vehículo al sistema. También fue creada una asociación con Ruta y Derrotero.

<b>Nombre</b>	<b>AgregarAccidentes_Ruta</b>
Responsabilidades	Registrar un accidente y guardarlo en el sistema.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.2, R1.6 (Pág 38)  Casos de Uso: Agregando AccidenteRuta
Excepciones	Si no se especifica la ruta, el derrotero y las placas indicar un error.
Precondiciones	Deben estar capturados todos los registros obligatorios.
Poscondiciones	Fue creado y agregado un nuevo accidente al sistema. Además fue creada una asociación con Ruta, Derrotero y PadronVehicularRuta.

<b>Nombre</b>	<b>AgregarQuejaRuta</b>
Responsabilidades	Registrar una queja y guardarla en el sistema.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.2, R1.6 (Pág 38)
Excepciones	Casos de Uso: Agregando QuejaRuta Si no se especifica la ruta y el derrotero indicar un error.
Precondiciones	Deben estar capturados todos los registros.
Poscondiciones	Fue creada y agregada una nueva queja al sistema. Se creo una asociación con Ruta y Derrotero.

<b>Nombre</b>	<b>CambiarRuta</b>
Responsabilidades	Permitir realizar modificaciones en los atributos pertenecientes a una ruta, así como guardar dichos cambios.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.3, R1.6 (Pág 38)
Excepciones	Casos de Uso: Cambiando Ruta
Precondiciones	Elegir el registro al cual se le va a realizar alguna modificación.
Poscondiciones	Fueron modificados los atributos deseados y se guardaron los cambios.

<b>Nombre</b>	<b>CambiarDerrotero</b>
Responsabilidades	Permitir realizar modificaciones en los atributos pertenecientes a derrotero, así como guardar dichos cambios.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.3, R1.6 (Pág 38)
Excepciones	Casos de Uso: Cambiando Derrotero
Precondiciones	Elegir el registro al cual se le va a realizar alguna modificación.
Poscondiciones	Fueron modificados los atributos deseados y se guardaron los cambios.

<b>Nombre</b>	<b>CambiarRecorridoDerrotero</b>
Responsabilidades	Permitir realizar modificaciones en los atributos pertenecientes al recorrido de un derrotero, así como guardar dichos cambios.
Tipo	Clase de Software

Referencias Cruzadas	Funciones del sistema: R1.1, R1.3, R1.6 (Pág 38) Casos de Uso: Cambiando Recorrido_Derrotero
Excepciones	
Precondiciones	Elegir el registro al cual se le va a realizar alguna modificación.
Poscondiciones	Fueron modificados los atributos deseados y se guardaron los cambios.

<b>Nombre</b>	<b>CambiarEdoCuentaRuta</b>
Responsabilidades	Permitir realizar modificaciones en los atributos pertenecientes a estado de cuenta, así como guardar dichos cambios.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.3, R1.6 (Pág 38) Casos de Uso: Cambiando EstadoCuentaRuta
Excepciones	Si las cantidades numéricas no son validas indicar error.
Precondiciones	Elegir el registro al cual se le va a realizar alguna modificación.
Poscondiciones	Fueron modificados los atributos deseados y se guardaron los cambios.

<b>Nombre</b>	<b>CambiarApercibimientoRuta</b>
Responsabilidades	Permitir realizar modificaciones en los atributos pertenecientes a apercibimiento, así como guardar dichos cambios.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.3, R1.6 (Pág 38) Casos de Uso: Cambiando Apercibimiento
Excepciones	Si la ruta o derrotero no son validos indicar error.
Precondiciones	Elegir el registro al cual se le va a realizar alguna modificación.
Poscondiciones	Fueron modificados los atributos deseados y se guardaron los cambios. Existe la posibilidad de haber cancelado o formado una nueva asociación con Ruta o Derrotero. Si es que se modificaron dichos atributos.

<b>Nombre</b>	<b>CambiarPadronRuta</b>
Responsabilidades	Permitir realizar modificaciones en los atributos pertenecientes al padrón vehicular, así como guardar dichos cambios.
Tipo	Clase de Software

Referencias Cruzadas	Funciones del sistema: R1.1, R1.3, R1.6 (Pág 38)  Casos de Uso: Cambiando PadronVehicularRuta
Excepciones	Si la ruta, derrotero o placas no son validos indicar error.
Precondiciones	Elegir el registro al cual se le va a realizar alguna modificación.
Poscondiciones	Fueron modificados los atributos deseados y se guardaron los cambios. Existe la posibilidad de haber cancelado o formado una nueva asociación con Ruta ó Derrotero; si es que se modificaron dichos atributos.

<b>Nombre</b>	<b>CambiarAccidenteRuta</b>
Responsabilidades	Permitir realizar modificaciones en los atributos pertenecientes a accidentes, así como guardar dichos cambios.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.3, R1.6 (Pág 38)  Casos de Uso: Cambiando AccidenteRuta
Excepciones	Si la ruta, derrotero o placas no son validos indicar error.
Precondiciones	Elegir el registro al cual se le va a realizar alguna modificación.
Poscondiciones	Fueron modificados los atributos deseados y se guardaron los cambios. Existe la posibilidad de haber cancelado o formado una nueva asociación con Ruta, Derrotero o PadronVehicular.

<b>Nombre</b>	<b>CambiarQuejaRuta</b>
Responsabilidades	Permitir realizar modificaciones en los atributos pertenecientes a una queja, así como guardar dichos cambios.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.3, R1.6 (Pág 38)  Casos de Uso: Cambiando QuejaRuta
Excepciones	Si la ruta o derrotero no son validos indicar error.
Precondiciones	Elegir el registro al cual se le va a realizar alguna modificación.
Poscondiciones	Fueron modificados los atributos deseados y se guardaron los cambios. Existe la posibilidad de haber cancelado o formado una nueva asociación con Ruta o Derrotero.

<b>Nombre</b>	<b>EliminarRuta</b>
Responsabilidades	Eliminar determinada ruta.
Tipo	Clase de Software

Referencias Cruzadas	Funciones del sistema: R1.1, R1.5 (Pág 38) Casos de Uso: Eliminando Ruta
Excepciones	
Precondiciones	Elegir la ruta que se desea eliminar.
Poscondiciones	Fue eliminada una ruta (sólo se cambio su estado a NoVigente)

<b>Nombre</b>	<b>EliminarDerrotero</b>
Responsabilidades	Eliminar un derrotero específico.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.5 (Pág 38) Casos de Uso: Eliminando Derrotero
Excepciones	
Precondiciones	Elegir el derrotero a eliminar.
Poscondiciones	Fue eliminado el derrotero (sólo se cambio su estado a NoVigente). Es cancelada su asociación con Ruta.

<b>Nombre</b>	<b>EliminarEstadoCuentaRuta</b>
Responsabilidades	Eliminar un estado de cuenta específico.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.4 (Pág 38) Casos de Uso: Eliminando EstadoCuentaRuta
Excepciones	
Precondiciones	Elegir el registro a eliminar.
Poscondiciones	Fue eliminado un registro y cancelada su asociación con Ruta.

<b>Nombre</b>	<b>EliminarApercibimiento</b>
Responsabilidades	Eliminar un determinado Apercibimiento
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.4 (Pág 38) Casos de Uso: Eliminando Apercibimiento
Excepciones	
Precondiciones	Elegir el registro a eliminar.
Poscondiciones	Fue eliminado un apercibimiento y cancelada su asociación con Ruta y Derrotero.

<b>Nombre</b>	<b>EliminarPadronVehicularRuta</b>
Responsabilidades	Eliminar un determinado vehículo
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.5 (Pág 38)

	<b>Casos de Uso: Eliminando PadronVehicularRuta</b>
Excepciones	
Precondiciones	Elegir el registro a eliminar.
Poscondiciones	Fue eliminado un vehículo (sólo se cambio su estado a NoVigente). Es cancelada su asociación con Ruta y Derrotero.

<b>Nombre</b>	<b>EliminarAccidenteRuta</b>
Responsabilidades	Eliminar un determinado accidente.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.4 (Pág 38)  Casos de Uso: Eliminando AccidenteRuta
Excepciones	
Precondiciones	Elegir el registro a eliminar.
Poscondiciones	Fue eliminado un accidente y cancelada su asociación con Ruta, Derrotero y PadronVehicularRuta.

<b>Nombre</b>	<b>EliminarQuejaRuta</b>
Responsabilidades	Eliminar una queja.
Tipo	Clase de Software
Referencias Cruzadas	Funciones del sistema: R1.1, R1.4 (Pág 38)  Casos de Uso: Eliminando QuejaRuta
Excepciones	
Precondiciones	Elegir el registro a eliminar.
Poscondiciones	Fue eliminado un registro y cancelada su asociación con Ruta y Derrotero.

---

### III.4 DISEÑO FÍSICO Y CONSTRUCCIÓN DEL SISTEMA

Se obtiene un esquema físico que hace la descripción de la construcción y elaboración de la base de datos y del programa de aplicación, las estructuras de almacenamiento y los métodos usados para tener acceso a los datos.

Para iniciar la construcción del sistema, se toma en cuenta que la aplicación es nueva, por lo tanto requiere de la creación de una base de datos que deberá implementarse antes de instalar los programas de aplicación, ya que éstos harán uso de ella. Además esta aplicación requiere de una red de área local, la cual ya ha sido instalada con anterioridad por la dependencia, teniendo así una arquitectura Cliente-Servidor.

#### III.4.1 ARQUITECTURA CLIENTE/ SERVIDOR

Para aclarar las posibles dudas acerca de la **Arquitectura Cliente/ Servidor**, a continuación se describen algunas características, aspectos importantes, ventajas, así como desventajas, que llevan a justificar la arquitectura cliente / servidor, como la base fundamental en el desarrollo del presente sistema.

La *arquitectura cliente / servidor* es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

Esta arquitectura se distingue porque la aplicación y la base de datos están separadas en 2 partes:

- ❖ **Cliente** ejecuta la aplicación que accede a la información de la base de datos(servidor) e interactúa con el usuario; proceso que inicia el diálogo o solicita los recursos. Realiza las siguientes funciones:
  - Maneja la interfaz del usuario.
  - Captura y validación de los datos de entrada.
  - Generación de consultas e informes sobre las bases de datos.
- ❖ **Servidor** ejecuta el software del DBMS y maneja las funciones requeridas para permitir el acceso concurrente a la base de datos; Realiza las siguientes funciones:
  - Administración de periféricos compartidos.

- o Control de accesos concurrentes a bases de datos compartidas.
- o Enlaces de comunicaciones con otras redes de área local o extensa.
- o Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le responde proporcionándolo.

Aunque la aplicación y el SGBD pueden ejecutarse en la misma máquina, será más efectivo si la parte del cliente y la del servidor se ejecutan en diferentes nodos, conectados a través de la red. (Nodo es cualquier máquina del sistema (cliente, servidor o ambos)).

Características de la arquitectura cliente / servidor:

- o El servidor presenta a todos sus clientes una interfaz única y bien definida.
- o El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- o El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- o Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas de la arquitectura cliente / servidor:

- ✓ Reduce los costos de producción de software y disminuye los tiempos requeridos, ya que para la construcción de una nueva aplicación pueden usarse los servidores que hay disponibles.
- ✓ Proporciona una mayor consistencia a la información de la empresa al contar con un control centralizado.
- ✓ Permite llevar más fácilmente la información a donde se necesita, además de que contribuye a aumentar su precisión, pues se puede obtener de su fuente (el servidor) y no de una copia en papel, etc.
- ✓ Los clientes y los servidores pueden existir en múltiples plataformas y tener acceso a datos de cualquier sitio en la red.
- ✓ Contribuye a una disminución de los costos de capacitación del personal, pues favorecen la construcción de interfaces gráficas interactivas, las cuales

---

son más fáciles de usar por el usuario final, teniendo como consecuencia el aumento de su productividad.

Desventajas de la arquitectura cliente / servidor:

- × El mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- × Los problemas de congestión de la red pueden degradar el rendimiento del sistema por debajo de lo que se obtendría con una única máquina (arquitectura centralizada).

Al analizar lo anterior se comprueba que el número de ventajas que presenta es superior a las desventajas; en este caso no se tendría problemas con el mantenimiento, ya que todo el equipo de cómputo con que se cuenta es de la misma marca, por lo tanto se evitan incompatibilidades entre sí.

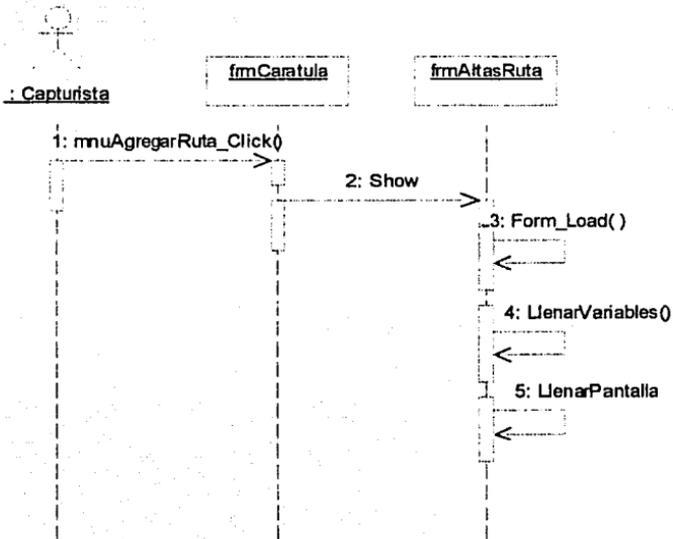
### **III.4.2 DIAGRAMAS DE SECUENCIA**

En el diseño lógico se usaron diagramas de secuencia para describir lo que hace el sistema, su comportamiento en general.

A continuación, se anexan algunos diagramas de secuencia que muestran el proceso para llevar a cabo una operación, desde la solicitud de formularios hasta la forma en que se realizan las operaciones del sistema como: agregar, cambiar y eliminar.

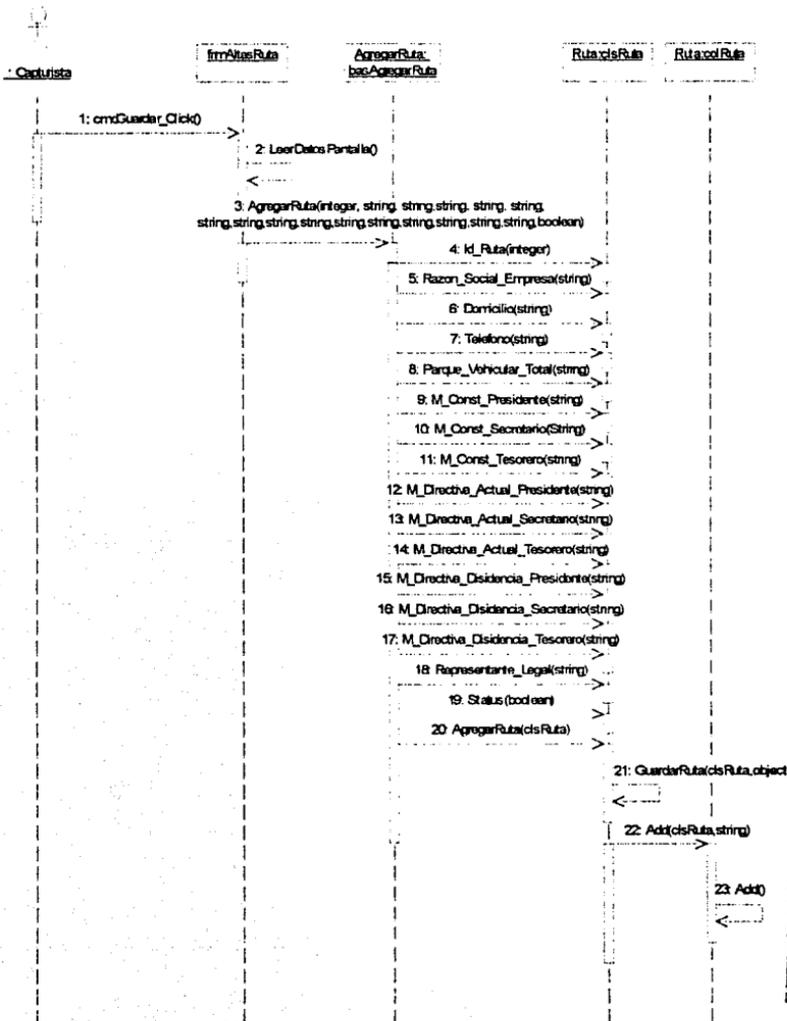
Es importante recordar que el diagrama de secuencia representa la forma en como un cliente (actor) u objetos (clases) se comunican entre sí en petición a un evento.

