

**UNIVERSIDAD MAYOR DE SAN ANDRES**  
**FACULTAD DE CIENCIAS PURAS Y NATURALES**  
**CARRERA DE INFORMATICA**



**PROYECTO DE GRADO**

Sistema de Control y Administración

**"SARIRI"**

Bolivian Bus

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

**AUTOR:** *Juan Carlos Paredes Mamani*  
**TUTOR:** *Ms. Sc. Mario Loayza Molina*  
**REVISOR:** *Lic. José Luís Zeballos Abasto*

La Paz – Bolivia  
2007

## **DEDICATORIA**

A mis padres, hermanos y sobrinas...

Sin su apoyo y tolerancia nada de esto  
sería posible.

## AGRADECIMIENTOS

Gracias por todo el apoyo...

Gracias a los que han estado ahí desde el primer día...

Hoy me siento afortunado de poder decirlo y convertirlo en palabras para que así  
puedan sentirlo.

Hoy quiero dar las gracias por existir, por sonreír y celebrar que aun queda mucho  
por venir.

Hoy quiero decir gracias por vivir.

Gracias por un día soleado donde poderme divertir.

Gracias por tener conciencia, tener el don de hacer ciencia que aboga por la no  
violencia.

Gracias a unos padres bondadosos que me arroparon con besos, hoy voy a serlos  
sentir orgullosos.

Y a mis hermanos por su confianza, por enseñarme que la base es el trabajo y la  
constancia.

No los defraudaré y les demostraré, que el pequeño ahora es la gran esperanza.

Gracias por no caer en la maldad y en la ignorancia, por nacer en tierra de luz y  
abundancia.

Gracias por mis momentos de paz sin malas sorpresas, por una cama y un plato  
lleno en mi mesa.

Gracias UMSA por educarme.

Gracias Maria Eugenia por estar aquí y acompañarme.

Gracias por quejarme solo de problemas mínimos y a mis amigos por su apoyo y  
sus ánimos gracias...

## RESUMEN

El presente trabajo contiene el proceso de desarrollo del sistema de información computarizado implementado en la empresa de transporte de pasajeros internacional "BOLIVIAN BUS" con el fin de procurar la confianza total de la información para mejorar el desarrollo y control de las actividades de la empresa.

En el primer apartado se presenta una descripción de la empresa y referencias a trabajos similares desarrollados anteriormente; además se expresan los problemas encontrados dentro la empresa y los objetivos propuestos para dar solución a los mismos. También se presentan las justificaciones para realizar el presente trabajo junto con los alcances, las metodologías utilizadas y los aportes que el sistema desarrollado brindará a la empresa.

Dentro el segundo capítulo se presentan el Marco Teórico y el Marco Conceptual; se realiza una descripción del Análisis Estructurado Moderno propuesto por Edward Yourdon, junto con la Metodología de Administración de Relaciones (RMM) ideado por Isakowitz, Stohr y Balasubramanian, y su extensión, el Modelo de Administración de Relaciones Dinámicas (DRM). También se introduce al ISO/IEC 9126 Tecnologías de la información – Calidad de los Productos de Software.

En el tercer capítulo se aplican las metodologías propuestas en el Marco Referencial, a este nivel ya se tiene desarrollado el producto de software (Sistema de Control y Administración "SARIRI") y se realiza su respectiva implementación y pruebas dentro la empresa, dicho software esta desarrollado en un entorno grafico utilizando herramientas como es el Visual FoxPro de Microsoft y el servidor de base de datos MySQL con su driver MyODBC.

Finalmente en el último apartado de éste trabajo se manifiestan las conclusiones a las cuales se llegan al final de todo el proceso de de pruebas y liberación para explotación del software desarrollado junto con algunas recomendaciones para la ampliación y mejora del mismo.

## Índice de Contenidos

Resumen

### Capítulo I: Introducción

1.1	Introducción	10
1.2	Antecedentes	11
1.2.1	De la Empresa	11
1.2.2	Del Sistema	12
1.3	Planteamiento del problema	12
	Formulación del Problema	13
1.4	Objetivos	14
1.4.1	Objetivo general	14
1.4.2	Objetivos específicos	14
1.5	Justificación	15
1.5.1	Justificación técnica	15
1.5.2	Justificación económica	16
1.5.3	Justificación social	16
1.6	Alcances	16
1.7	Metodología	17
1.8	Aportes	18

### Capítulo II: Marco Referencial

2.1	Marco teórico	19
2.1.1	Análisis Estructurado Moderno	19
2.1.1.1	Análisis de Sistemas	19
2.1.1.2	Modelo estructurado	20
2.1.1.3	Modelo Esencial	20
2.1.1.3.1	Modelo Ambiental	21

2.1.1.3.2 Herramientas usadas para definir el ambiente	21
Declaración de Propósitos	22
Diagrama de Contexto	22
Lista de Acontecimientos	23
2.1.1.3.3 Modelo de Comportamiento	23
Diagramas de Flujo de Datos	24
Proceso	24
Flujo	25
Almacén	26
Terminador	26
Guía para la construcción de un DFD	27
Diccionario de Datos	27
Notación del Diccionario de Datos	28
Especificación de Procesos	28
Herramientas de la especificación de Procesos	28
Lenguaje Estructurado	29
2.1.1.4 Diagrama Entidad-Relación	31
Entidades	31
Relaciones	32
Atributos	33
2.1.1.5 Diagrama Entidad-Relación Extendido	33
Entidades Fuertes y Débiles	33
Cardinalidad de las relaciones	34
Atributos en las relaciones	35
Herencia	35
2.1.1.5.1 Modelado Entidad-Relación	35
2.1.2 Metodología RMM	36
2.1.2.1 El modelo RMDM	36
2.1.2.2 Etapas de la metodología	41
Etapa 0	41
Etapa 1: Diseño Entidad-Relación	41

Etapa 2: Diseño de Slices	42
Etapa 3: Diseño navegacional	42
Etapa 4: Diseño de protocolos de conversión	44
Etapa 5: Diseño de la Interfaz de Usuario	44
Etapa 6: Diseño de comportamiento en tiempo de ejecución	44
Etapa 7: Construcción y Pruebas	45
2.1.2.3 Deficiencias del modelamiento conceptual RMM y Propuesta de Solución: Modelo DRM	45
2.1.3 Calidad del Software	51
2.1.3.1 Definiciones de Calidad de Software	51
2.1.3.2 Diferentes aspectos de la Calidad	52
2.1.3.3 ISO/IEC 9126: Tecnologías de la Información – Calidad de los Productos de Software	52
2.1.3.4 Modelo de Calidad: Para la calidad Interna y Externa	53
Funcionalidad	53
Fiabilidad	54
Usabilidad	55
Eficiencia	55
Mantenibilidad	56
Portabilidad	56
2.2 Marco conceptual	59
2.2.1 Tecnologías Web	59
2.2.1.1 Internet	59
2.2.1.2 Protocolos de Internet	59
Protocolo TCP/IP	59
Protocolo HTTP	60
2.2.1.3 Navegador	60
2.2.1.4 Servidor Web	60
2.2.1.5 Sistema Cliente-Servidor	61
2.2.2 Sistema Gestor de Base de Datos	61
MySQL	62

Conexión ODBC	62
2.2.3 Lenguajes	63
HTML	63
Javascript	63
PHP	64
SQL	64
Visual FoxPro	65
2.2.4 Control de Acceso	65
Autenticación	66
<b>Capítulo III: Análisis y Diseño</b>	
3.1 Estructura organizacional	68
3.2 Descripción de funciones	69
Propietarios	69
Gerente General	69
Personal de Operaciones	69
Administrador de Sucursal	69
Vendedor de Pasajes	70
3.3 Análisis del Sistema a Desarrollar	70
3.3.1 Modelo Ambiental	70
3.3.1.1 Declaración de Propósitos	70
3.3.1.2 Diagrama de Contexto	71
3.3.1.3 Lista de Acontecimientos	71
3.3.2 Modelo de Comportamiento	73
3.4 Diseño del Sistema	83
3.4.1 Diseño de Almacenamiento	83
3.5 Modelo Web del Sistema	84
Etapa 1: Diseño Entidad-Relación	84
Etapa 2: Diseño de Slices	85
Etapa 3: Diseño navegacional	87



Etapa 4: Diseño de protocolos de conversión	89
3.6 Recurso de Hardware y Software	89
3.6.1 Recursos de Hardware	89
3.6.2 Recursos de Software	90
Aplicación WIN32	90
Aplicación Web	90
3.7 Interfaz de Usuario	91
Aplicación WIN32	92
Aplicación Web	98
3.8 Medidas de Calidad	104
3.8.1 Funcionalidad	104
3.8.2 Fiabilidad	106
3.8.3 Usabilidad	107
3.8.4 Eficiencia	108
3.8.5 Mantenibilidad	109
3.8.6 Portabilidad	110
<b>Capítulo IV: Conclusiones y Recomendaciones</b>	
4.1 Conclusiones	111
4.2 Recomendaciones	112
<b>REFERENCIA BIBLIOGRAFICA</b>	
<b>ANEXOS</b>	
Anexo A	
Anexo B	
Anexo C	
Anexo D	
<b>DOCUMENTOS</b>	



## Capítulo I: Introducción

### 1.1 INTRODUCCION

La Internet tiene un impacto profundo en el trabajo, el ocio y el conocimiento. La Web ha permitido una descentralización repentina y extrema de la información y de los datos. Gracias a la Web, millones de personas tienen acceso fácil e inmediato a una cantidad extensa y diversa de información en línea.

La Internet ha impulsado el fenómeno de la Globalización y junto con la llamada desmaterialización de la economía ha dado lugar al nacimiento de una Nueva Economía caracterizada por la utilización de la red en todos los procesos de incremento de valor de la empresa, así de esta manera las empresas que se encuentran fragmentadas en diferentes ciudades y países optan por la utilización de Internet para incrementar sus utilidades, mejorar su administración y tener un mayor control sobre sus unidades operacionales, en este sentido, *"BOLIVIAN BUS"* una empresa de transporte de pasajeros internacional desea implementar un sistema que colabore con todos los aspectos ya mencionados, aprovechando junto con el Internet, los avances tecnológicos referentes a servidores Web, servidores de base de datos, páginas Web dinámicas, etc.

Por ello el siguiente proyecto esta orientado en la elaboración de un sistema de control y administración que combine tanto tecnologías actuales como métodos de desarrollo de sistemas de información como son el "Análisis Estructurado Moderno" y la "Metodología de Administración de Relaciones" junto con su extensión, la "Metodología de Administración de Relaciones Dinámicas"; que colabore con acciones básicas como son la venta de pasajes, asignación de destinos, recopilación de información distribuida por todas sus sucursales de venta de pasajes, elaboración de informes de ingresos, etc.

## 1.2 ANTECEDENTES

### 1.2.1 De la empresa

*BOLIVIAN BUS* es una empresa de transporte de pasajeros y encomiendas internacional conformada por un grupo de cuatro inversionistas bolivianos.

A mediados de 2006 se conforma la empresa, comenzando sus actividades en noviembre del mismo año con una cantidad de 4 omnibuses VOLVO cubriendo las rutas

*La Paz – Arica – Iquique – Arica - La Paz*  
*Cochabamba – Arica – Iquique – Arica - Cochabamba*

Esta empresa cuenta con una oficinas establecida en la ciudad de La Paz y cuatro sucursales de venta de pasajes y recepción de encomiendas ubicadas en: El Alto, La Paz, Arica e Iquique, cada una de ellas brindan sus servicios en las terminales de buses respectiva.

Los viajes, se realizan en horarios diurnos todos los días. Asimismo, la atención al público es ininterrumpida hasta cumplir con la última salida del día.

Los buses con los que cuentan actualmente, cumplen perfectamente los requerimientos para este tipo de viajes, brindando comodidad a los pasajeros, ya que cuentan con calefacción, aire acondicionado, servicio a bordo, música y video para hacer más entretenido el viaje además que cada bus cuenta con servicio de baño.

Actualmente la empresa "*BOLIVIAN BUS*", dirige y desenvuelve sus actividades mediante un presidente de grupo (gerente general), y en lo que respecta a la

administración del mismo no tiene implementado un sistema de información computarizado.

### 1.2.2 Del sistema

En la carrera de Informática de la Universidad Mayor de San Andrés (UMSA) se verificó la existencia de proyectos de grado relacionados con el sistema de control y administración de empresas de servicio de transporte, dichos proyectos están orientados a la automatización de los procesos manuales y la integración de la información.