## 1. SOLICITAR AGREGAR RUTA



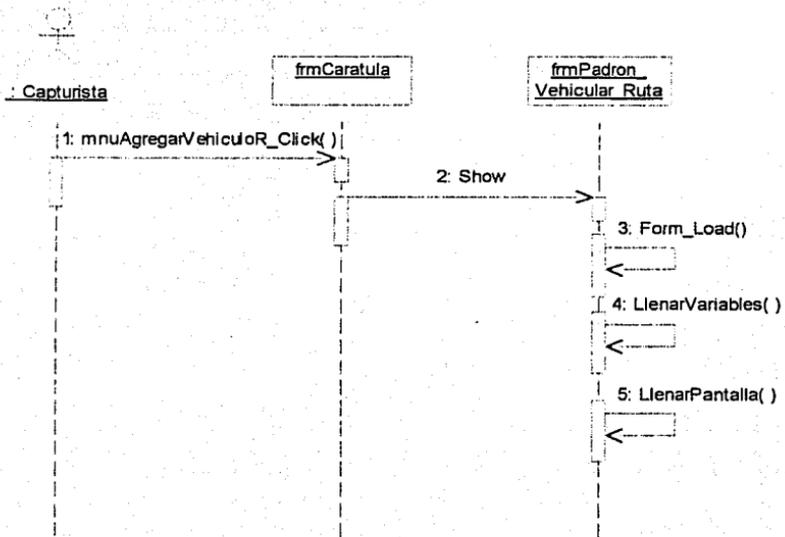
TESIS CON  
FALLA DE ORIGEN

2. AGREGAR NUEVA RUTA



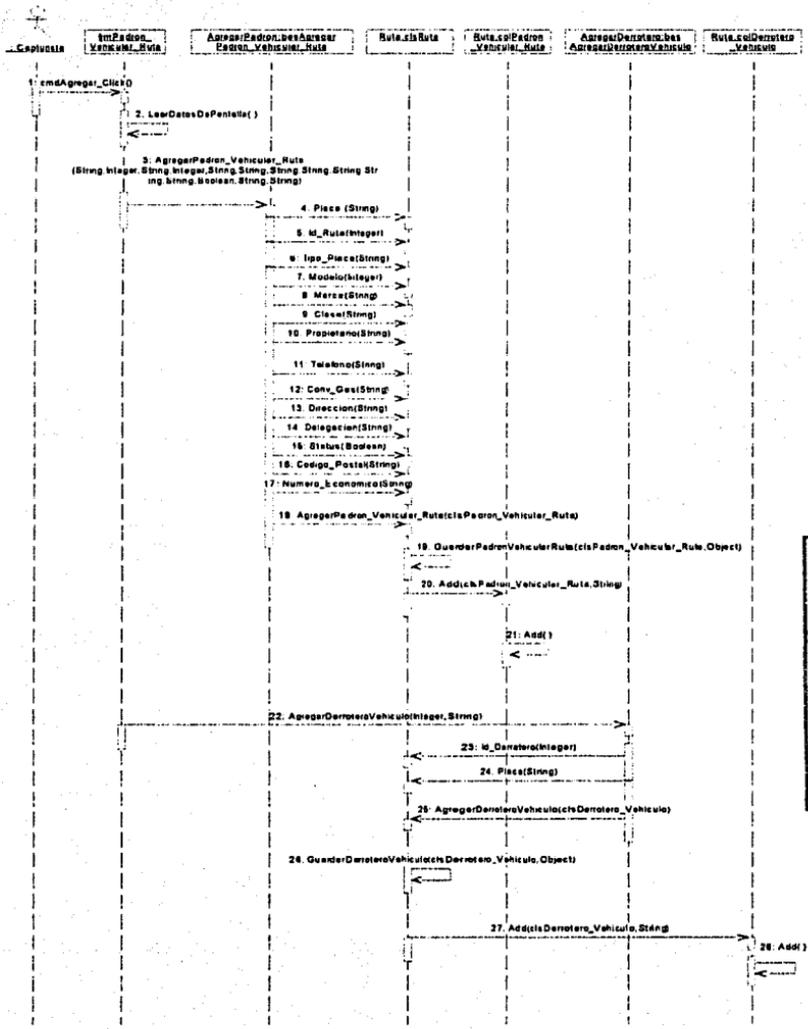
**TESIS CON  
FALLA DE ORIGEN**

## 3. SOLICITAR AGREGAR VEHÍCULO



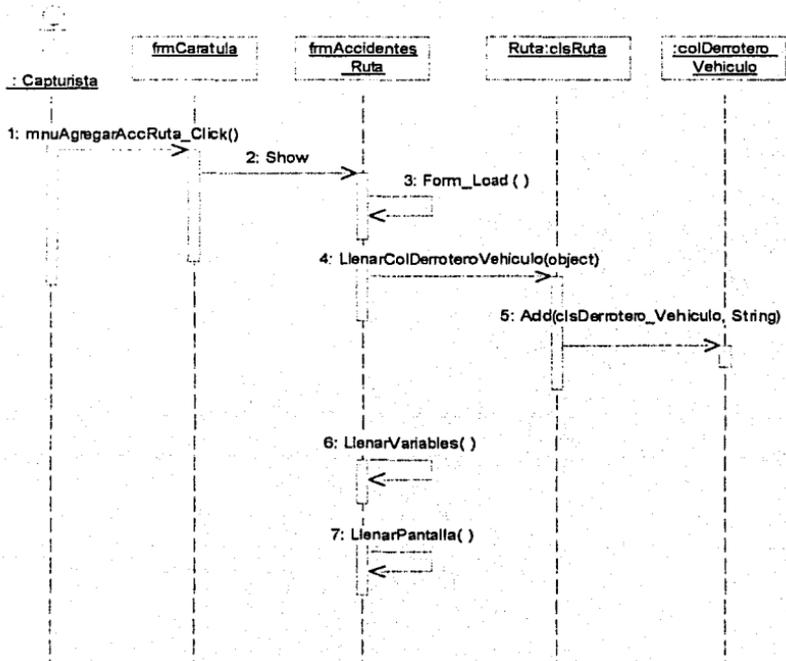
TESIS CON  
FALLA DE ORIGEN

### 4. AGREGAR NUEVO VEHÍCULO



**TESIS CON  
FALLA DE ORIGEN**

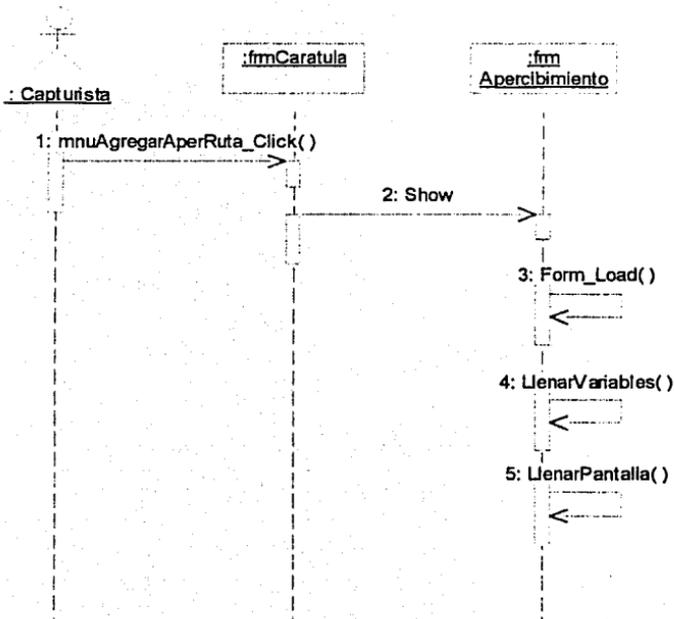
## 5. SOLICITAR AGREGAR ACCIDENTE



TESIS CON  
FALLA DE ORIGEN

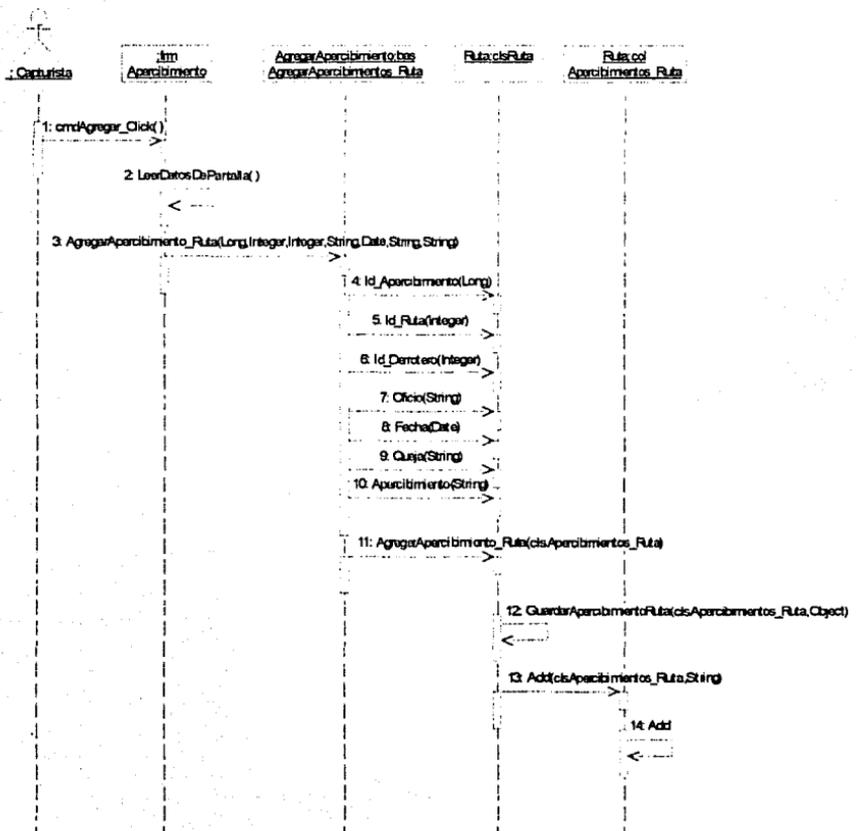


## 7. SOLICITAR AGREGAR APERCIBIMIENTO



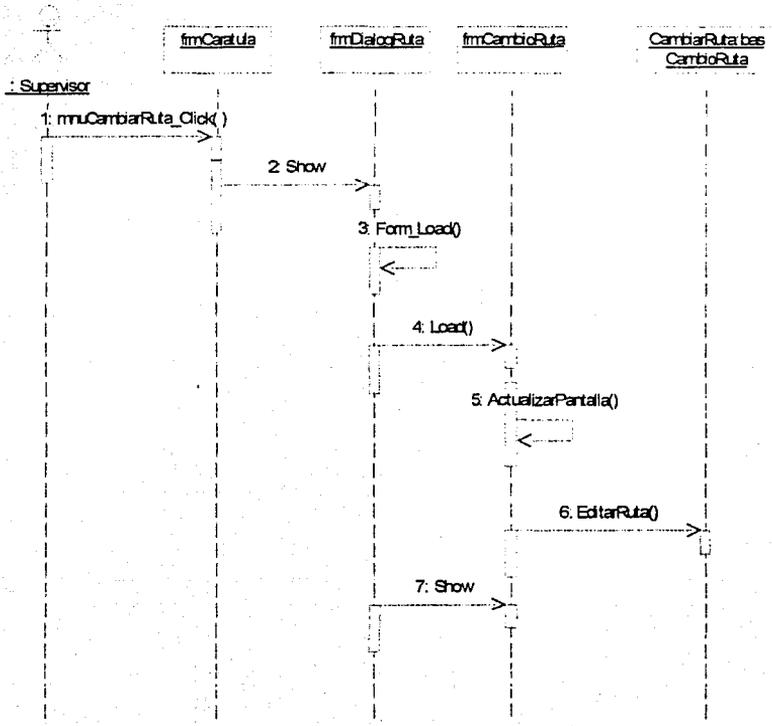
TESIS CON  
FALLA DE ORIGEN

## 8. AGREGAR NUEVO APERCIBIMIENTO



TESIS CON  
FALLA DE ORIGEN

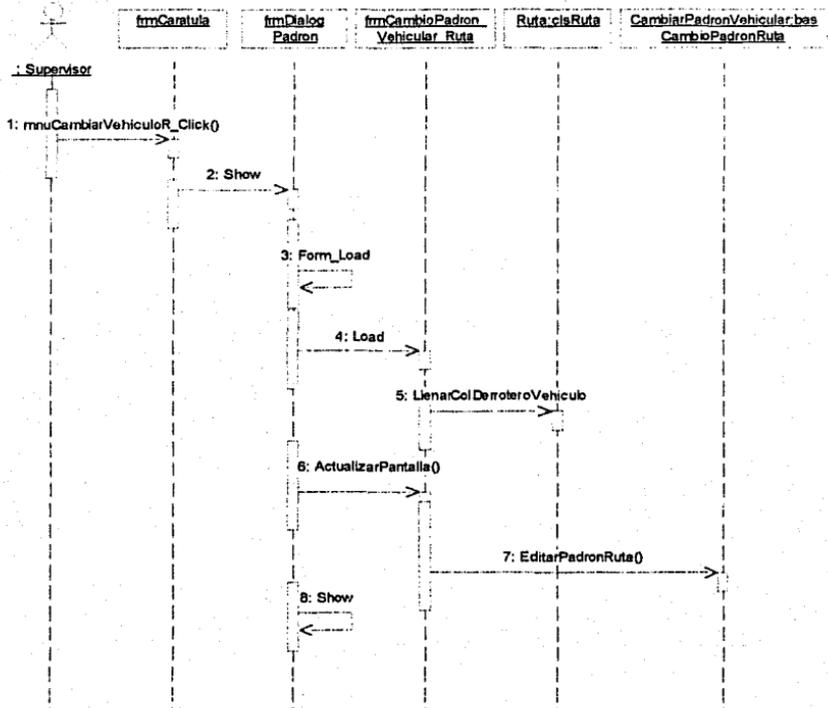
9. SOLICITAR CAMBIAR RUTA



TESIS CON FALLA DE ORIGEN

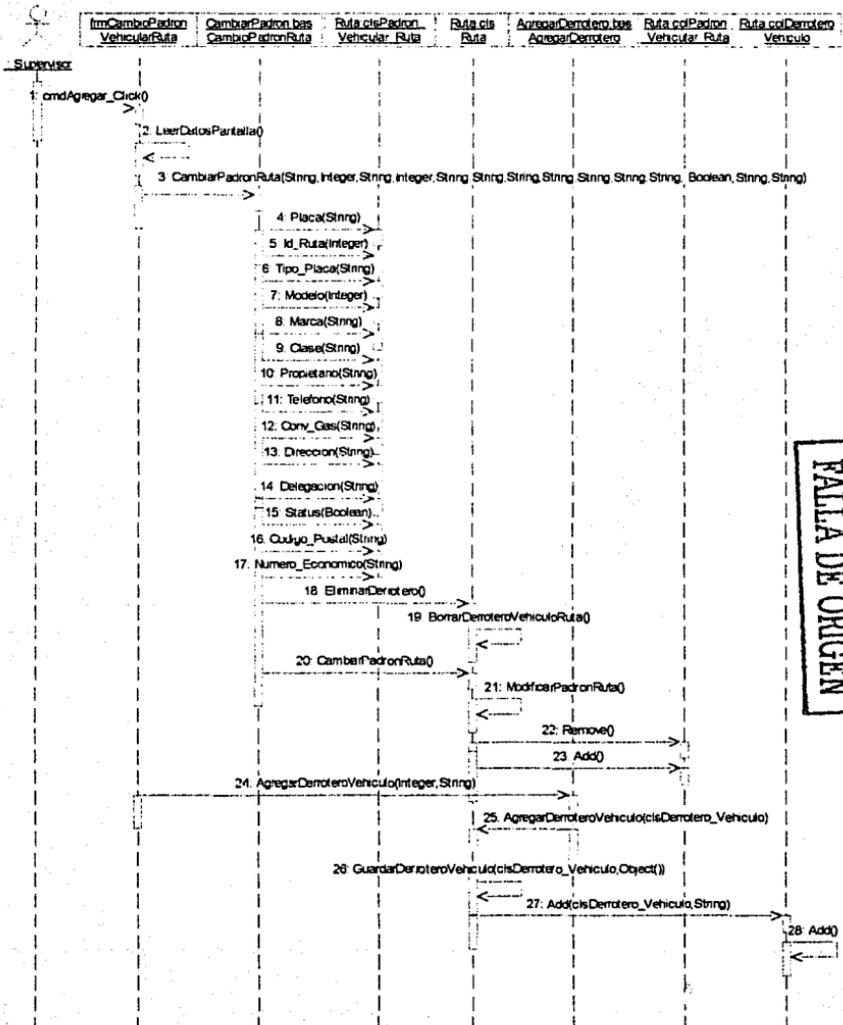


11. SOLICITAR CAMBIAR PADRÓN VEHICULAR



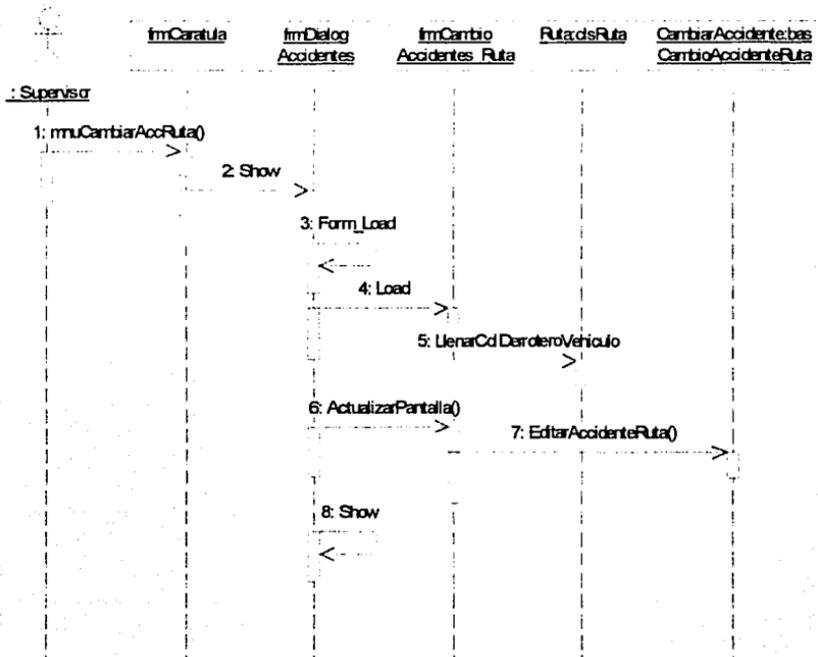
TESIS CON FALLA DE ORIGEN

12. CAMBIAR PADRÓN VEHICULAR



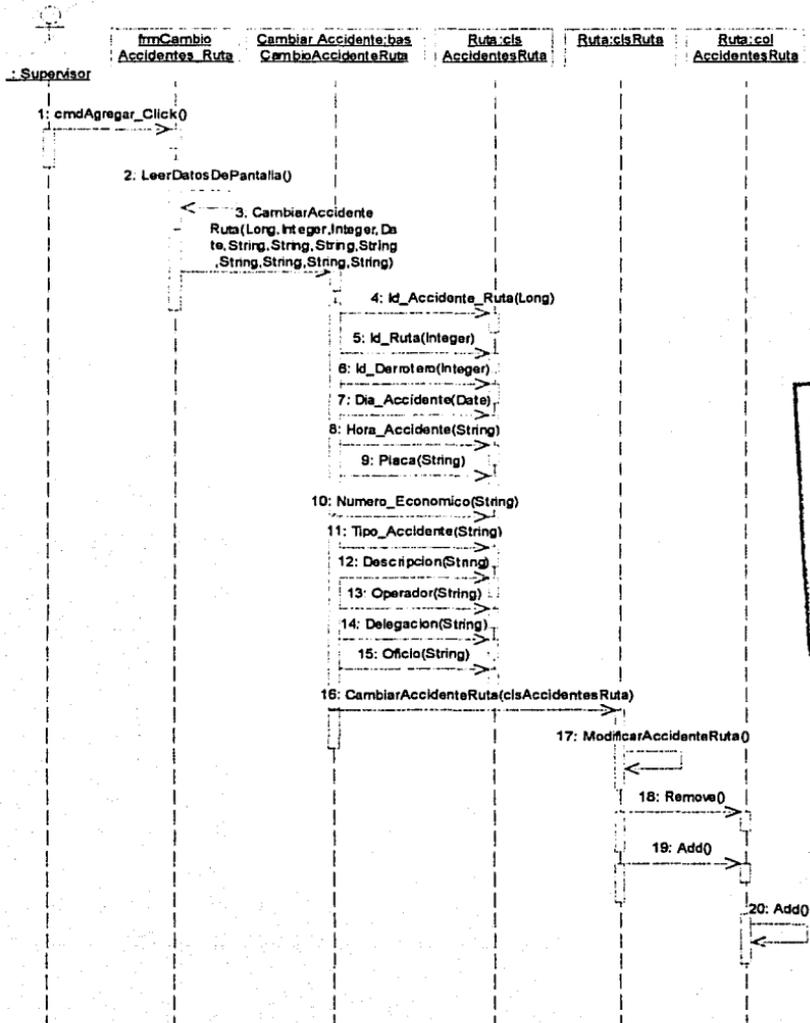
TESIS CON FALLA DE ORIGEN

## 13. SOLICITAR CAMBIAR ACCIDENTE



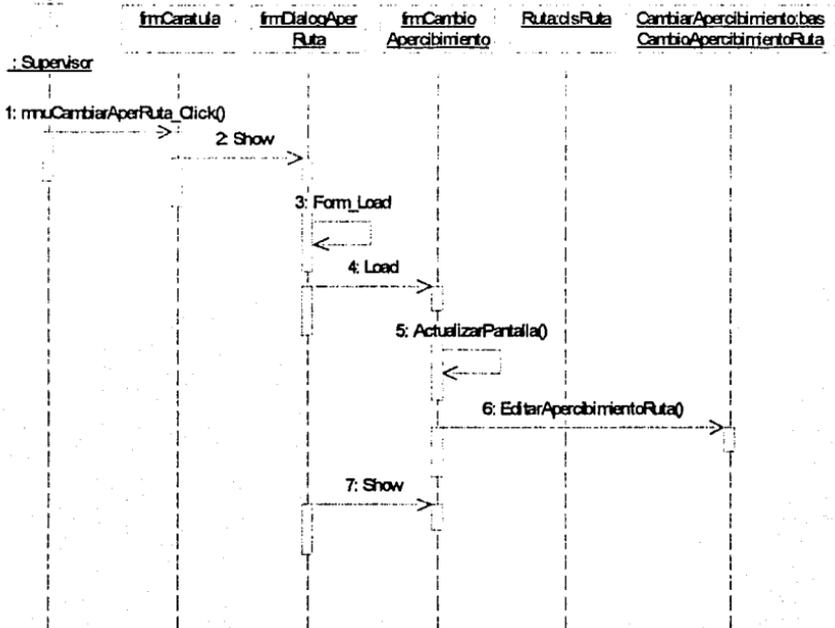
TESIS CON  
FALLA DE ORIGEN

14. CAMBIAR ACCIDENTE



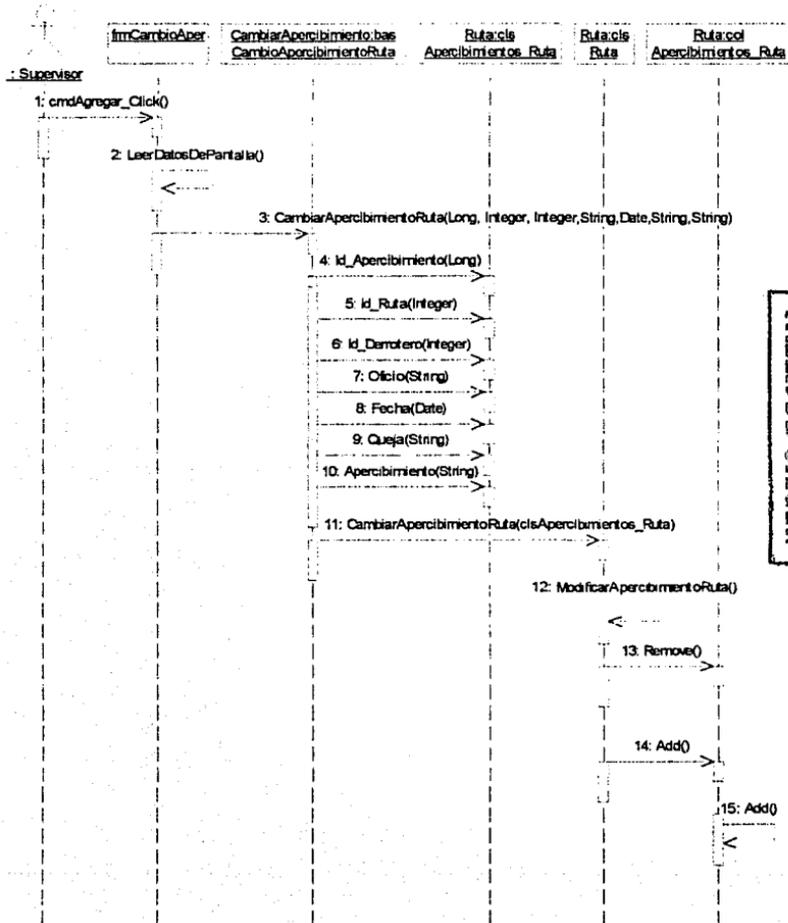
TESIS CON  
 FALLA DE ORIGEN

## 15. SOLICITAR CAMBIAR APERCIBIMIENTO



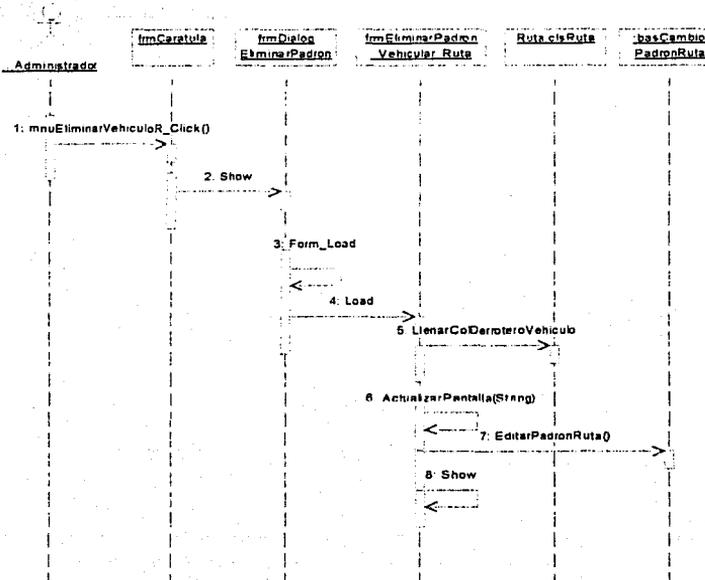
TESIS CON  
FALLA DE ORIGEN

## 16. CAMBIAR APERCIBIMIENTO



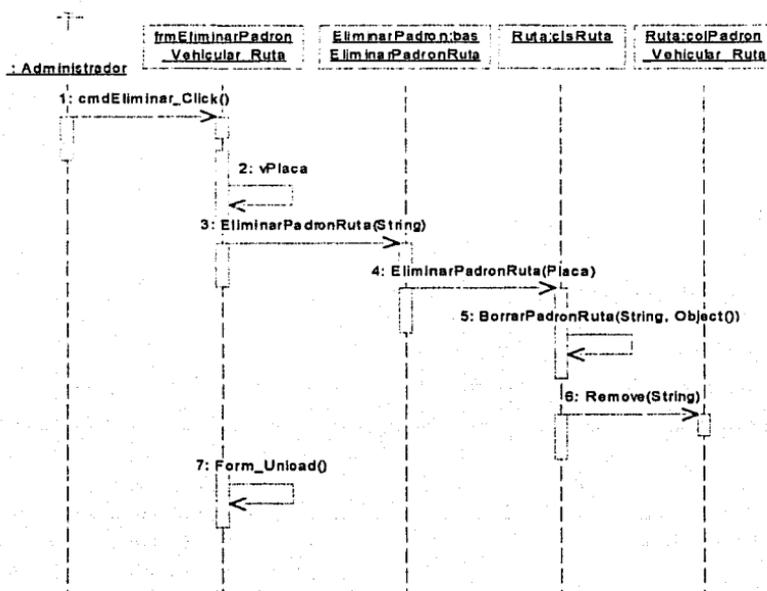
TESIS CON FALLA DE ORIGEN

17. SOLICITAR ELIMINAR PADRÓN VEHICULAR



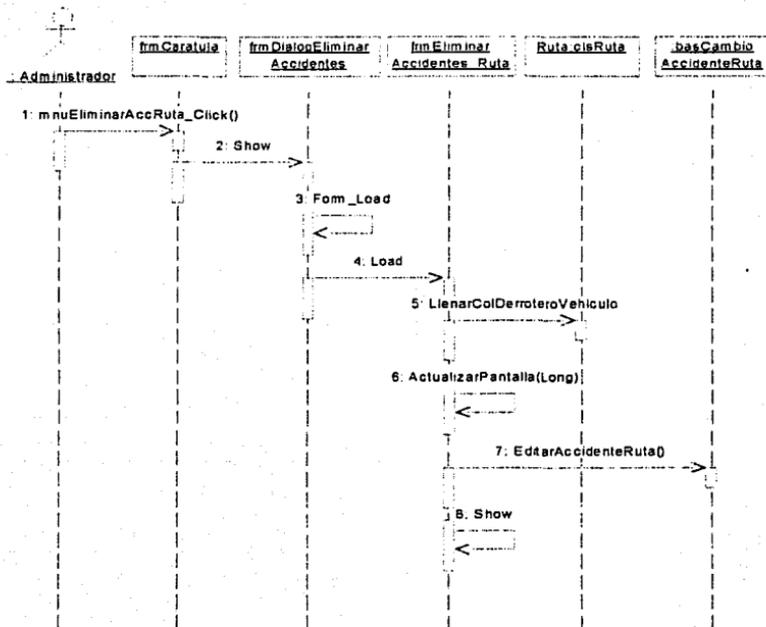
TESIS CON  
FALLA DE ORIGEN

## 18. ELIMINAR PADRÓN VEHICULAR



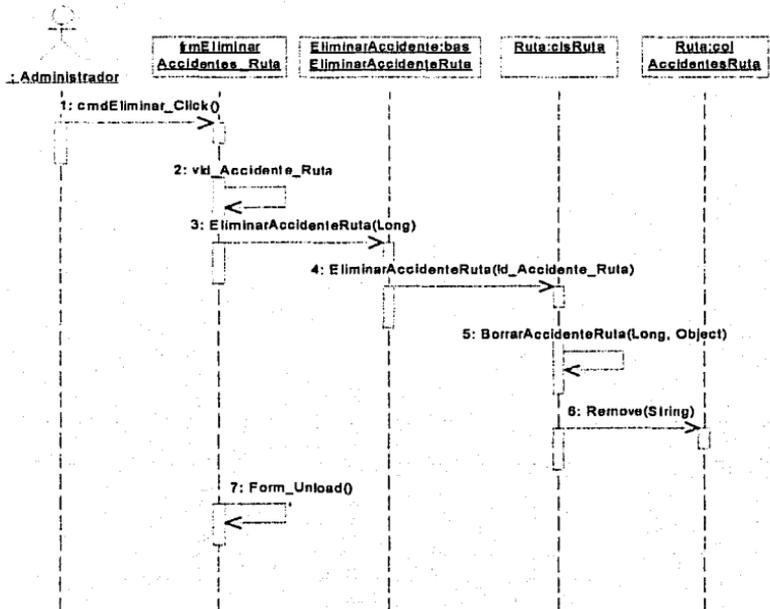
TEXTO CON  
FALLA DE ORDEN

## 19. SOLICITAR ELIMINAR ACCIDENTE



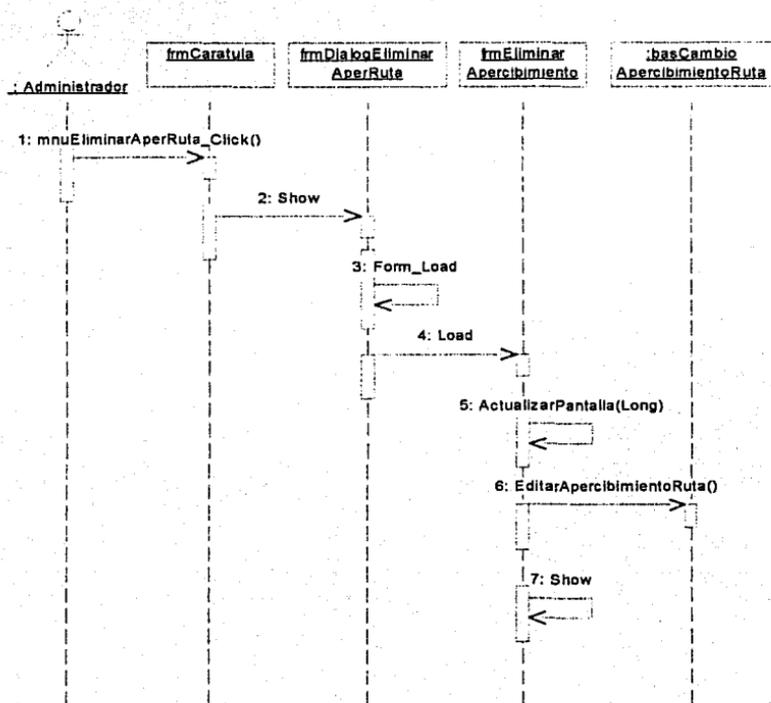
TESIS CON  
FALLA DE ORIGEN

20. ELIMINAR ACCIDENTE



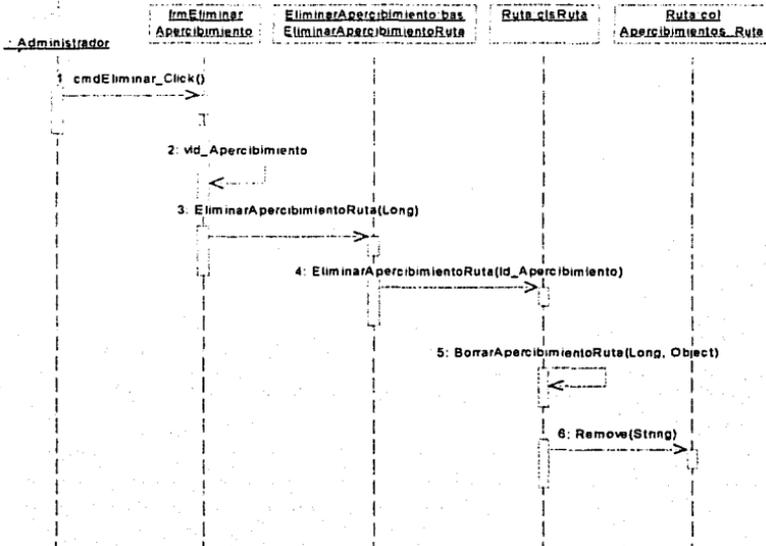
TESIS CON  
FALLA DE ORIGEN

## 21. SOLICITAR ELIMINAR APERCIBIMIENTO



TESIS CON  
FALLA DE ORIGEN

## 22. ELIMINAR APERCIBIMIENTO



TESIS CON  
FALLA DE ORIGEN

---

### III.4.3 CICLO DE VIDA DE DESARROLLO DEL SOFTWARE

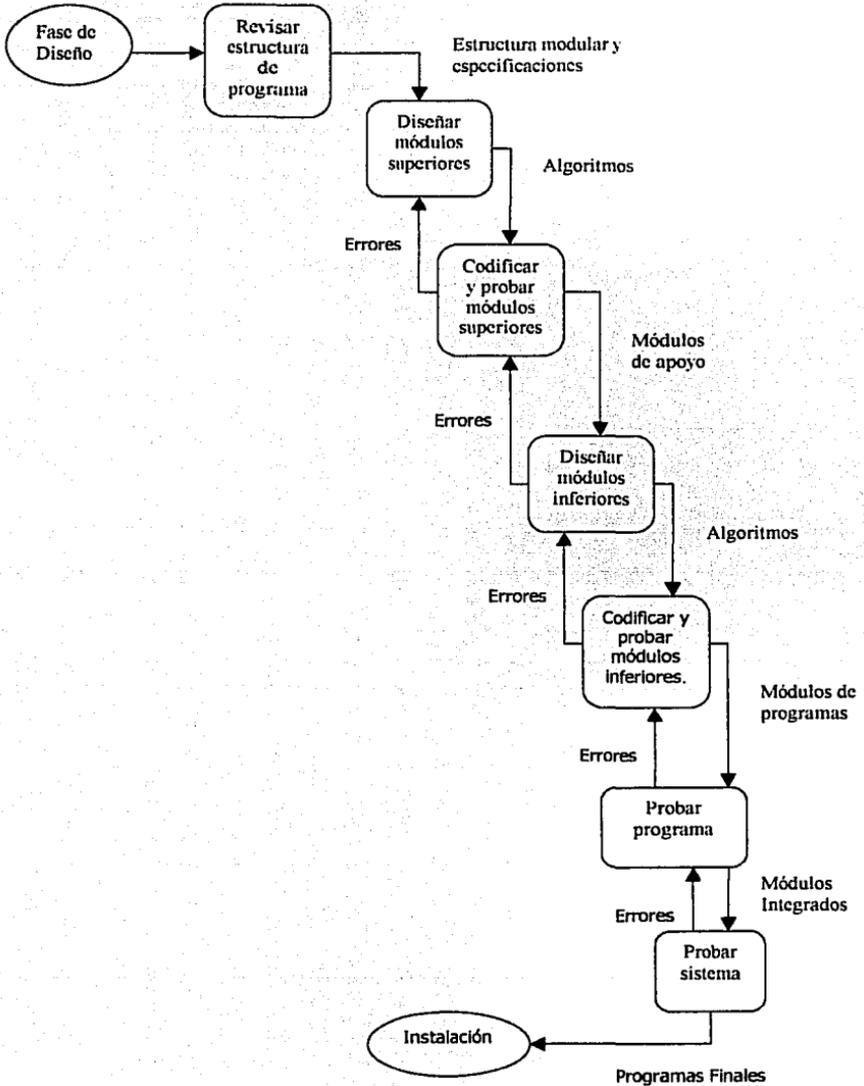
Antes de comenzar con la construcción del sistema, se *congeló* el documento con las especificaciones de diseño, es decir se restringen los cambios; si se trata de un cambio crítico que puede hacer que el sistema funcione o no, debe realizarse de inmediato y si sólo se trata de una mejora puede realizarse más adelante.

Ahora se menciona el ciclo de vida del desarrollo de software en que se basa la construcción del software del sistema; éste comienza con una revisión de la estructura de los programas, entendiendo por estructura a la *descomposición modular descendente* de los programas; es decir los módulos se diseñan, se codifican y se prueban empezando por el superior. Es importante mencionar que los módulos de nivel superior conducen a los de nivel más bajo, como se puede apreciar en la figura III.4.3.A

Cada módulo atraviesa por tres etapas de desarrollo: diseño, codificación y pruebas, una vez que hayan sido codificados y probados individualmente todos estos programas se integra y prueban en conjunto, ya conformados como el sistema final.

NOTA: Si los módulos se han codificado en estructura descendente deben probarse y depurarse en la misma forma descendente en que se han escrito.

FIG III.4.3.A CICLO DE VIDA DEL DESARROLLO DE SOFTWARE. ESTRATEGIA DESCENDENTE



### III.4.4 ¿CÓMO SE LLEVA A CABO LA IMPLEMENTACIÓN DE LA ARQUITECTURA DE 3 CAPAS?

- ❖ **CAPA DE PRESENTACIÓN** se tienen las pantallas en Visual Basic, a las cuales tiene acceso el usuario.

En la elaboración de la interfaz gráfica se siguen las siguientes reglas, para mejorar la comunicación con el usuario:

- Hacer que el usuario sienta que tiene el control de la interacción. Incluir siempre la opción de que el usuario cancele una acción luego que la haya iniciado.
- Enviar mensajes al usuario que le indiquen que los datos han sido introducidos de manera correcta, de lo contrario indicarle donde está el error.
- Dar al usuario diferentes formas de realizar una misma acción, como el cerrar una ventana.
- Colocar la información más importante en la esquina superior izquierda de la pantalla.
- Mantener los componentes que están relacionados juntos o con un marco que los delimite.
- Tener legibilidad.
- Restringir los colores de la interfaz, ya que demasiados colores distraen al usuario.
- Evitar el uso de fuentes cursivas y decorativas.
- Intentar mantener los botones, cuadros de lista, cuadros de texto, etc; del mismo tamaño.
- Alinear componentes y campos de datos a la izquierda, ya que esto reduce los movimientos oculares cuando el usuario tiene que revisar la pantalla.
- Cuando el usuario tiene que leer o procesar información y luego hacer clic en un botón, coloque los botones en una columna a la derecha de la información o en fila abajo. Esto es importante debido

---

a la tendencia natural de leer de izquierda a derecha. Si un botón es predeterminado debe colocarse primero y remarcarlo.

El objetivo de estas reglas es crear interfases que comuniquen la información adecuada en un contexto visual elemental, directo e intuitivo.