Se verificó los siguientes proyectos de grado:

- Sistema de información de control y administración de empresa de transporte. Caso: "CISNE TOURS", realizado por la universitaria Norah Giovanna Jaliri Martínez.
- Sistema de información para el control y administración del sindicato "VILLA VICTORIA", realizado por la universitaria María Cecilia Vásquez Coronel.
- Sistema de información integrado de control y administración de empresas de transporte. Caso: "Trans Illimani", realizado por la universitaria Gregoria Nancy Blanco Acarapi.

### 1.3 PLANTEAMIENTO DEL PROBLEMA

Frente a las dificultades que se presentan en empresas de transporte de pasajeros, con sus sucursales de venta de pasajes en diferentes lugares y alejados entre sí, que hace que la recopilación de la información (para informes, sobre utilidades, sobre venta de pasajes, ubicación de los buses, etc.) sea dificultosa, el administrador no puede llevar un control pleno de la empresa (de los buses, de la venta de pasajes, de los ingresos por la venta de los mismos,

etc.), tiene dificultades en comunicar información necesaria a sus sucursales de venta de pasajes (principalmente información sobre destinos, buses asignados, horarios, entre otros), y los problemas a nivel sucursales se deben al factor distancia, por que debido a la inexistencia de una buena comunicación entre las mismas, la información que manejan sobre disponibilidad de asientos no esta actualizada en todo momento, esto lleva a una posible venta de pasajes duplicados, también el problema de la administración de la información de las encomiendas, realizado manualmente, puede presentar cualquier error, mismo que puede alcanzar grandes magnitudes, por estos y otros motivos se debe implementar un sistema de información computarizado que sea capaz de permitir un control de las actividades concernientes a la venta y emisión de pasajes, así como la captura y almacenamiento de la información sobre encomiendas (por parte de los vendedores dentro las sucursales de venta de pasajes) de manera eficiente y eficaz, así mismo, permita una administración de los recursos disponibles para una mejor operación de la empresa, dicho sistema debe estar diseñado en un entorno de red para hacer frente al problema de la dispersión de la información, misma que es necesaria e importante para los usuarios finales.

#### **FORMULACION DEL PROBLEMA**

*¿El sistema de información computarizado para la empresa de transporte de pasajeros internacional BOLIVIAN BUS, será capaz de capturar, almacenar y proporcionar información precisa de manera oportuna e integra y así de esta manera mejorar las actividades concernientes al control y administración de los recursos de información de la empresa de forma centralizada?*

## 1.4 OBJETIVOS

### 1.4.1 OBJETIVO GENERAL

Diseñar, desarrollar e implementar un sistema de información computarizado que interactúe en un entorno de red y centralice el control y la administración de los recursos de información en general de la Empresa de Transporte de Pasajeros Internacional "BOLIVIAN BUS".

### 1.4.2 OBJETIVOS ESPECIFICOS

- Aplicar metodologías de análisis, diseño de sistemas de información.
- Diseñar una base de datos centralizada que distribuya la información entre las unidades operacionales de la empresa.
- Automatizar el seguimiento de los buses para llevar un mejor control de los horarios de los mismos para ofrecer un mejor servicio a los clientes de la empresa.
- Diseñar una nomina de pasajeros que cumpla con normal internacionales de control.
- Diseñar procesos de control en línea sobre asignación de asientos.
- Automatizar y facilitar las operaciones de control y de administración de pasajes y encomiendas.
- Diseñar reportes y listados para incrementar y mejorar el control de recursos.

## 1.5 JUSTIFICACION

Para solucionar los problemas de la empresa de transportes de pasajeros internacional "BOLIVIAN BUS", se plantea un sistema de información para el control y administración de la misma, dicho software será producto de la utilización de herramientas de análisis, diseño y desarrollo de sistema de información, el mismo que debido a la naturaleza de su creación sirve para solucionar problemas puntuales encontrados en la institución.

### 1.5.1 JUSTIFICACION TECNICA

Con el software se logra la integración de la administración de la empresa, las ventas, la contabilidad, la administración de los buses, los recursos humanos entre otros. Con la ayuda de tecnologías tanto en lenguajes de programación (HTML, JAVASCRIPT, PHP, Visual FoxPro, etc.), como recursos software (Servidores Web, Servidores de base de datos, etc.) y recursos de hardware (Computadoras personales, servidores, etc.), se aprovechan las virtudes del Internet ayudando a incrementar la efectividad en las operaciones de la empresa.

Así de esta manera se adquieren ventajas competitivas como: la reducción de costos y tiempo en las operaciones administrativas de intercambio y recolección de información, en la interconexión de los usuarios y administradores, en el reemplazo de documentos físicos por electrónicos, eliminando errores por manipulación (extravío, deterioro) y re-digitación, en la eliminación de la barrera de la distancia ya que se puede trabajar con un mismo sistema en puntos distantes, mejora en la disponibilidad de la información para todos los usuarios por ser en tiempo real.

### 1.5.2 JUSTIFICACION ECONOMICA

Con la ayuda de tecnologías de hardware y software que se integran para lograr un sistema en red, el software logra un mejor control de ingresos tanto por el concepto de venta de pasajes como por el servicio de encomiendas.

Se aprovechan las facilidades del Internet en cuanto a comunicación entre las sucursales de ventas y el administrador con costos realmente bajos, reduciendo de esta manera los gastos en los que se incurre por concepto de comunicación, de supervisión de sucursales, de errores de liquidación de ingresos por pasajes y encomiendas, de errores de pago de impuestos.

### 1.5.3 JUSTIFICACION SOCIAL

Con una interfaz diseñada y desarrollada en un entorno grafico, el software mejora la imagen presentada ante los usuarios de los servicios de la empresa, colabora con captura de información ofreciendo una mejor entorno de trabajo, facilitando las operaciones que realizan los vendedores de pasajes en las sucursales, ayuda a una mejor administración de los recursos de la empresa por parte del administrador facilitando las tareas de asignación de destinos, liquidación de ingresos entre otros.

### 1.6 ALCANCES

Este proyecto concluirá con la entrega del producto de software "SARIRI" (Sistema de Control y Administración) de forma completa a la institución (BOLIVIAN BUS), es decir, la aplicación ejecutable (instalador), código fuente del mismo, documentación del desarrollo del sistema y manual de operaciones (manual de usuario).



Los alcances están enmarcados en la obtención de resultados en los objetivos secundarios planteados anteriormente, el sistema a desarrollar esta orientado a:

- La implementación del *módulo de ventas* para mejorar las operaciones de venta y control de pasajes.
- La implementación de *módulos de manejo de encomiendas* para mejorar y facilitar el manejo de la información concerniente a las encomiendas.
- La implementación del *módulo de asignación de destinos* para facilitar la elaboración y comunicación de destinos de viajes.
- La implementación del *módulo de llegadas y salidas* para llevar un mejor control de los horarios de los buses para poder ofrecer un mejor servicio a los clientes.
- La implementación del *módulo de reportes y listados* para centralizar la información necesaria para la elaboración de reportes y listados de manera automatizada.

## 1.7 METODOLOGIA

El desarrollo del presente trabajo utiliza la metodología del Análisis y Diseño Estructurado Moderno de Sistemas de Información de Edward Yourdon para la aplicación de ventas; para la aplicación de administración y control se aplica la Metodología de Administración de Relaciones (RMM - Relationship Management Methodology) ideado por Isakowitz, Stohr y Balasubramanian y el modelo DRM (Dynamic Relationship Management Methodology), que mantiene las características del modelo RMM original y agrega otras funcionalidades que permiten modelar un dominio mayor de aplicaciones hipermediales, incluyendo aplicaciones dinámicas, que permiten obtener pantallas con resultados de consultas realizadas por el usuario a la base de datos.

## 1.8 APORTES

El aporte institucional que se pretende alcanzar es minimizar el tiempo sobre las actividades desarrolladas dentro la institución y sobre los factores de conservación de los clientes y de esta manera hacer más eficiente el servicio.

De manera más específica, una vez implantado el sistema, tanto en la oficina central como en las sucursales, se pretende:

- Mejorar la comunicación entre sucursales ya que la transmisión de la información se realiza en un entorno de red (Internet).
- Mejorar el servicio al cliente, al evitar la venta de pasajes duplicados y facilitar la venta de pasajes y reservas.
- Conseguir información confiable y segura, ya que el sistema asegura la veracidad de la misma al implementar mecanismos de control sobre información de precios, ventas, ingresos, destinos, datos de encomiendas y acceso de usuarios.

## Capítulo II: Marco Referencial

### 2.1 MARCO TEORICO

#### 2.1.1 ANALISIS ESTRUCTURADO MODERNO

##### 2.1.1.1 ANALISIS DE SISTEMAS

El Análisis de Sistemas trata básicamente de determinar los objetivos y límites del sistema objeto de análisis, caracterizar su estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos y evaluar sus consecuencias.

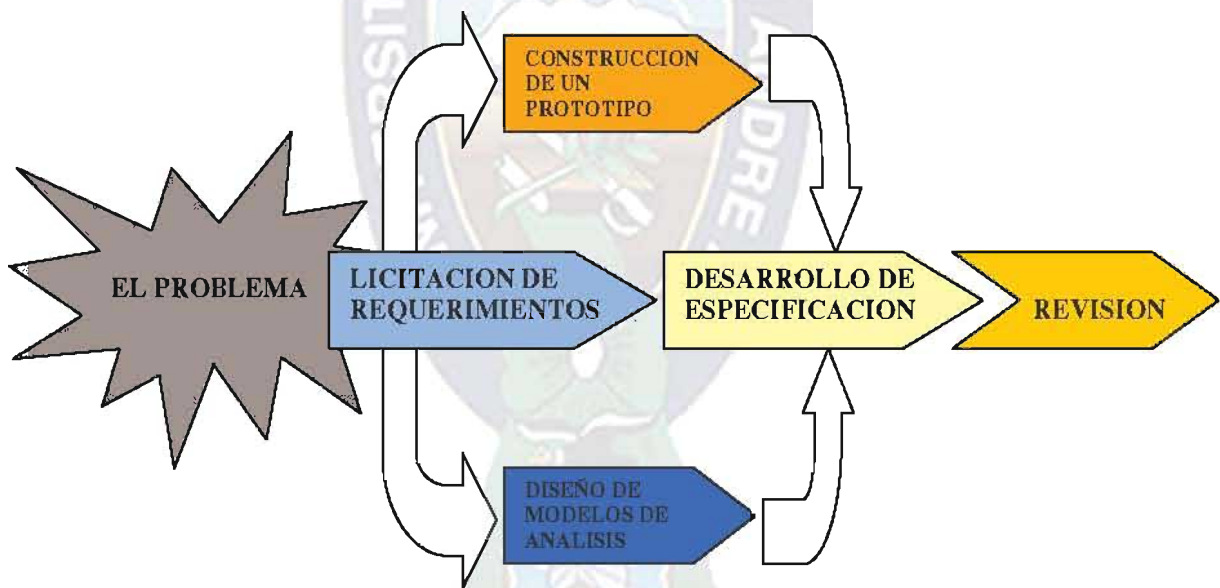


Figura 2.1: El proceso de Análisis  
Fuente: Elaboración Propia

### 2.1.1.2 MODELO ESTRUCTURADO

En el modelo estructurado se examinan brevemente las nueve actividades y los tres terminadores que lo componen, como se muestra en la figura 2.2. Los terminadores son los usuarios, los administradores, y el personal de operaciones [YOURDON93].