A continuación se lleva a cabo la **normalización de pantallas**, tomando como base las reglas antes mencionadas.

Para todos **los formularios en general**, se toma en cuenta que:

- ☐ El nombre del formulario debe estar en el extremo superior izquierdo de éste.
- ☐ El símbolo de la dependencia se debe localizar en el parte superior de los formularios.
- ☐ La información obligatoria debe estar escrita con letra tipo negrita.
- ☐ Cuando se pueda elegir entre una serie de opciones predeterminadas, se deben usar combos que desplieguen dicha información, por ejemplo para las delegaciones, tipos de placa, etc (véase figura III.4.4.A).
- ☐ Los botones deben estar colocados en la parte inferior o en el extremo derecho del formulario, según el tamaño de éste.
- ☐ Acceso a los botones por medio de la combinación: *Ctrl + la letra subrayada*, la cual es la primer letra de la palabra, en caso que se duplique se toma la segunda letra.
- ☐ El botón predeterminado siempre debe estar remarcado.
- ☐ Las fechas también se manejan por medio de un DTPicker (control de Visual Basic que despliega información de tipo fecha / hora, en este caso se usa la opción que despliega un calendario, véase figura III.4.4.B).

Para **los formularios de captura**:

- ☐ El color de fondo es amarillo claro.

- 
- ☐ Deben contar con un botón para *Agregar* y otro para *Cerrar*, siendo predeterminado el botón *Agregar*.

Para los **formularios de cambios:**

- ☐ El primer formulario que aparece en pantalla es color azul marino, aquí se selecciona la información específica de lo que se desea modificar.
  - ⊙ Debe contar con un botón para *Aceptar* y otro para *Cancelar*, siendo predeterminado el botón *Aceptar*.
- ☐ El segundo formulario, donde se llevan a cabo las modificaciones es color amarillo claro.
  - ⊙ Debe tener un botón para *Guardar* y otro para *Cerrar*, siendo predeterminado el botón *Guardar*.

Para los **formularios de bajas:**

- ☐ El primer formulario que aparece en pantalla es color gris claro, aquí se selecciona la información específica de lo que se desea eliminar.
  - ⊙ Debe contar con un botón para *Aceptar* y otro para *Cancelar*, siendo predeterminado el botón *Aceptar*.
- ☐ El segundo formulario, donde se muestra la información que se desea eliminar, es color amarillo claro.
  - ⊙ Debe tener un botón para *Eliminar* y otro para *Cancelar*, siendo predeterminado el botón *Eliminar*.

Para los **reportes:**

- ⊙ El formulario debe ser amarillo claro.

Nombre del Formulario



Símbolo de SETRAVI

Información Obligatoria:  Fechas: 18/12/01

Información General:  Opciones:

FIG III.4.4.A. EJEMPLO DE PANTALLA CON DESPLIEGUE DE OPCIONES.

Nombre del Formulario



Símbolo de SETRAVI

Información Obligatoria:  Fechas: 18/12/01

Información General:  Opciones:

2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	29	29
30	31					

ayer: 13/10/02

FIG III.4.4.B. EJEMPLO DE PANTALLA CON DESPLIEGUE DE FECHA.

- ❖ **CAPA DE NEGOCIOS** indica qué, cómo y por qué el sistema hace lo que hace, es aquí donde se especifica qué es lo que se desea que haga el sistema; en esta capa se tienen clases con sus propiedades particulares con atributos de lectura-escritura, módulos bas que contienen los métodos que se emplean como: guardar, agregar, etc.
- ❖ **CAPA DE DATOS**, es responsable del manejo de éstos; como se especificó anteriormente se usa como manejador de la bases de datos Microsoft SQL Server 2000.

El Universal Data Access (UDA, Acceso Universal de Datos) especifica una interfaz de programación de alto nivel llamada ActiveX Data Object (ADO) que usa OLE DB para acceder a los datos, el cual representa un modelo de programación más viable para los desarrolladores.

En esta arquitectura, la capa de presentación tiene muy poca responsabilidad y se dice que es *delgada*, ya que las ventanas no tienen código que se encargue de la lógica o procesamiento de la aplicación, por lo tanto las funciones son enviadas a las capas lógicas de la aplicación.

#### **III.4.4.1 ¿Cómo se lleva a cabo la comunicación entre las capas (com) de nuestro sistema?**

Para realizar la comunicación entre capa de presentación y la capa de negocios se usan los objetos *ocx*, es decir los controles que se ven en la pantalla (textbox, label, button, form, etc), los cuales pertenecen a la capa de presentación pero la capa de negocios los utiliza.

Para la comunicación de la capa de datos y la de negocios se utiliza los objetos de ADO (Conexion, Recorset).

#### **III.4.5 PRUEBAS DEL SISTEMA**

Las pruebas de un sistema, son las que se llevan a cabo para demostrar que no existen errores de programación, tanto lógicos como de estructura, comprobar que cumple con los requerimientos especificados durante la etapa de diseño y que funciona en la forma en que los usuarios esperan que lo haga.

Se ejecuta el sistema con la intención explícita de encontrar errores, es decir hacer que el programa falle; realizando las siguientes pruebas:

- **Pruebas Individuales**, son pruebas realizadas sobre módulos individuales para detectar errores de código y lógica, contenidos dentro de ese módulo.