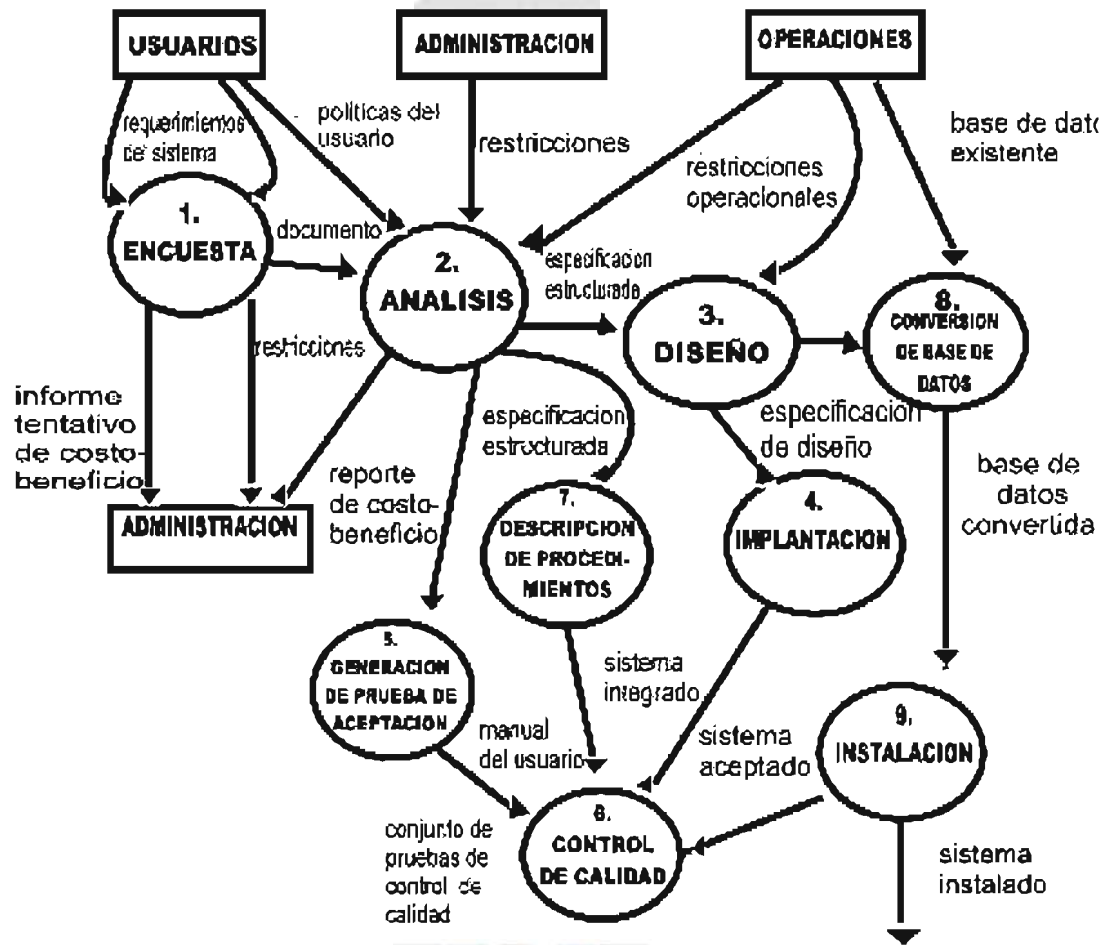


Figura 2.2: Modelo Estructurado  
Fuente: [YOURDON93]

### 2.1.1.3 MODELO ESENCIAL

El modelo esencial del sistema es un modelo de lo que el sistema debe hacer para satisfacer los requerimientos del usuario, diciendo lo mínimo posible acerca de como se implanta [YOURDON93].

El modelo esencial contiene dos componentes principales:

- Modelo ambiental
- Modelo de comportamiento

#### **2.1.1.3.1 MODELO AMBIENTAL**

La labor más difícil en la especificación de un sistema es a menudo determinar qué es parte del sistema y qué no [YOURDON93].

Así, el primer modelo importante que se debe desarrollar es uno que no haga más que definir las interfaces entre el sistema y el resto del universo, es decir, el ambiente. Por razones obvias, este modelo se conoce como el modelo ambiental [YOURDON93]. Por lo tanto, se necesita saber qué información entra al sistema desde el ambiente exterior, y qué información produce como salida al ambiente externo.

Otro aspecto crítico del modelo ambiental consiste en identificar los acontecimientos que ocurren en el ambiente al cual debe responder el sistema. No para todos los acontecimientos; después de todo, el ambiente en su totalidad genera un número infinito de acontecimientos. Sólo nos preocupan aquellos que (1) ocurren en el ambiente exterior y (2) requieren una respuesta del sistema [YOURDON93].

#### **2.1.1.3.2 HERRAMIENTAS USADAS PARA DEFINIR EL AMBIENTE**

El modelo ambiental consta de tres componentes:

- Declaración de propósitos.
- Diagrama de contexto.
- Lista de acontecimientos.

## DECLARACION DE PROPOSITOS

La declaración de propósitos consiste en la declaración textual breve y concisa del propósito del sistema, dirigida al nivel administrativo superior, la administración de los usuarios, y otros que no están directamente involucrados con el desarrollo del sistema [YOURDON93].

## DIAGRAMA DE CONTEXTO

El diagrama de contexto es un caso especial de diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema. La figura 2.3 muestra un diagrama de contexto de un sistema de pedidos de libros [YOURDON93].

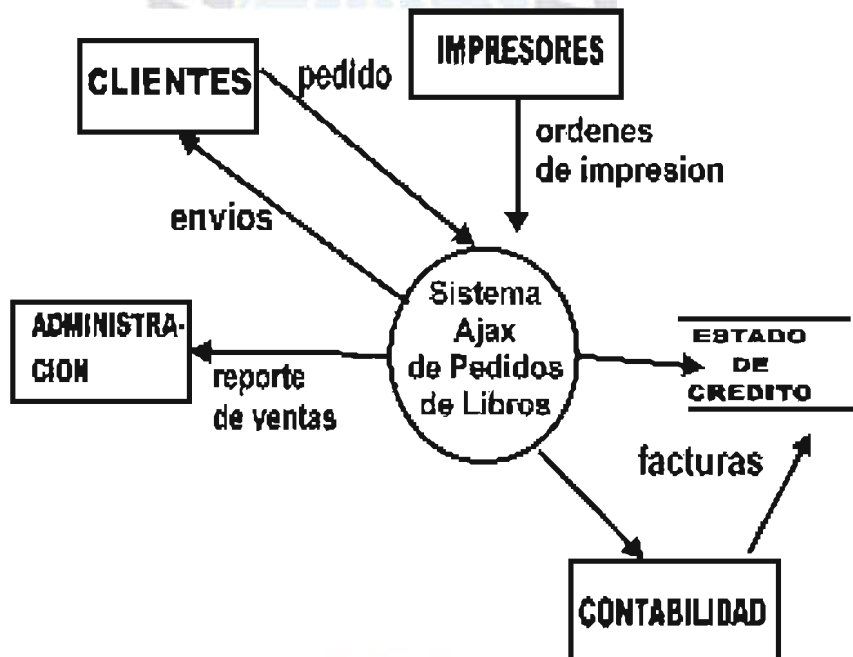


Figura 2.3: Diagrama de contexto  
Fuente: [YOURDON93]

El diagrama de contexto enfatiza varias características importantes del sistema:

- Las personas, organizaciones y sistemas con los que se comunica el sistema, se conocen como terminadores.

- Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
- Los datos que el sistema produce y que se envían al mundo exterior.
- Los almacenes de datos que el sistema comparte con los terminadores. Estos almacenes de datos se crean fuera del sistema para su uso, o bien son creados en él y usados fuera.
- La frontera entre el sistema y el resto del mundo.

### **LISTA DE ACONTECIMIENTOS**

La lista de acontecimientos es una lista narrativa de los estímulos que ocurren en el mundo exterior a los cuales el sistema debe responder [YOURDON93].

Cada acontecimiento se etiqueta como F, T, C. Con ello se muestra si es de tipo de flujo, temporal, o de control. El orientado a flujos es el que se asocia con un flujo de datos; es decir, el sistema se da cuenta de que ha ocurrido el acontecimiento cuando llega algún dato (o posiblemente varios). Los acontecimientos temporales arrancan con la llegada de un momento dado en el tiempo [YOURDON93].

Los acontecimientos de control deben considerarse un caso especial del acontecimiento temporal: un estímulo externo que ocurre en algún momento impredecible. A diferencia de un acontecimiento temporal normal, el acontecimiento de control no se asocia con el paso regular del tiempo, por lo que el sistema no puede anticiparlo utilizando un reloj interno [YOURDON93].

#### **2.1.1.3.3 MODELO DE COMPORTAMIENTO**

Dentro del modelo de comportamiento se involucrará el desarrollo de un diagrama de flujo de datos y un diagrama de entidad-relación preliminares, además de la elaboración de las entradas iniciales del diccionario.

## DIAGRAMAS DE FLUJO DE DATOS

El diagrama de flujo de datos (DFD), es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos" y "tanques de almacenamiento" de datos.

Los componentes de un diagrama típico de flujo de datos:

- Proceso.
- Flujo.
- Almacén.
- Terminador.

### PROCESO

El primer componente del DFD se conoce como proceso. Los sinónimos comunes son burbuja, función, transformación. El proceso muestra una parte del sistema que transforma entradas en salidas. El proceso se representa gráficamente como un círculo, como se muestra en la figura 2.4 [YOURDON93].



Figuras 2.4: Ejemplos de un proceso  
Fuente: [YOURDON93]

El proceso se nombra o describe con una sola palabra, frase u oración sencilla. Un buen nombre para un proceso generalmente consiste en una frase verbo-objeto tal como validar entradas o calcular impuesto [YOURDON93].



## FLUJO

Un flujo se representa gráficamente por medio de una flecha que entra o sale de un proceso. El flujo se usa para describir el movimiento de bloques o paquetes de información de una parte del sistema a otra [YOURDON93].



Figura 2.5: Ejemplo de un flujo  
Fuente: [YOURDON93]

El flujo de la figura 2.5 tiene nombre. El nombre representa el significado del paquete que se mueve a lo largo del flujo. Un corolario de esto es que el flujo sólo lleva un tipo de paquete, como lo indica su nombre.

Los flujos muestran también la dirección: una cabeza de flecha en cualquier extremo (o posiblemente ambos) del flujo indica si los datos (o el material) se está moviendo hacia adentro o hacia fuera de un proceso (o ambas cosas).



Figuras 2.6(a): Flujo de entrada  
Fuente: [YOURDON93]

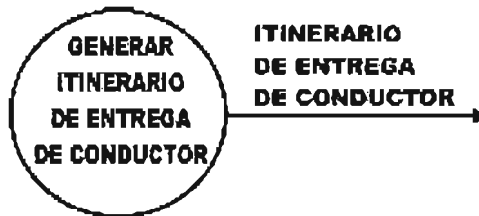


Figura 2.6 (b): Flujo de salida  
Fuente: [YOURDON93]

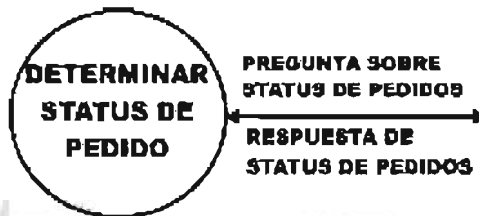


Figura 2.6(c): Flujo de diálogo  
Fuente: [YOURDON93]

### ALMACEN

El almacén se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por dos líneas paralelas, como lo muestra la figura 2.7. De modo característico el nombre que se utiliza para identificar al almacén es el plural del que se utiliza para los paquetes que entran y salen del almacén por medio de flujos [YOURDON93].

---

## PEDIDOS

---

Figura 2.7: Representación gráfica de un almacén  
Fuente: [YOURDON93]

### TERMINADOR

El terminador gráficamente se representa como un rectángulo, como se muestra en la figura 2.8. Los terminadores representan entidades externas con las cuales el sistema se comunica [YOURDON93].



**DEPARTAMENTO  
DE  
CONTABILIDAD**

**Figura 2.8:** Representación gráfica de un terminador  
Fuente: [YOURDON93]

### ***GUIA PARA LA CONSTRUCCION DE UN DFD***

Además de la regla básica que existen para la elaboración de DFD tal como, los componentes básicos de DFD son: proceso (burbuja), flujo, almacenes y terminadores. Existen otras reglas adicionales que nos permitirán no elaborar DFD erróneos y gratos a la vista de los usuarios.

Las reglas incluyen las siguientes:

- Escoger nombres con significado para los procesos, flujos, almacenes y terminadores.
- Numerar los procesos.
- Evitar los DFD excesivamente complejos
- Redibujar el DFD tantas veces como sea necesario estéticamente.
- Asegurarse de que el DFD sea lógicamente consistente y que también sea con cualesquiera DFD relacionados con él.

### ***DICCIONARIO DE DATOS***

Herramienta necesaria. El diccionario de datos es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios [YOURDON93].

## NOTACION DEL DICCIONARIO DE DATOS

=	"esta compuesto de". Para referirnos a un dato compuesto
+	"y"
()	Optativo (puede estar presente o ausente)
{}	Para una iteración
[]	Seleccionar una de varias alternativas
**	Comentarios
@	Indicador (campo clave) para un almacén
	separa operaciones alternativas

## ESPECIFICACION DE PROCESOS

Es una descripción de que es lo que sucede en cada burbuja primitiva o de nivel más bajo en un diagrama de flujo de datos [YOURDON93].

La especificación de procesos describe que es lo que debe hacerse para transformar las entradas en salidas.

## HERRAMIENTAS PARA LA ESPECIFICACION DE PROCESOS

- Lenguaje estructurado
- Pre / Post condición
- Tablas de Decisión
- Diagramas de flujo
- Lenguaje natural

Se puede usar cualquier método mientras que cumpla dos condiciones cruciales:

- La especificación de procesos debe describirse de modo que puede ser verificada tanto por el usuario como por el analista.
- La especificación de proceso debe ser descrita de manera que pueda ser comunicada efectivamente a un público amplio.