Al encontrar algún error de programación, se realiza la ejecución de cada instrucción del módulo en que se detectó el error, para encontrar el origen de éste. También se verifica el rango de valores que debe aceptar el módulo.

No es recomendable realizar todas las pruebas al final del sistema, es por ello que se realizan las pruebas individuales sobre cada uno de los módulos.

- **Pruebas de sistemas** es una prueba que garantiza que los programas de aplicación escritos individualmente funcionan de modo adecuado cuando se integran en el sistema global.

Es necesario realizar sobre el conjunto integrado de programas una prueba de sistemas, para asegurar que un programa acepta correctamente como entrada, la salida de otros programas.

---

Las pruebas hechas al sistema no prueban el software en sí, sino la integración de cada módulo en el sistema.

La preocupación principal es la compatibilidad de los módulos individuales, tratando de hallar áreas donde los módulos hayan sido diseñados con especificaciones distintas para la longitud y tipo de datos, así como los nombres de dichos datos, por ejemplo en un módulo se encontró que el tipo de dato para la identificación del derrotero es de tipo carácter, cuando realmente se almacena de tipo numérico.

- **Pruebas de volumen**, el volumen de la prueba extiende el sistema a sus límites de diseño con grandes volúmenes de transacciones, datos, etc. Puede ser útil llevar el sistema no sólo a sus límites de diseño, sino sobrepasarlos; de esta forma cualquier margen de seguridad puede detectarse.
- **Pruebas de tensión**, difieren de las pruebas de volumen en que el sistema se sujeta a cargas pesadas, aunque no necesariamente alcance alguno de los límites del sistema. En las aplicaciones de tiempo real es esencial sujetar al sistema a tanta tensión como sea posible; ya que algunas situaciones que parecen una posibilidad remota y no se prueban pueden ocurrir bajo ciertas condiciones, por ejemplo puede parecer imposible que todos los usuarios del sistema quieran registrarse al mismo tiempo, pero esto puede pasar cuando el sistema se recupera de una caída.
- **Prueba de aceptación de sistema** es una prueba final del sistema realizada por los usuarios mediante el uso de datos reales durante un lapso de tiempo, permite comparar el producto de cómputo final con el contrato original. Se realizaron 2 pruebas:
  - **Prueba de verificación (Prueba Alfa)** que comprueba el sistema en un entorno simulado haciendo uso de datos simulados, su objetivo es buscar posibles errores y omisiones con respecto a las especificaciones de diseño y de los usuarios finales, que no se lograron satisfacer durante la construcción.
  - **Prueba de validación (Prueba Beta)** que comprueba el sistema en forma real mediante el uso de datos reales, aquí se analizan varios aspectos:
    - **Rendimiento del sistema**, en el que se determinó aceptable la productividad y el tiempo de respuesta de los procesos durante una jornada normal de trabajo.

- **Prueba de ergonomía:** Fácil de aprender y manejar, por alguien que ha sido capacitado o al menos haya leído el manual de usuario.

Esta prueba es la última oportunidad que tienen los usuarios y directivos para aceptar o rechazar el sistema.

### III.4.6 EVALUACIÓN DE LA CALIDAD DEL SISTEMA.

La calidad del sistema depende de su diseño, desarrollo y pruebas; como reflejo de calidad tenemos su *confiabilidad*, es decir un sistema es confiable si al usarse no produce fallas importantes o costosas. Las fallas son el resultado de errores de diseño introducidos cuando se formularon las especificaciones y se codifica el sistema.

Cuando los usuarios no están convencidos que el sistema hace lo correcto, se puede decir que *no es un sistema confiable*.

El sistema realizado pasó satisfactoriamente las pruebas efectuadas, cumpliendo con las especificaciones indicadas por el usuario, fue muy importante la realización de pruebas independientes de cada módulo, apoyando a las pruebas de integración del sistema, las cuales permiten identificar los errores de codificación. Cabe señalar, que un programa de prueba efectivo no garantiza la confiabilidad del sistema, éste es asunto del diseño del sistema.

El número de errores que se presentan, se pueden minimizar si se revisa cada etapa en el desarrollo del software, sin embargo, aunque estas revisiones reducen drásticamente el número de éstos, son insuficientes para identificar todos los problemas, como por ejemplo errores de análisis y definición de requerimientos, éstos últimos son los errores más complicados ya que significa que el desarrollo del sistema no cumple con los objetivos, en este caso no hubo este tipo de problemas ya que el sistema cumple con todas las especificaciones de requerimientos planteados.

Una aspecto importante en la actualidad que demanda un diseño sólido son las adquisiciones corporativas. Cuando una empresa adquiere a otra y su sistema fue bien diseñado se facilita enormemente la conversión, además de que la implementación procede con menos problemas.

# ***CAPITULO IV***

## ***IMPLEMENTACIÓN Y SOPORTE***

Después de que se ha concluido el sistema, hay que instalarlo al cliente para que determine y visualice que todo aquello que fue requerido cumple satisfactoriamente sus necesidades.