### *LENGUAJE ESTRUCTURADO*

Es un subconjunto de un idioma con importantes restricciones sobre el tipo de frases que pueden utilizarse y la manera en que puedan combinarse dichas frases [YOURDON93].

#### *Construcción SI-ENTONCES-OTRO*

Se usa para describir frases alternativas que se ejecutan según el resultado de una decisión binaria [YOURDON93].

Esta construcción puede tomar cualquiera de las formas siguientes:

SI condicion\_1

    Frase\_1

FIN SI

SI condicion\_1

    Frase\_1

OTRO

    Frase\_2

FIN SI

### *Construcción CASO*

Se utiliza para describir frases alternativas que se ejecutaran basándose en los resultados de una decisión multivaluada [YOURDON93].

El formato de la construcción es la siguiente:

HACER CASO

CASO variable=valor\_1

Frase\_1

...

CASO variable=valor\_n

Frase\_n

...

OTRO

Frase\_n+1

FIN CASO

### *Construcción HACER-MIENTRAS*

Se utiliza para describir una frase que se llevara a cabo repetitivamente hasta que la condición se haga verdadera [YOURDON93].

Su formato es:

HACER-MIENTRAS condicion\_1

Frase\_1

FIN HACER

### *Construcción REPITE-HASTA*

Se utiliza para describir la ejecución de una frase especificada por lo menos una vez.

Su formato es el siguiente:

REPITE

Frase\_1

HASTA condicion\_1

#### **2.1.1.4 DIAGRAMA ENTIDAD-RELACION (E-R)**

Los diagramas E-R son un lenguaje gráfico para describir conceptos. Informalmente, son simples dibujos o gráficos que describen la información que trata un sistema de información y el software que lo automatiza [WIKI07].

Los elementos de dicho lenguaje se describen a continuación, por orden de importancia:

#### **ENTIDADES**

Una entidad es cualquier "objeto" discreto sobre el que se tiene información. Se representa mediante un rectángulo o "caja" etiquetada en su interior mediante un nombre. Ejemplos de entidades habituales en los sistemas de información son: factura, persona, albarán, empleado, etc. [WIKI07].

Cada ejemplar de una entidad se denomina instancia.



**CLIENTE**

**Figura 2.9:** Representación grafica de una Entidad  
Fuente: [YOURDON93]

## **RELACIONES**

Una relación describe cierta interdependencia (de cualquier tipo) entre una o más entidades. Se representa mediante un rombo etiquetado en su interior mediante un verbo. Además, dicho rombo debe unirse mediante líneas con las entidades que relaciona (es decir, los rectángulos) [WIKI07].

Una relación no tiene sentido sin las entidades que relaciona.



**Figura 2.10:** Una relación  
Fuente: [YOURDON93]

Las relaciones entre dos entidades se denominan binarias, las relaciones entre tres entidades se denominan ternarias, y las relaciones entre cuatro o más entidades se denominan múltiples. Las relaciones múltiples son poco frecuentes, mientras que las relaciones binarias son habituales en cualquier problema. También son posibles las relaciones reflexivas donde una entidad se relaciona consigo misma. Esto significa que una instancia de una entidad se relaciona con otra instancia distinta de la misma entidad [WIKI07].



## **ATRIBUTOS**

Los atributos son propiedades relevantes propias de una entidad y sólo una. Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta.

Por motivos de legibilidad, los atributos no suelen representarse en un diagrama entidad-relación, sino que se describen textualmente en otros documentos adjuntos.

Los atributos describen información útil sobre las entidades. En particular, los atributos identificativos son aquellos que permiten diferenciar a una instancia de la entidad de otra distinta [WIKI07].

### **2.1.1.5 DIAGRAMA ENTIDAD-RELACION EXTENDIDO**

Los diagramas Entidad-Relación no cumplen su propósito con eficacia debido a que tienen limitaciones semánticas. Por ese motivo se suelen utilizar los diagramas Entidad-Relación extendidos que incorporan algunos elementos más al lenguaje [WIKI07]:

#### ***ENTIDADES FUERTES Y DEBILES***

Cuando una entidad participa en una relación puede adquirir un papel fuerte o débil. Una entidad débil es aquella que no puede existir sin participar en la relación, es decir, aquella que no puede ser unívocamente identificada solamente por sus atributos. Una entidad fuerte es aquella que "presta" algunos de sus atributos a la entidad débil para que se pueda identificar cualquier instancia. Esto lo hace a través de la relación que las une.

Las entidades débiles se representan mediante un doble rectángulo, es decir, un rectángulo con doble línea [WIKI07].

### **CARDINALIDAD DE LAS RELACIONES**

Las relaciones, en principio binarias, pueden involucrar a un número distinto de instancias de cada entidad. Así, son posibles tres tipos de cardinalidades [WIKI07]:

- Relaciones de uno a uno: una instancia de la entidad A se relaciona con una y solamente una de la entidad B.
- Relaciones de uno a muchos: cada instancia de la entidad A se relaciona con varias instancias de la entidad B.
- Relaciones de muchos a muchos: cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B.

El tipo de cardinalidad se representa mediante una etiqueta en el exterior de la relación, respectivamente: "1:1", "1:N" y "N:M". Otra forma de expresar la cardinalidad es situando un símbolo cerca de la línea que conecta una entidad con una relación [WIKI07]:

- "0" si la entidad no está obligada a participar en la relación.
- "1" si la entidad está obligada a participar en la relación y, además, cada instancia solamente participa una vez.
- "N", "M", ó "\*" si la entidad no está obligada a participar en la relación y cada instancia puede participar cualquier número de veces.

## **ATRIBUTOS EN RELACIONES**

Las relaciones también pueden tener atributos asociados. Se representan igual que los atributos de las entidades. Un ejemplo típico son las relaciones de tipo "histórico" donde debe constar una fecha o una hora [WIKI07].

## **HERENCIA**

La herencia es un intento de adaptación de estos diagramas al paradigma orientado a objetos. La herencia es un tipo de relación entre una entidad "padre" y una entidad "hijo". La entidad "hijo" hereda todos los atributos y relaciones de la entidad "padre". Por tanto, no necesitan ser representadas dos veces en el diagrama. La relación de herencia se representa mediante un triángulo interconectado por líneas a las entidades. La entidad conectada por el vértice superior del triángulo es la entidad "padre". Solamente puede existir una entidad "padre". Las entidades "hijo" se conectan por la base del triángulo [WIKI07].