Una vez puesto en funcionamiento el sistema, la función del analista es darle soporte, de hecho una buena parte del tiempo y el esfuerzo de la mayoría de los analistas se invierte en proporcionar un soporte continuo a los sistemas existentes.

#### **IV.1 IMPLEMENTACIÓN DEL SISTEMA**

En la Implementación se da la transición entre la construcción del nuevo sistema y el paso de éste a producción.

En esta etapa se integran los componentes técnicos, se lleva a cabo la instalación, plan de conversión, evaluación y documentación del sistema, así como la capacitación del personal.

Las principales actividades que se llevan a cabo son:

- Instalación de software comercial
- Instalación del sistema
- Proporcionar capacitación y documentación a usuarios
- Realizar Plan de Conversión
- Evaluación del sistema

##### **IV.1.1 Instalar y probar los nuevos paquetes de software comerciales.**

El servidor ya cuenta con la instalación del sistema operativo Windows Advanced Server, únicamente se instala el software de Microsoft SQL Server y Visual Basic 6.0 y se realiza una prueba, para garantizar su correcto funcionamiento.

Los equipos clientes tienen instalado Windows 98 como sistema operativo, por lo tanto no es necesario instalar alguna paquetería adicional, ya que solo se les copia un archivo ejecutable el cual contiene todos los archivos necesarios para la ejecución del sistema. Es requisito indispensable ser parte de la red local de la dependencia.

---

### **IV.1.2 Instalar el sistema.**

La base de datos se instaló directamente en el servidor por medio de un disco compacto.

Para la aplicación se creó un archivo ejecutable que carga automáticamente todos los archivos que requiere el sistema, posteriormente se hicieron copias de éste en discos compactos, facilitando de esta manera su instalación (las instrucciones para llevar a cabo dicha instalación, se encuentran en el manual técnico).

La instalación se llevo a cabo de manera exitosa, sin ningún tipo de conflicto por resolver, ya que se trata de un sistema que no requiere de una configuración complicada.

### **IV.1.3 Capacitación y documentación a usuarios**

La calidad de la capacitación recibida por el personal ayuda, obstruye y puede llegar a impedir la implementación exitosa de un sistema de información, es por ello que aquellos que estén asociados con el sistema deben conocer con detalle su funcionamiento, se debe tomar en cuenta que el descuido en cualquier aspecto de la capacitación puede llevar a situaciones que produzcan frustración y errores del usuario.

La capacitación realizada fue grupal, ya que se aprovecha mejor el tiempo y es posible abordar una mayor cantidad de dudas, se llevó a cabo dentro de las instalaciones y con los equipos de la dependencia.

La capacitación de los usuarios incluyó una breve introducción en el manejo de los equipos, ya que algunos desconocían el manejo de éstos. Posteriormente se explican las operaciones que realiza el sistema. Estas son: la captura de datos (como guardar nuevos registros), la edición de datos (como realizar cambios en los datos grabados previamente), la realización de consultas (como localizar registros específicos) y el borrado de registros. Estas actividades ocuparon la mayor parte del tiempo de la capacitación.

Por último se explica cómo identificar problemas, determinando si el problema que surge es causado por el equipo, por el software o por los usuarios del sistema.

En lo referente a documentación, un buen manual, entendible y sin errores, facilita no sólo el empleo del programa, sino también la aceptación por parte de usuarios escépticos al empleo de estos medios para el manejo de información; es por ello que se proporciona manuales de usuario.

#### IV.1.4 Realizar un plan de conversión

El plan de conversión indica la estrategia de instalación que se seguirá para convertir un sistema existente (manual) en un nuevo sistema de información automatizado. Las estrategias más comunes son:

- ⊙ ***Puesta en servicio total (Big Bang).*** En una fecha específica, se da fin al sistema antiguo y se pone en funcionamiento el nuevo sistema; éste es un planteamiento de alto riesgo ya que aún pueden surgir problemas que el nuevo sistema no pueda solucionar al menos en cierto tiempo.
- ⊙ ***Conversión en paralelo.*** Se hace coexistir ambos sistemas (el anterior y el nuevo) durante cierto tiempo para garantizar la resolución de todos los posibles problemas que el sistema nuevo pueda producir antes de que el antiguo sea retirado.

Esta estrategia reduce el riesgo de errores en el nuevo sistema que puedan provocar un daño irreparable a la empresa, la desventaja es que se debe solventar el costo de trabajar con dos sistemas al mismo tiempo.

La puesta en servicio puede ser total o gradual(sólo funcionan algunos segmentos del sistema).

- ⊙ ***Conversión por puestos.*** Cuando el nuevo sistema se va a utilizar en distintos puestos de la empresa, se suele probar primero en un determinado puesto (llamado puesto de prueba beta), cuando éste lo haya aprobado entonces podrá ser instalado en los demás, ya que de esta manera se evita la instalación en otros puestos si se detectan errores. Puesta en servicio total o conversión en paralelo.
- ⊙ ***Conversión por etapas.*** Se basa en la instalación de versiones del sistema y se puede poner en servicio total, en paralelo o por puestos.

En el caso de DGT se lleva a cabo una conversión en paralelo, ya que será un cambio progresivo, el sistema antiguo es básicamente manual por lo tanto no presenta ningún problema en el uso de ambos sistemas al mismo tiempo; el motivo principal de esta elección, es que se corren menos riesgos al instalar el sistema total, ya que existió una etapa previa donde se pudo detectar la presencia de errores graves, que pueden afectar a la dependencia.

---

### **IV.1.5 Evaluación del sistema**

Nos ayuda a determinar si el sistema cumple con los requerimientos planteados; cuando se realiza en forma adecuada proporciona información valiosa que puede ayudar a mejorar la efectividad en el desarrollo de aplicaciones posteriores.

Para evaluar el sistema que se ha desarrollado se ha tomado en cuenta lo siguiente:

- ◇ Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.
- ◇ Identificación de beneficios para la organización en áreas como costos, eficiencia operacional e impacto competitivo.
- ◇ Opinión de directivos y administradores dentro de la organización así como de los usuarios finales acerca del nuevo sistema.

Por último se realiza una evaluación del funcionamiento del sistema, en donde encontramos que cumple con los objetivos establecidos en la fase de planeación, fue realizado en el tiempo esperado y con el presupuesto determinado en un principio.

Los beneficios obtenidos más palpables hasta el momento, es la reducción del tiempo de elaboración de oficios y reportes, menos uso de papelería y consulta de información actualizada en cualquier momento.

La mayoría de los usuarios comentan que el sistema les facilita muchas de las actividades que antes realizaban manualmente, sólo que necesitan tiempo para familiarizarse más con él.

Posteriormente se recopiló información para el mantenimiento, el sistema permanecerá mientras no se requieran cambios debido a desarrollos internos, como nuevos directivos. La revisión después de la implementación es la primer fuente de información de los requisitos de mantenimiento.

## **IV.2 SOPORTE DE SISTEMAS**

El soporte de sistemas es el mantenimiento permanente después de que haya sido puesto en funcionamiento el sistema, incluyendo tanto el mantenimiento de los programas como las mejoras que puedan realizarse.

Es la estrategia para mantener en óptimas condiciones el funcionamiento del sistema.

Durante esta etapa se realizan las siguientes actividades:

★ **Mantenimiento Correctivo (Corrección de errores)**

Su objetivo es modificar los programas existentes para corregir errores (un error se define como la desviación de la especificación) que se cometieron en el diseño o desarrollo del sistema, por medio de acciones correctivas. No se toman en cuenta las nuevas necesidades ni las mejoras.

Las actividades de análisis durante el mantenimiento implican la comprensión del alcance y efecto de una modificación, además de las restricciones para realizar dicha modificación.

Las actividades de diseño implican rediseñar el software para incorporar los cambios deseados. La documentación interna se debe actualizar, se deben realizar las pruebas correspondientes para evaluar la integración de la reciente modificación y posteriormente llevar a cabo su implementación. Por último, se deben actualizar los manuales.

★ **Mantenimiento Adaptativo**

Implica la modificación o adaptación del sistema a nuevas necesidades de la empresa.; en su mayoría son propuestas por los usuarios. Algunas mejoras pueden requerir la ampliación del almacenamiento de datos.

★ **Mantenimiento Perfectivo (Mantenimiento Productivo)**

No altera las especificaciones, pero mejora el desempeño al hacer que el software consuma menos recursos, entiéndase como recursos al tiempo de ejecución y uso de memoria.

★ **Mantenimiento Preventivo.**

Implica hacer cambios al software que por si solos no mejoran ni la corrección ni el desempeño, pero provocan que las actividades futuras de mantenimiento sean más fáciles y baratas de realizar.

★ **Reingeniería**

La reingeniería es adaptar el sistema ante un cambio tecnológico, así como arreglar el sistema antes de que falle o hacer el sistema más sencillo de manejar para cuando falle o tenga que ser adaptado. Los proyectos de reingeniería se deben a cambios en la tecnología o a la necesidad de aprovechar la tecnología existente.

★ **Apoyo a los usuarios.**

Independientemente de la capacitación y la documentación proporcionada, los usuarios siempre requerirán asistencia adicional que puede consistir en: observación del uso del sistema, elaboración de encuestas para saber si el usuario está conforme con el funcionamiento del sistema, ofrecer capacitación adicional, registrar ideas y solicitudes sobre futuras mejoras.

El objetivo principal del desarrollo de software debe ser la producción de sistemas que faciliten su propio mantenimiento, para esto es importante crear programas cuyas características sean la claridad, modularidad y la buena documentación interna del código fuente.

En cuanto al sistema sólo se ha realizado mantenimiento adaptativo al realizar algunos cambios al sistema, como la adición de información al formato de reportes. También se ha auxiliado a los usuarios en pequeños problemas que se les presentan.

Actualmente se está trabajando en el mantenimiento perfectivo y preventivo, para hacer más eficiente este sistema.

Por ejemplo, se está reduciendo el número de módulos .bas unificándolos por tabla y no por operación, por ejemplo existe un basAgregarApercibimiento, basCambiarApercibimiento, basEliminarApercibimiento, y lo que se propone es integrarlos en un sólo módulo basApercibimiento, para mejorar la eficiencia del sistema.

Muchas organizaciones invierten cuantiosos recursos económicos para dar un buen soporte a sus sistemas. Estos costos pueden llegar a elevarse a niveles alarmantes, por lo que se sugiere llevar un control estricto sobre éste.

---

# CONCLUSIONES

A pesar del poco tiempo que lleva funcionando el sistema, se puede decir que se cumplió satisfactoriamente el objetivo de este trabajo, al crear un sistema automatizado de información que satisface las necesidades expuestas en un inicio por la dependencia, permitiendo obtener una mejoría notable en cuanto a:

- ➊ Acceso a la información en menos tiempo y de manera confiable.
- ➋ Información actualizada para tener un mayor control sobre las rutas.
- ➌ Actividades administrativas que se realizan en menos tiempo, en comparación con el sistema tradicional.
- ➍ Apoyándose en las consultas y reportes, se detectan irregularidades que afectan directamente al usuario, y en base a esta información se toman acciones correctivas.
- ➎ El contar con una interfaz gráfica se permitió el fácil manejo del sistema, disminuyó considerablemente el tiempo de capacitación del usuario.

Ahora, desde el punto de vista de la tecnología de información usada, se concluye que:

- ➏ Al seguir la metodología de análisis y diseño orientado a objetos se facilita el mantenimiento del sistema, ya que la complejidad no se maneja de manera directa sino que se encapsula, lo cual representa una gran ventaja para el desarrollo del mismo.
- ➐ Gracias a la arquitectura de 3 capas, será posible migrar las pantallas hechas en Visual Basic a un sitio web, en un futuro próximo.

Este trabajo, es el primer paso de un largo camino que falta por recorrer, ya que se podrá expandir este sistema de acuerdo a las necesidades de la dependencia.

---

# BIBLIOGRAFÍA

📖 Castaño, Adoración de Miguel; Piattini Velthuis, Mario Gerardo.  
FUNDAMENTOS Y MODELOS DE BASES DE DATOS.  
México, Alfaomega 2a ed, 1999

📖 Cohen Karen, Daniel.  
SISTEMAS DE INFORMACIÓN PARA LA TOMA DE DECISIONES.  
México, McGraw-Hill, 2a ed, 1996

📖 Date, C. J.  
INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS.  
México, Pearson Educación, 7a ed. 2001

📖 Elmasri, Ramez; Navathe, Shamkant B.  
FUNDAMENTALS OF DATABASE SYSTEMS  
Addison-Wesley, 3tr ed, 1997

📖 Halpin, Terry A.  
INFORMATION MODELING AND RELATIONAL DATABASES : FROM  
CONCEPTUAL ANALYSIS TO LOGICAL DESIGN.  
San Francisco, Morgan Kaufman Publishers, 2001

📖 Harmon, Paul; Sawyer, Brian.  
UML FOR VISUAL BASIC 6.0 DEVELOPERS : USING VISUAL MODELER AND  
RATIONAL ROSE 98  
Morgan Kaufman Publishers, 1999

📖 Larman, Craig.  
APPLYING UML AND PATTERNS : AN INTRODUCTION TO OBJECT-ORIENTED  
ANALYSIS AND DESIGN.  
Prentice Hall PTR, 1997

📖 Lucas, Henry C.  
SISTEMAS DE INFORMACIÓN : ANÁLISIS, DISEÑO, PUESTA A PUNTO.  
Madrid, Paraninfo, 1984

📖 Murdick, Robert G; Munson, John C.  
SISTEMAS DE INFORMACIÓN ADMINISTRATIVA.  
México Prentice Hall, 1988

TESIS CON  
FALLA DE ORIGEN

---

📖 Navathe, Batini Ceri.  
DISEÑO CONCEPTUAL DE BASES DE DATOS : UN ENFOQUE DE ENTIDADES-  
INTERRELACIONES  
Madrid, Addison-Wesley

📖 Norris, Mark; Rigby, Peter.  
INGENIERÍA DE SOFTWARE EXPLICADA.  
México, Megabyte, 1994

📖 Schmuller, Joseph.  
APRENDIENDO UML EN 24 HORAS.  
México, Pearson Educación, 2000

📖 Silberschatz, Abraham; Korth, Henry F.  
FUNDAMENTOS DE BASES DE DATOS.  
McGraw-Hill Interamericana, 3a ed. 1998

📖 Whitten, Jeffrey L; Bentley, Lonnie D.  
SYSTEMS ANALYSIS AND DESIGN METHODS.  
McGraw-Hill, 4th ed. 1998

📖 Yourdon, Edward.  
MODERN STRUCTURED ANALYSIS.  
Englewood cliffs, New Jersey 1989

🌐 <http://www.microsoft.com/catalog/display.asp?site=729&subid=44&pg=2>

🌐 <http://www.microsoft.com/catalog/display.asp?site=10145&subid=44&pg=2>

🌐 <http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/INDEX.HTM>

🌐 <http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib615/cap2062.HTM>

TESIS CON  
FALLA DE ORIGEN

---

# ***ANEXO A***

## ***MANUAL DE INSTALACIÓN***

# INTRODUCCIÓN

Este manual consiste en mencionar los aspectos técnicos necesarios, para llevar a cabo la instalación del sistema en la dependencia.

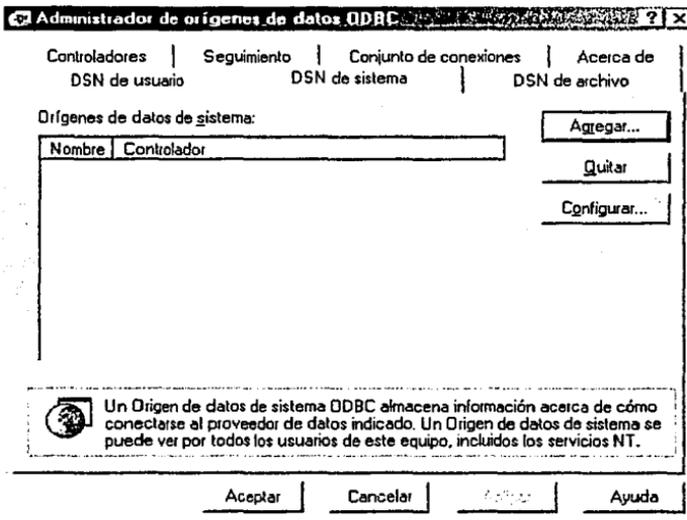
Con las siguientes especificaciones se podrán conectar los equipos cliente al servidor que contiene la base de datos.

Es importante mencionar que este manual no es apto para los usuarios finales, por lo tanto sólo el administrador o personal autorizado podrá llevar a cabo esta instalación.

Para comenzar con la instalación, se presiona **Inicio – Configuración –Panel de Control**, se busca el siguiente icono y se da doble clic.

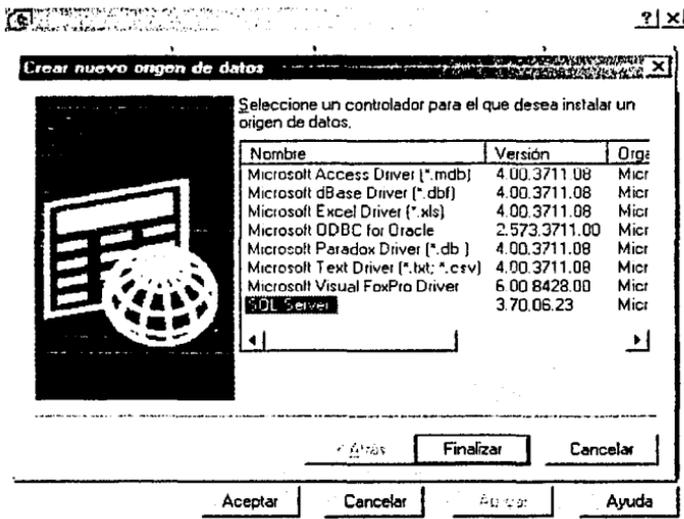


Aparece la siguiente pantalla y se elige la pestaña **DSN de Sistema** y se presiona **Agregar**.



TESIS CON FALLA DE ORIGEN

Ahora se define el origen de datos, en este caso es SQL Server y se presiona el botón **Finalizar**.



Se muestra la siguiente pantalla donde se escriben los datos con los que nos conectamos al origen de datos. La dirección IP es la correspondiente al servidor al que nos conectamos. Se presiona **Siguiente**.

TESIS CON  
FALLA DE ORIGEN

## Crear un nuevo origen de datos para SQL Server



Este asistente le ayudará a crear un origen de datos ODBC que podrá usar para conectarse a SQL Server.

¿Qué nombre desea utilizar para referirse al origen de datos?

Nombre:

¿Cómo desea describir el origen de datos?

Descripción:

¿Con qué servidor SQL Server desea conectarse?

Servidor:





Ahora se escribe el identificador y contraseña, para tener acceso a la base y se presiona **Configuración del Cliente**, en ésta misma pantalla.

## Crear un nuevo origen de datos para SQL Server



¿Cómo desea que SQL Server compruebe la autenticidad del Id. de inicio de sesión?

- Con la autenticación de Windows NT, usando el Id. de inicio de sesión de red.
- Con la autenticación de SQL Server, usando un Id. de inicio de sesión y una contraseña introducidos por el usuario.

Para cambiar la biblioteca de red usada para comunicarse con SQL Server, haga clic en Configuración del cliente.

Conectar con SQL Server para obtener la configuración predeterminada de las opciones de configuración adicionales

Id. de inicio de sesión:

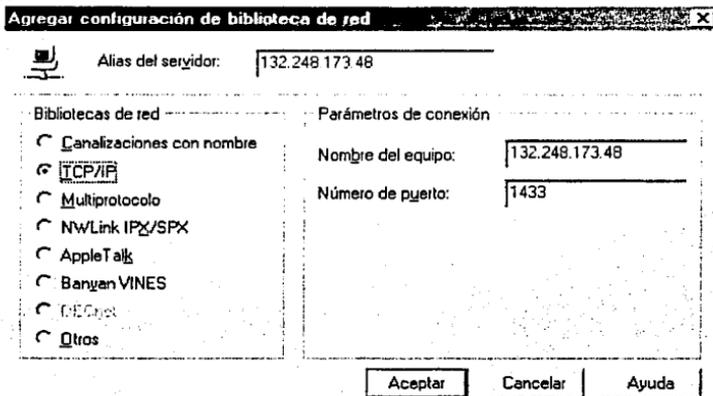
Contraseña:





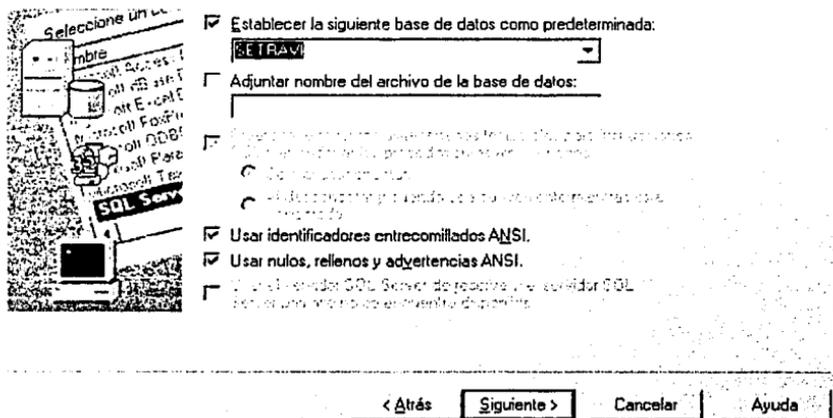
**TESIS CON  
FALLA DE ORIGEN**

Después de presionar el botón **Configuración del Cliente**, aparece la siguiente pantalla. Se selecciona el protocolo TCP/IP y se deja intacta la dirección del servidor, se presiona **Aceptar**.



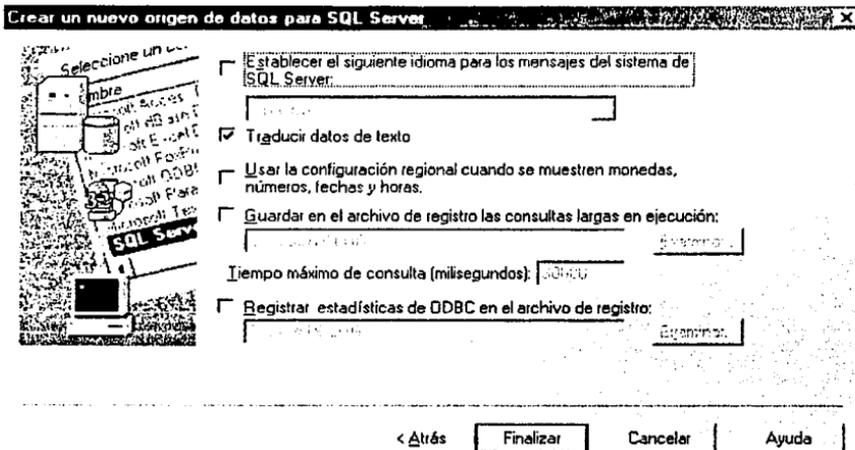
A continuación se despliega el combo que se tiene en pantalla, se selecciona el nombre de la base. En un futuro este combo podrá tener una lista más amplia, si se construyen más bases de datos. Se presiona **Siguiente**.

Crear un nuevo origen de datos para SQL Server

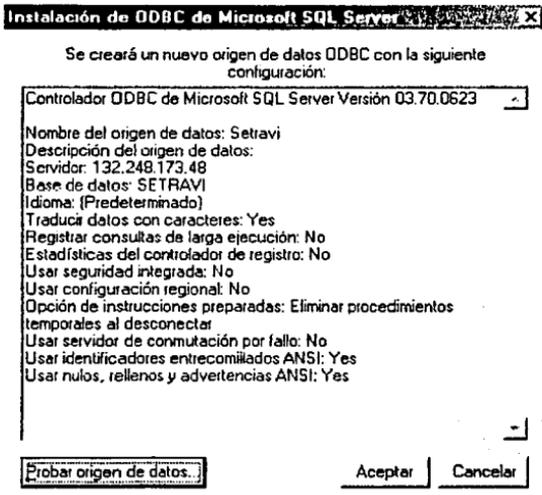


TESIS CON  
 FALLA DE ORIGEN

Se deja únicamente seleccionado Traducir datos de texto y se presiona el botón **Finalizar**.

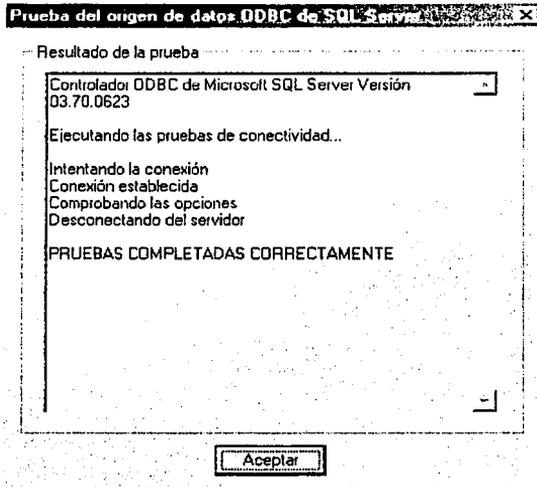


Posteriormente se presenta el siguiente cuadro con las especificaciones que se le han dado. Se presiona **Probar Origen de datos**.



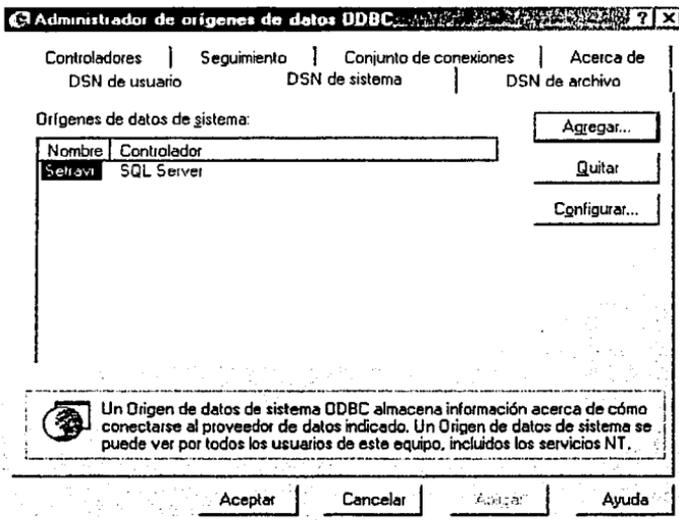
NOTA: El servidor debe estar prendido.

Si las pruebas fueron satisfactorias, se despliega la siguiente pantalla.  
Se presiona **Aceptar**.



Nos regresa a la primera pantalla, pero ahora se tiene un origen de datos establecido, con el cual se tiene una conexión con la base de datos, como lo muestra la siguiente imagen:

TESIS CON  
FALLA DE ORIGEN



Se presiona **Aceptar** y se cierra esta ventana.

Por último sólo es necesario introducir el CD con el sistema e instalarlo en cada uno de los clientes.

De esta manera se crea un icono de acceso directo al sistema **DGT**, el cual al darle doble clic aparece la pantalla que solicita su contraseña, pero eso se explica en el manual de usuario.

TESIS CON  
FALLA DE ORIGEN

---

# ***ANEXO B***

## ***MANUAL DE USUARIO***

---

---

# INDICE

TEMA	PÁGINA
Introducción .....	B-1
B.1 Acceso al sistema.....	B-2
B.2 Agregar Información.....	B-3
B.2.1 Ruta.....	B-3
B.2.2 Derrotero.....	B-5
B.2.3 Padrón Vehicular.....	B-7
B.2.4 Riesgo.....	B-8
B.2.5 Apercibimiento.....	B-9
B.2.6 Queja.....	B-10
B.2.7 Estado de Cuenta.....	B-11
B.3 Modificar Información.....	B-13
B.3.1 Ruta.....	B-13
B.3.2 Derrotero.....	B-14
B.3.3 Padrón Vehicular.....	B-16
B.3.4 Riesgo.....	B-17
B.3.5 Apercibimiento.....	B-18
B.3.6 Queja.....	B-19
B.3.7 Estado de Cuenta.....	B-20
B.4 Eliminar Información.....	B-21
B.5 Reportes.....	B-22
B.5.1 Reporte por Ruta.....	B-23
B.5.2 Reporte de Derroteros por Ruta.....	B-24
B.5.3 Reporte del Padrón Vehicular por Ruta.....	B-25
B.5.4 Reporte de Riesgos por Ruta.....	B-26

---

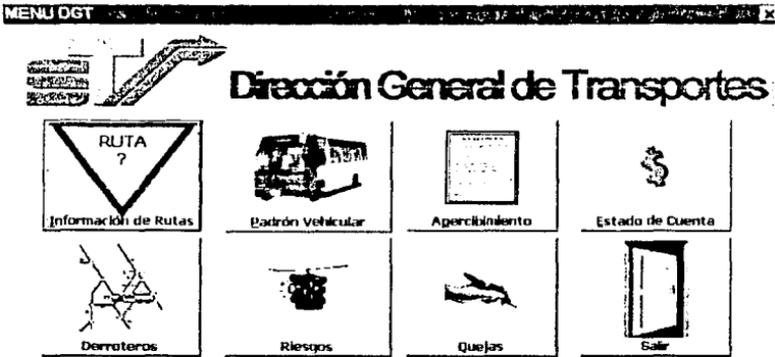
# INTRODUCCIÓN

La finalidad del presente manual, es apoyar a los usuarios en su capacitación sobre el nuevo sistema, explicando de manera sencilla, el manejo de éste; así como los pasos a seguir cuando se desea realizar alguna operación, como agregar, modificar, etc.

El sistema cuenta con reportes preestablecidos que facilitan la toma de decisiones, es por ello, que se sugiere seguir las indicaciones contenidas en este manual, para asegurar que se cumplan las necesidades del usuario.



Posteriormente se presenta la pantalla de menú, al dar un clic sobre alguno de los botones, aparece un submenú en el cual se selecciona la operación que se desea realizar.



## B.2 AGREGAR INFORMACIÓN

### B.2.1 Ruta

Para agregar una Ruta, en el botón correspondiente se da un clic y se elige **Agregar**, como se muestra en la figura:



Posteriormente se muestra la siguiente pantalla donde se puede empezar a capturar, existe un control interno que nos indica si la ruta ya ha sido capturada.

TESIS CON  
FALLA DE ORIGEN

<b>Ruta</b> 1906		
Razón Social Empresa		
Domicilio		<input type="button" value="Guardar"/>
Teléfono	Parque Vehículo Total 0	
<b>Mesa Constitutiva</b>		
Presidente		<input type="button" value="Salir"/>
Secretario		
Tesoroero		
<b>Mesa Directiva Actual</b>		
Presidente	Ya existe esa ruta	
Secretario	<input type="button" value="Aceptar"/>	
Tesoroero		
<b>Mesa Directiva Disidencia</b>		
Presidente		
Secretario		
Tesoroero		
Representante Legal		

Después de capturar toda la información requerida, se presiona **Guardar**.

Nos aparece un mensaje que indica que se ha agregado la información en el sistema y pregunta si se desea agregar otro registro o no. Si se acepta, limpia el formulario y lo deja listo para la siguiente captura; de lo contrario se cierra la ventana, mostrando nuevamente el menú principal.

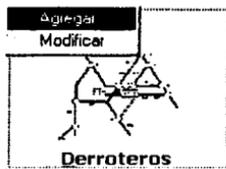
<b>Ruta</b> 201		
Razón Social Empresa	UNION DE CINCES ONARIOS DEL NORTE	
Domicilio	AVENIDA DE LAS TORRES 856	<input type="button" value="Guardar"/>
Teléfono	6721496 Parque Vehículo Total 0	
<b>Mesa Constitutiva</b>		
Presidente	ARTURO HERNANDEZ SANCHEZ	<input type="button" value="Salir"/>
Secretario	ARCE LA BERNAL DIAZ	
Tesoroero		
<b>Mesa Directiva</b> Se ha agregado al sistema, ¿Desea agregar otro registro?		
Presidente	<input type="button" value="Si"/> <input type="button" value="No"/>	
Secretario		
Tesoroero	FLOR HERNANDEZ LORTIZ	
<b>Mesa Directiva Disidencia</b>		
Presidente	ANGELICA NIETO HERNANDEZ	
Secretario	JUAN MOY LARA	
Tesoroero	ERNESTO MONTALVO GARCIA	
Representante Legal	ANTONIO HERNANDEZ GONZALEZ	

TESIS CON  
 FALTA DE ORIGEN

En el caso de presionar el botón **Cerrar**, no se guarda la información, sólo se cierra la ventana y nos envía nuevamente al menú principal

## B.2.2 Derrotero

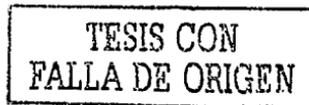
Para agregar un Derrotero, se da clic en el submenú **Agregar**.



Posteriormente se muestra pantalla donde se puede empezar a capturar:

Ruta	<input type="text"/>	<input type="checkbox"/> <b>RECORRIDO IDA</b>
Ramal	<input type="text"/>	
Base Origen	<input type="text"/>	
Eliminar los Base Origen	<input type="text"/> <b>RECORRIDO IDA</b>	
Base Destino	<input type="text"/>	
Delegaciones Base Destino	<input type="text"/> <b>RECORRIDO IDA</b>	Método de Cálculo <input type="text"/>
Asesoría	<input type="text"/>	<input type="checkbox"/> Eliminar Renglón
Superficie en Legal	<input type="text"/>	Fecha de Actualización <input type="text"/>
Delegado del Ramal	<input type="text"/>	<input type="checkbox"/> IMPRIMIR

Es obligatorio capturar la ruta y el ramal, de lo contrario aparecen mensajes como el siguiente:



Al presionar el botón **Recorrido Ida**, se muestra la ventana donde se puede capturar su trayectoria de ida. Después de capturar una trayectoria, es necesario presionar el botón **Agregar**, de esta forma aparece en la tabla de abajo, cuya información es la que es guardada posteriormente.

Si se desea eliminar el registro se pulsa el botón **Eliminar Renglón**. Al terminar la captura se presiona el botón **Cerrar**.

Formulario de configuración de recorridos. Incluye campos para "Recorrido", "Inicio", "Destino", "Alfabeto", "Doble", "Delegación", "Agregar", "Eliminar Recorrido", y "Eliminar Recorrido".

Radio buttons:  Autorizado  No autorizado

Plazo	Doble	Alfabeto	Delegación
1 BUENAVISTA	Derecha	AVEN. DANIEL F. RIC	Banco Juárez
2 CALLE ZAPALDOZA	Izquierda	YRABÁ	Banco Juárez

Botón: Cerrar

Lo mismo pasa con el botón **Recorrido Regreso**, en donde se muestra otra ventana similar a la anterior, con las mismas especificaciones de uso.

Formulario de configuración de recorridos de regreso. Incluye campos para "Recorrido", "Inicio", "Destino", "Alfabeto", "Doble", "Delegación", "Agregar", "Eliminar Recorrido", y "Eliminar Recorrido".

Radio buttons:  Autorizado  No autorizado

Plazo	Doble	Alfabeto	Delegación
1 SAN COSME	Centros	CALLE SAN O DOMINGO	Banco Juárez
2 REP. URUGUAY	Derecha	CALLE SAN LUIS POTOSÍ	Banco Juárez

Botón: Cerrar

TESIS CON FALLA DE ORIGEN

Es importante aclarar que en este momento todavía **NO SE HA GUARDADO LA INFORMACIÓN**, esto se lleva a cabo al presionar el botón **Guardar** de la ventana de **Derrotero**, en donde se muestra el siguiente mensaje:

SE I HA VI

Se ha agregado al sistema. ¿Decea agregar otro derrotero?

Botones: SI No

Si se acepta, limpia el formulario y lo deja listo para la siguiente captura; de lo contrario se cierra la ventana, mostrando nuevamente el menú principal.

En el caso de presionar el botón **Cerrar** de la ventana Derrotero, no se guarda la información (tanto de derrotero, como recorridos), sólo se cierra la ventana y nos envía nuevamente al menú principal.

### B.2.3 Padrón Vehicular

Para agregar un vehículo al Padrón Vehicular, se da clic en el submenú **Agregar**:



Aparece la siguiente pantalla de captura, donde es necesario escribir la placa del vehículo, así como la ruta a la que pertenece, de lo contrario nos manda mensajes indicando lo que falta.

Padrón Vehicular

Número de Placas: [ ] Ruta: [ ]

Rennales Disponibles: [ ] Rennales Elegidos: [ ]

Tipo de Placas: [ ] Marca: [ ]

Clase: [ ] **Falta Agregar el Número de Placa** **Agregar** Número Económico: [ ]

Propietario: [ ]

Dirección: [ ]

Teléfono: [ ] Delegación: [ ] Código Postal: [ ]

[Agregar] [Cerrar]

Padrón Vehicular

Número de Placas: [ICPWS] Ruta: [ ]

Rennales Disponibles: [ ] Rennales Elegidos: [ ]

Tipo de Placas: [ ] Marca: [ ]

Clase: [ ] **Falta Agregar la Ruta** **Agregar** Número Económico: [ ]

Propietario: [ ]

Dirección: [ ]

Teléfono: [ ] Delegación: [ ] Código Postal: [ ]

[Agregar] [Cerrar]

TESIS CON FALLA DE ORIGEN

Después se puede capturar toda la información requerida, cabe mencionar que el vehículo puede pertenecer a varios ramales, luego se presiona el botón **Agregar**.

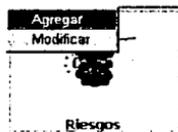
Nos aparece un mensaje que nos indica que se ha agregado la información en el sistema y pregunta si se desea agregar otro registro o no. Si se acepta, limpia el formulario y lo deja listo para la siguiente captura; de lo contrario se cierra la ventana, mostrando nuevamente el menú principal.

The screenshot shows a software interface for vehicle registration. At the top, there are fields for 'Número de Placas' (Plate Number) with the value '471311' and 'Ruta' (Route) with the value '4'. Below this are two lists of 'Ramales Esportivos' (Sportive Branches) and 'Formas de Pagos' (Payment Methods). A modal dialog box titled 'SE TRAV...' is centered on the screen, asking 'Registro Agregado. ¿Desea agregar otro Registro?' (Record Added. Do you want to add another record?). It has 'Si' (Yes) and 'No' (No) buttons. The background form includes fields for 'Tipo de Placa' (Plate Type) set to 'Motos', 'Clase' (Class) set to 'Comun', 'Propietario' (Owner) 'ALEJANDRO CORPE', 'Dirección' (Address) 'AV. CAROL SUR 21205. CEP. VALLE', 'Teléfono' (Phone) '876622', 'Delegación' (Delegation) 'Pinar del Río', and 'Código Postal' (Postal Code) '51140'. There are 'Agregar' (Add) and 'Cerrar' (Close) buttons at the bottom.

En el caso de presionar el botón **Cerrar**, no se guarda la información, sólo se cierra la ventana y nos envía nuevamente al menú principal.

### B.2.4 Riesgo

Para agregar un Riesgo, se da clic en el submenú **Agregar**:



Aparece la siguiente pantalla de captura, donde es necesario escribir la ruta, las placas del vehículo y el derrotero al que pertenece, de lo contrario nos manda mensajes indicando lo que falta.

TESIS CON  
FALLA DE ORIGEN

Ruta	14	Piezas		
Denominero				
Hora del Suceso	12:00 a.m.	RETRAV	13/12/2007	
Tipo de Riesgo	Falta Agregar las Piezas			Agregar
Numero de Orden	Aceptar			Cerrar
Descripcion				
Operador				

Ruta	14	Piezas	1471576	
Denominero				
Hora del Suceso	12:00 a.m.	RETRAV	13/12/2007	
Tipo de Riesgo	Falta Agregar un Denominero			Agregar
Numero de Orden	Aceptar			Cerrar
Descripcion				
Operador				

Después de capturar todos los datos, se presiona el botón **Agregar** y aparece un mensaje que nos indica que se ha agregado la información en el sistema y pregunta si se desea agregar otro registro o no. Si se acepta, limpia el formulario y lo deja listo para la siguiente captura; de lo contrario se cierra la ventana, mostrando nuevamente el menú principal.

Ruta	14	Piezas	1471576	
Denominero	SANTAMARTA CENTRAL DE ANALISIS			
Hora del Suceso	12:00 a.m.	RETRAV	13/12/2007	
Tipo de Riesgo	14	Registro Agregado, ¿Desea agregar otro Registro?		Agregar
Numero de Orden	15197	Si No		Cerrar
Descripcion	OPERA CON UN VENTILADOR ROTARIO			
Operador	MARCOS JUAN CARLOS PEREZ ALBA			

TESIS CON  
 FALLA DE ORIGEN

En el caso de presionar el botón **Cerrar**, no se guarda la información, sólo se cierra la ventana y nos envía nuevamente al menú principal.

## B.2.5 Apercibimiento

Para agregar un Apercibimiento, se da clic en el submenú **Agregar**:



Aparece la siguiente pantalla de captura. Después de capturar todos los datos, se presiona el botón **Agregar**, aparece un mensaje que nos indica que se ha agregado la información en el sistema y pregunta si se desea agregar otro registro o no. Si se acepta, limpia el formulario y lo deja listo para la siguiente captura; de lo contrario se cierra la ventana, mostrando nuevamente el menú principal.

 A screenshot of the 'Apercibimientos' data entry form. The form includes fields for 'Ruta' (14), 'Derrotorio' (SANTA MARTA - FANAJ DE GARAY), 'Numero de Utico' (144E5D), and 'Fecha' (05/12/2001). A dialog box is overlaid on the form, asking 'Registro Agregado, ¿Desea agregar otro Registro?' with 'Si' and 'No' buttons. At the bottom of the form are 'Agregar' and 'Cerrar' buttons.

En el caso de presionar el botón **Cerrar**, no se guarda la información, sólo se cierra la ventana y nos envía nuevamente al menú principal.

## B.2.6 Queja

Para agregar una Queja, se da clic en el submenú **Agregar**:



Aparece la siguiente pantalla de captura, donde es necesario escribir la ruta, las placas del vehículo y el derrotorio al que pertenece, los dos últimos pueden ser opcionales en el caso que se desconozca esa información, en este caso se elige NINGUNO.

TESIS CON  
 FALLA DE ORIGEN

<b>Clientes</b>	
Estado	Departamento
Plaza	Numero de Cliente
Apellido	Fecha
Demandante	
Ubicación	
Atendido	<input type="checkbox"/> Si
	<b>Agregar</b>
	<b>Cerrar</b>

Después de capturar todos los datos, se presiona el botón **Agregar** y aparece un mensaje que nos indica que se ha agregado la información en el sistema y pregunta si se desea agregar otro registro o no. Si se acepta, limpia el formulario y lo deja listo para la siguiente captura; de lo contrario se cierra la ventana, mostrando nuevamente el menú principal.

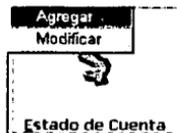
Estado	Departamento
Plaza	Numero de Cliente
Apellido	Fecha
Demandante	
Ubicación	
Atendido	<input type="checkbox"/> Si
	<b>Agregar</b>
	<b>Cerrar</b>

Registro Agregado, ¿Desea agregar otro Registro?

En el caso de presionar el botón **Cerrar**, no se guarda la información, sólo se cierra la ventana y nos envía nuevamente al menú principal.

### B.2.7 Estado de Cuenta

Para agregar un nuevo Estado de Cuenta, se da clic en el submenú **Agregar**:



Aparece la siguiente pantalla de captura, en el cuadro de **Total** sólo se escriben números de lo contrario manda un mensaje de error.

TESIS CON  
FALLA DE ORIGEN

Ruta : 201

Año 1984

Referencia :

Cantidad Total Incorrecta

Aceptar

Después de capturar todos los datos, se presiona el botón **Agregar**, aparece un mensaje que nos indica que se ha agregado la información en el sistema y pregunta si se desea agregar otro registro o no. Si se acepta, limpia el formulario y lo deja listo para la siguiente captura; de lo contrario se cierra la ventana, mostrando nuevamente el menú principal.

Ruta : 201

Registro Agregado, ¿Desea agregar otro Registro?

Si No

En el caso de presionar el botón **Cerrar**, no se guarda la información, sólo se cierra la ventana y nos envía nuevamente al menú principal.

TESIS CON  
FALLA DE ORIGEN

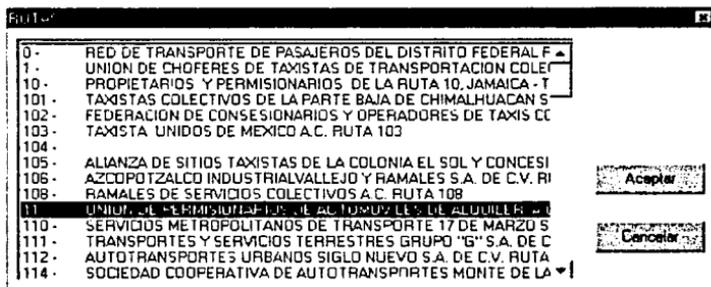
## B.3 MODIFICAR INFORMACIÓN

### B.3.1 Ruta

Para modificar una Ruta, en el botón correspondiente se da un clic y se elige **Modificar**, como se muestra en la figura:



Aparece una ventana donde se listan las rutas existentes, de entre las cuales se elige la que se desea modificar.



Al presionar el botón **Aceptar**, se muestra la ventana con la información de la ruta seleccionada, donde se pueden realizar los cambios pertinentes. Posteriormente se presiona el botón **Guardar**, y aparece en la pantalla un mensaje de aceptación.

TESIS CON  
FALLA DE ORIGEN

Ruta	11	
Razon Social Empresa	UNION DE PERMISIONARIOS DE AUTOMOVILES DE	
Domicilio	CALZ. ERMITA IZTAPALAPA No. 2588 COL. JACARANDAS	
Teléfono	5692-11-66	Parque Vehículo Total 10
<b>Mesa Constitutiva</b>		
Presidente	FILEMON VARGAS MAGUEY	<input type="button" value="Guardar"/>
Secretario	FILEMON NERIA ANGELES	
Tesoro	HAMON VARGAS <b>EL TRAM</b>	<input type="button" value="Salir"/>
Mesa Directiva Actual	Se ha Modificado la Ruta	
Presidente	JOSE JUAN SANI	<input type="button" value="Aceptar"/>
Secretario	JOSE ERIC TORI	
Tesoro	RAYMUNDO ALBARRAN MARTINEZ	
<b>Mesa Directiva Diferencia</b>		
Presidente	ANTONIO AYALA ORTIZ	
Secretario		
Tesoro		
Representante Legal	NICOLAS HERNANDEZ CASTILLO	

En el caso de presionar el botón **Salir**, no se guardan los cambios, sólo se cierra la ventana y nos envía nuevamente al menú principal.

### B.3.2 Derrotero

Para modificar un **Derrotero**, en el botón correspondiente se da un clic y se elige **Modificar**, como se muestra en la figura:



Aparece una ventana donde se listan las rutas existentes y sus derroteros, de entre los cuales se elige el que se desea modificar.

TESIS CON  
FALLA DE ORIGEN

0 - RED DE TRANSPORTE DE PASAJEROS DEL DISTRITO FEDERAL	Modificar Derrotero
1 - UNION DE CHOFERES DE TAXISTAS DE TRANSPORTACION COLECTIVA	
10 - PROPIETARIOS Y PERMISIONARIOS DE LA RUTA 10, JAMAICA - T	Cancelar
101 - TAXISTAS COLECTIVOS DE LA PARTE BAJA DE CHIMALHUACAN S	
102 - FEDERACION DE CONCESIONARIOS Y OPERADORES DE TAXIS CC	Modificar Recorrido Ida
103 - TAXISTA UNIDOS DE MEXICO A.C. RUTA 103	
104 -	Modificar Recorrido Regreso
105 - ALIANZA DE SITIOS TAXISTAS DE LA COLONIA EL SOL Y CONCESI	
106 - ACCOPOTZALCO INDUSTRIAL VALLEJO Y RAMALES S.A. DE C.V. RI	
109 - RAMALES DE SERVICIOS COLECTIVOS A.C. RUTA 108	
110 - UNIDAD DE SERVICIOS TAXI MUNICIPIALES DEL DISTRITO FEDERAL	
110 - SERVICIOS METROPOLITANOS DE TRANSPORTE 17 DE MARZO S	
111 - TRANSPORTES Y SERVICIOS TERRESTRES GRUPO "G" S.A. DE C	
112 - AUTOTRANSPORTES URBANOS SIGLO NUEVO S.A. DE C.V. RUTA	
114 - SOCIEDAD COOPERATIVA DE AUTOTRANSPORTES MONTE DE LA V	
AGRICOLA ORIENTAL CENTRAL DE ABASTOS	
BORDO DE XOCHIMILCO-CENTRAL DE ABASTOS	
BUENAVISTA SAN CUSME	
C. DE ABASTO - AGRICOLA ORIENTAL	
CENTRAL DE ABASTO-ORIENTAL-BALNEARIO OLIMPICO	
CONSTITUCION DE 1917 - BORDO DE XOCHIMILCO	
CONSTITUCION DE 1917 - M. CANAL DE S. JUAN	
CONSTITUCION DE 1917 - METRO PANTITLAN	
CONSTITUCION DE 1917 - METRO AEROPUERTO	
CONSTITUCION DE 1917 - METRO PINTO SUAREZ	
CONSTITUCION DE 1917 - METRO PUEBLA	
CTAPALAPA - CENTRAL DE ABASTOS	
CTAPALAPA-CANAL SAN JUAN	
METRO AEROPUERTO DEPORTIVO SANTA CRUZ	

Si se desea modificar la información del derrotero, se presiona el botón **Modificar Derrotero**. Después de realizar los cambios pertinentes, se presiona **Guardar**, desplegándose el siguiente mensaje:

SE TRABAJO

Se ha Modificado en el Sistema

Aceptar

Si se desea modificar la información de la trayectoria de Ida, se presiona el botón **Modificar Recorrido Ida**. Después de realizar los cambios pertinentes, se presiona **Guardar**.

TESIS CON  
FALLA DE ORIGEN

Autorizado  No autorizado

Parte	Se ha agregado al sistema	Delegación
CALLE MONTEVIDEO	<input type="checkbox"/>	Caceres
CALLE ZARAGOZA	<input type="checkbox"/>	San Juan

Si se desea modificar la información de la trayectoria de Regreso, se presiona el botón **Modificar Recorrido Regreso**. Después de realizar los cambios pertinentes, se presiona **Guardar**.

Autorizado  No autorizado

Parte	Se ha agregado al sistema	Delegación
CALLE MONTEVIDEO	<input checked="" type="checkbox"/>	Caceres
CALLE ZARAGOZA	<input checked="" type="checkbox"/>	San Juan

TESIS CON FALLA DE ORIGEN

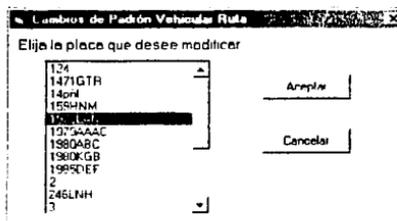
En el caso de presionar el botón **Cerrar**, no se guardan los cambios, sólo se cierra la ventana y nos envía nuevamente al menú principal.

### B.3.3 Padrón Vehicular

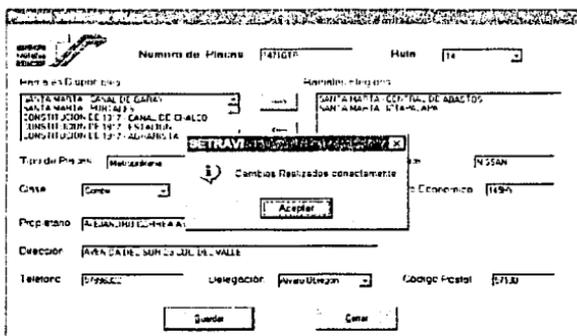
Para modificar un vehículo del **Padrón Vehicular**, en el botón correspondiente se da un clic y se elige **Modificar**, como se muestra en la figura:



Aparece una ventana donde se listan las placas existentes, de entre las cuales se debe elegir las del vehículo que se desea modificar.



Al presionar el botón **Aceptar**, se muestra la ventana con la información del vehículo seleccionado, donde se pueden realizar los cambios pertinentes. Posteriormente se presiona el botón **Guardar**, aparece en la pantalla un mensaje de aceptación.



En el caso de presionar el botón **Cerrar**, no se guardan los cambios, sólo se cierra la ventana y nos envía nuevamente al menú principal.

### B.3.4 Riesgo

Para modificar un **Riesgo**, en el botón correspondiente se da un clic y se elige **Modificar**, como se muestra en la figura:



TESIS CON  
FALLA DE ORIGEN

Aparece una ventana donde se listan las rutas existentes y se elige una, en el cuadro de abajo se muestran las placas de los vehículos de esa ruta, fecha y hora en que sucedió el acontecimiento, se selecciona el riesgo que se desea modificar.

Elija la ruta que desea modificar.

11
14
27
28
29
74
78

Elija la fecha y hora que desea modificar

Placa	Fecha	Hora
1978LAN	15/01/02	11:59
1978LAN	16/12/01	00:00

Al presionar el botón **Aceptar**, se muestra la ventana con la información del riesgo seleccionado, donde se pueden realizar los cambios pertinentes. Posteriormente se presiona el botón **Guardar**, aparece en la pantalla un mensaje de aceptación

Ruta: [ ] Placas: [ ]

Destinatario: [ ]

Hora del suceso: [ ]

Tipo de Furgón: [ ]

Número de Choque: [ ]

Descripción: [ ]

Operador: [ ]

Cambios Realizados correctamente

Aceptar Cerrar

En el caso de presionar el botón **Cerrar**, no se guardan los cambios, sólo se cierra la ventana y nos envía nuevamente al menú principal.

### B.3.5 Apercibimiento

Para modificar un Apercibimiento, en el botón correspondiente se da un clic y se elige **Modificar**, como se muestra en la figura:



TESIS CON  
FALLA DE ORIGEN

Aparece una ventana donde se listan las rutas existentes se elige una y en el cuadro de abajo se muestran la fecha y número de oficio de los apercibimientos realizados, se selecciona el apercibimiento que se desea modificar.

Elige la ruta que deseas modificar

Ruta: 104

Elige fecha y número de oficio

Fecha	Numero de Oficio
02/12/01	44EUS0

Aceptar

Cancelar

Al presionar el botón **Aceptar**, se muestra la ventana con la información del apercibimiento seleccionado, donde se pueden realizar los cambios pertinentes. Posteriormente después de realizar los cambios, se presiona el botón **Guardar**, apareciendo en la pantalla un mensaje de aceptación.

Ruta: 74

Dirección: SANTA MARTA - FAMILIA DE GARAY

Numero de Oficio: FABER0 Fecha: 05/02/00

Queja: SETRAV

Cambios Realizados correctamente

Aceptar

Guardar Cerrar

En el caso de presionar el botón **Cerrar**, no se guardan los cambios, sólo se cierra la ventana y nos envía nuevamente al menú principal.

### B.3.6 Queja

Para modificar un Queja, en el botón correspondiente se da un clic y se elige **Modificar**, como se muestra en la figura:



TESIS CON  
FALLA DE ORIGEN

Aparece una ventana donde se listan las rutas existentes, se elige una y en el cuadro de abajo se muestran la fecha y número de oficio de las quejas realizadas a la ruta seleccionada, se marca el apercibimiento que se desea modificar.

Al presionar el botón **Aceptar**, se muestra la ventana con la información de la queja seleccionada, donde se pueden realizar los cambios pertinentes. Posteriormente se presiona el botón **Guardar**, aparece en la pantalla un mensaje de aceptación.

En el caso de presionar el botón **Cerrar**, no se guardan los cambios, sólo se cierra la ventana y nos envía nuevamente al menú principal.

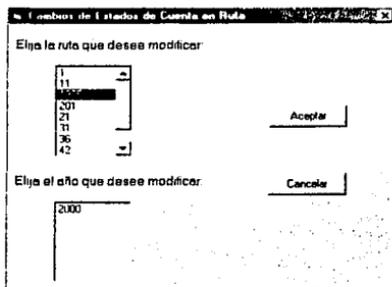
### B.3.7 Estado de Cuenta

Para modificar un Estado de Cuenta, en el botón correspondiente se da un clic y se elige **Modificar**, como se muestra en la figura:

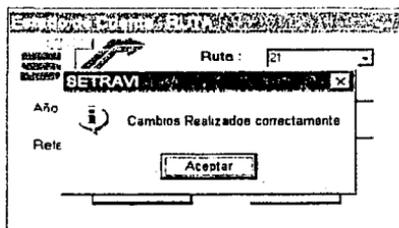


TESIS CON  
FALLA DE ORIGEN

Aparece una ventana donde se listan las rutas existentes, se elige una y en el cuadro de abajo se muestra el año de los estados de cuenta pertenecientes a la ruta seleccionada, se marca el año del estado de cuenta que se desea modificar.



Al presionar el botón **Aceptar** y se muestra la ventana con la información del estado de cuenta seleccionado, donde se pueden realizar los cambios pertinentes. Posteriormente se presiona el botón **Guardar** y aparece en la pantalla un mensaje de aceptación.



En el caso de presionar el botón **Cerrar**, no se guardan los cambios, sólo se cierra la ventana y nos envía nuevamente al menú principal.

#### B.4 ELIMINAR INFORMACIÓN

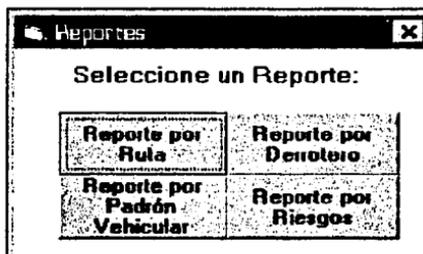
Se sigue el mismo procedimiento que el de modificar, la única diferencia es que el elemento seleccionado, será borrado del sistema.

Se elige la opción **Eliminar** desde el menú, luego se muestra la ventana donde se elige el elemento deseado y se muestra su contenido, para brindarle al usuario la oportunidad de revisar que realmente es el registro que desea borrar, posteriormente sólo se presiona el botón **Eliminar** y listo.

## B.5 REPORTES

En nuestra pantalla principal se encuentra el siguiente botón, que al presionarlo nos muestra un pequeño menú con los tipos de reportes con que cuenta el sistema.

**Informacion  
General de  
Transporte**



**TESIS CON  
FALLA DE ORIGEN**

**B.5.1 REPORTE POR RUTAS.**

En este reporte se detalla la información de la rutas registradas, con la opción de seleccionar, si están vigentes, si tienen representante legal o simplemente mostrar todas las rutas.

REPORTE DE RUTAS

Todas las Rutas
  Rutas Vigentes
 Rutas No Vigentes

Con Representante Legal

Ruta	Razón Social	Domicilio	Teléfono	Parque Veh	M_Const_P	M_Const_Si	M_Const_Ti	M_Directiva	M_Directiva	M_Directiva	M_D
21	UNION DE I	AV MEXICO			0			ADELAIDO	FLIBERTO	FODOLFO V	
25	UNION DE I				0			FRANCISCO	SALVADOR	JOSE TORF	
31	TRANSPOR	CALLE No.			0						
38	RUTA 111E			12				SALVADOR	JOSE EUSE	JUAN GAR	
42	ORGANIZA	AV CORON			0						
50	ASOCIACION	CALLE FIDE			0			EFRAIN	MIF		
51	UNION DE	AV. IGHACI			0	MIGUEL RA	GREGORIO	PABLO CIRI	HERMELINI	TOMAS OLI	ELIAS URB
62	SERVICIO C	QUINTANA			0			MARCOS H	CARLOS H	ALEJANDRI	
74	UNION DE				0	BARTOLOM	MARGARIT	ADOLFO R	INDENCIO	JUAN ORTI	JESÚS FRA
78	UNION DE I	HACIENDA	4785125		120	TOÑO	PACO	LUIS	LUISA	ERNESTO	ADRIANA ART
87	RUTA DE T	IZAMAL MZ			0						

Rutas Registradas: 15

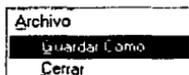
TESIS CON  
FALLA DE ORIGEN

## B.5.2 REPORTE DE DERROTOS POR RUTA

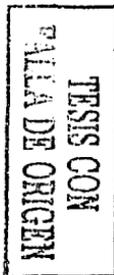
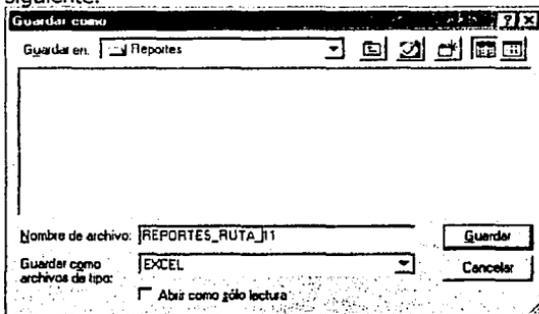
En este reporte se selecciona la ruta y automáticamente nos muestra cuántos y cuáles son los derroteros que integran dicha ruta.

DERROTOS			
Archivo			
RUTAS:			
24 Derroteros			
Itemal	no. oficio	fecha autorizacion	lanzadera
BORDO DE XOCHILACA-CENTRAL DE ABASTOS	DGAU/0303/90		26/07/90
BUENAVISTA-SAN COSME	123FR		16/01/02 BUENAVISTA FERF
BUENAVISTA-SAN COSME	123FR		16/01/02 BUENAVISTA
C. DE ABASTO - AGRICOLA ORIENTAL			PENDIENTE_CENTR
CENTRAL DE ABASTO-ORIENTAL-BALNEARIO OLIMPICO	DGAU/0303/90		26/07/90
CONSTITUCION DE 1917 - BORDO DE XOCHILACA			PENDIENTE_CENTR
CONSTITUCION DE 1917 - M. CANAL DE S. JUAN			PENDIENTE_CENTR
CONSTITUCION DE 1917 - METRO PANTITLAN			PENDIENTE_CENTR
CONSTITUCION DE 1917 - METRO AEROPUERTO			PENDIENTE_CENTR
CONSTITUCION DE 1917 - METRO PINO SUAREZ			PENDIENTE_CENTR
CONSTITUCION DE 1917 - METRO PUEBLA			PENDIENTE_CENTR
ICTAPALAPA - CENTRAL DE ABASTOS	DGAU/0303/90		26/07/90
ICTAPALAPA-CANAL SAN JUAN	DGAU/0303/90		26/07/90
METRO AEROPUERTO DEPORTIVO SANTA CRUZ	DGAU/0303/90		26/07/90
METRO AEROPUERTO-PANIFICADORA SAN FELIPE	DGAU/0303/90		26/07/90
METRO AEROPUERTO-PANIFICADORA SAN FELIPE	DGAU/0303/90		26/07/90
METRO PANTITLAN-CENTRAL DE ABASTOS	DGAU/0303/90		26/07/90
METRO ZARAGOZA-ORIENTAL-ICTAPALAPA	DGAU/0303/90		26/07/90
METRO ZARAGOZA-SAN FELIPE-TINACOS V GUERRERO	DGAU/0303/90		26/07/90
PANIFICADORA SAN FELIPE METRO PANTITLAN	DGAU/0303/90		26/07/90
SAN ANGEL - CENTRAL DE ABASTO	DGAU/1782/88		16/01/02
TINACOS-CENTRAL DE ABASTOS	DGAU/0303/90		26/07/90

Si se da clic en el menú **Archivo**, se despliegan las siguientes opciones:



Si se elige **Guardar Como**, permite guardar la consulta en una hoja de Excell. Como muestra la figura siguiente:

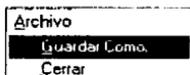


### B.5.3 REPORTE DEL PADRÓN VEHICULAR POR RUTA

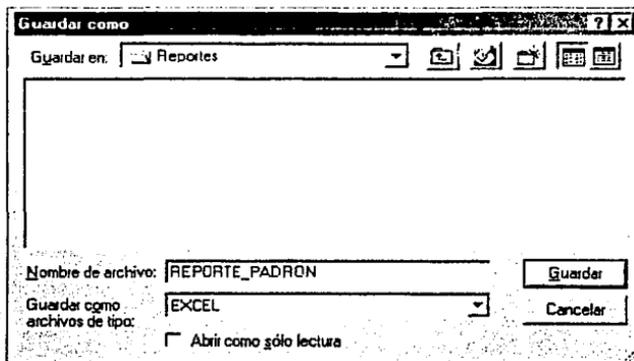
En este reporte se selecciona la ruta y automáticamente nos muestra cuántos y cuáles son los vehículos que integran dicha ruta.

PDRÓN VEHICULAR RUTA									
Archivo									
RUTA:									
Ruta 11 - UNION DE PERMISIONARIOS DE AUTOMOVILES DE ALQUILER, A.C.									
3 Vehiculos									
placa	tipo_placa	modelo	marca	clase	numero_economico	conv_gas	propietario	telefono	direccion
153HNM	Metropolitana	2002	MMM	Autobus	87LD	Gas	JJJJJ	54121	KKKKKK
158DABC	Federal	2002	tsuru	Autobus	145TF	Convertidor	lluna	14551211	vale
5645TF	Federal	2002	Ford	Autobus	1	Convertidor	lluna	14551211	5645TF

Si se da clic en el menú **Archivo**, se despliegan las siguientes opciones:



Si se elige **Guardar Como**, permite guardar la consulta en una hoja de Excell. Como muestra la figura siguiente:



TESIS CON  
 FALTA DE ORIGEN

### B.5.4 REPORTE DE RIESGOS POR RUTA

En este reporte se selecciona la razón social de la ruta, se presiona el botón **Aceptar** y nos muestra cuántos y cuáles son los riesgos registrados en esa ruta. También se puede ver la lista de riesgos de los últimos 30 días.

RIESGOS POR RUTA						
Razon Social o Empresa						
TRANSPORTES COLECTIVOS DEL SURESTE DEL DISTRI		Aceptar		<input type="checkbox"/> Ver lista de riesgos de los últimos 30 días		
razon social	fecha del accidente	hora	razon social empresa	numero	razon social empresa	id
<input checked="" type="checkbox"/> TRANSPORTES COLE	06/12/2001	06:00	1973AAC	1978	145	NORMA ALFARO VILLI G
<input type="checkbox"/> TRANSPORTES COLE	16/12/2001	22:00	987ABC	765RF	98	0000000
<input type="checkbox"/> TRANSPORTES COLE	18/12/2001	00:00	987ABC	765RF	784	HHHHH

No. de riesgos registrados:

 Imprimir

Se tiene la opción de imprimir. El siguiente es un ejemplo de reporte impreso:

TESIS CON  
 FALLA DE ORIGEN

---

**Riesgos por Ruta**Jueves, 25 de Julio de 2002

---

**Razon Social o Ruta:** TRANSPORTES COLECTIVOS DEL SURESTE DEL DISTRITO FEDERAL, A.C.  
**Dia del acontecimiento:** 06/12/2001      **Hora del acontecimiento:** 08:00  
**Placa:** 1979AAAC      **Numero económico:** 1978  
**Tipo de riesgo:** 145  
**Operador:** NORMA ALFARO VILLAGOMEZ      **Delegación:** Gustavo A. Madero  
**Descripción:** CHOQUE TOTAL  
**Oficio:** KIJU879

---

**Razon Social o Ruta:** TRANSPORTES COLECTIVOS DEL SURESTE DEL DISTRITO FEDERAL, A.C.  
**Dia del acontecimiento:** 16/12/2001      **Hora del acontecimiento:** 22:00  
**Placa:** 987ABC      **Numero económico:** 765RF  
**Tipo de riesgo:** 98  
**Operador:** QQQQQQQ      **Delegación:** Coyoacán  
**Descripción:** OOOOO  
**Oficio:** 744OLJK

---

TESIS CON  
FALLA DE ORIGEN