### **2.1.1.5.1 MODELADO ENTIDAD-RELACION**

El modelado entidad-relación es una técnica para el modelado de datos utilizando diagramas entidad relación. No es la única técnica pero sí la más utilizada. Brevemente consiste en los siguientes pasos [WIKI07]:

- Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
- Se hace una lista de los sustantivos y verbos que aparecen.
- Los sustantivos son posibles entidades o atributos.
- Los verbos son posibles relaciones.
- Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.

- Se elabora el diagrama (o diagramas) entidad-relación.
- Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

## 2.1.2 METODOLOGIA RMM

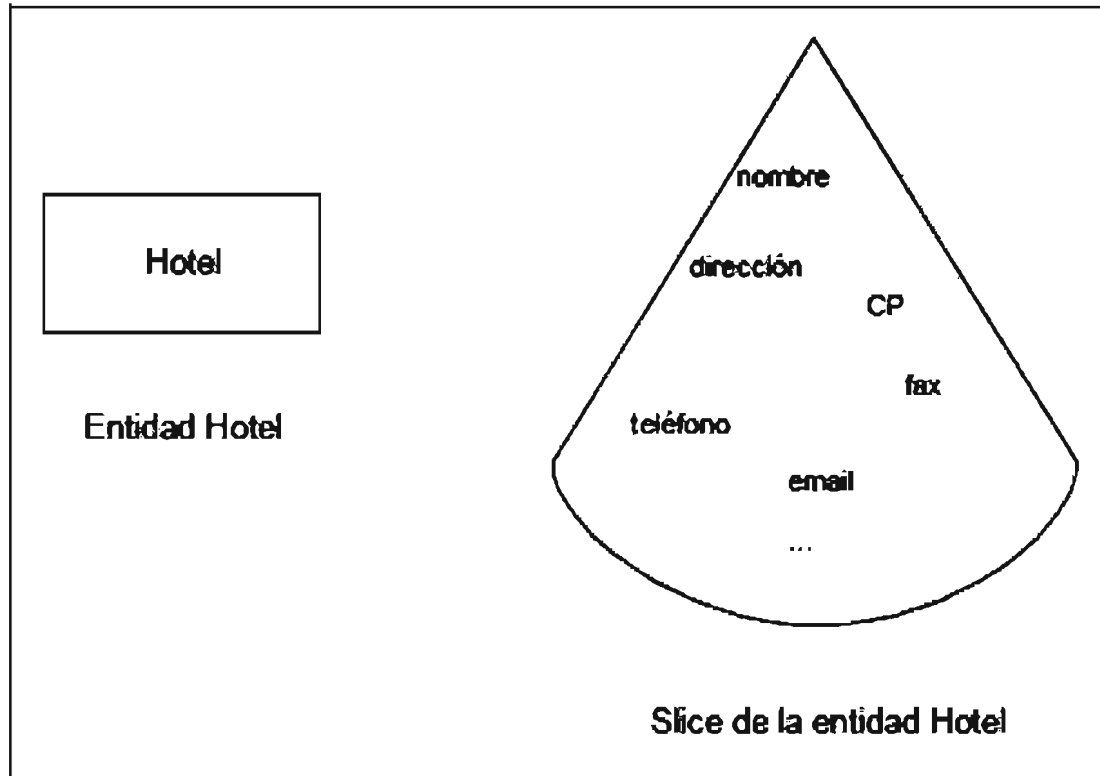
La metodología RMM (Relationship Management Methodology) ha sido ideada por Isakowitz, Stohr y Balasubramanian. Esta metodología es apropiada para dominios con estructuras regulares (es decir, con clases de objetos bien definidas, y con claras relaciones entre esas clases). Por ejemplo, catálogos, front-ends de bases de datos tradicionales. Según sus autores, está orientada a problemas con datos volátiles, que cambian con mucha frecuencia, más que a entornos estáticos [NAVARRETE96].

### 2.1.2.1 EL MODELO RMDM [NAVARRETE96]

La base de la metodología es el modelo de datos RMDM (Relationship Management Data Model), que se genera a partir de un diagrama entidad-relación. Con él se describirá no sólo la información referente a las clases de objetos, sino también a la navegación entre ellos.

Hay definidas primitivas para modelar los dominios (clases de objetos) y otras para el acceso a tales objetos. De entre las primeras, la más típica es la entidad. Como en la teoría relacional una entidad está compuesta por varios atributos. Además, en RMDM se incorpora una nueva primitiva muy importante denominada slice, que define conjuntos de atributos de una entidad que se agrupan de forma lógica.

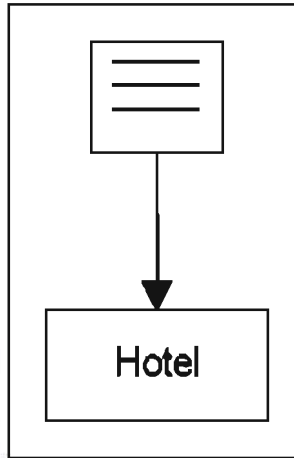
Una entidad se representa mediante un recuadro y su nombre, mientras que un slice se representa mediante a una figura similar a una gota de agua. Lo vemos en la figura 2.11.



**Figura 2.11:** Representación de una Entidad y un Slice  
**Fuente:** [NAVARRETE96]

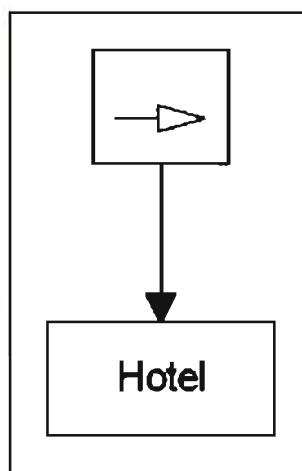
En cuanto a las primitivas de acceso, se definen tres tipos de acceso diferentes. Para ver qué significa cada una veremos un ejemplo. Supongamos una entidad "Hotel". Las instancias de esta entidad se pueden visitar según esos tres tipos de acceso:

Índice condicional (Conditional Index): en una pantalla aparecerá un índice alfabético de los nombres de los hoteles, y pulsando sobre uno, iremos a ver su información. Para ver los demás deberemos volver al índice. En realidad el índice no es necesario que sea siempre una lista de las instancias como tal. Por ejemplo un índice para acceder a una entidad "ciudad" pudiera ser, por ejemplo, un mapa, donde al pulsar sobre una ciudad, se visitase su información.



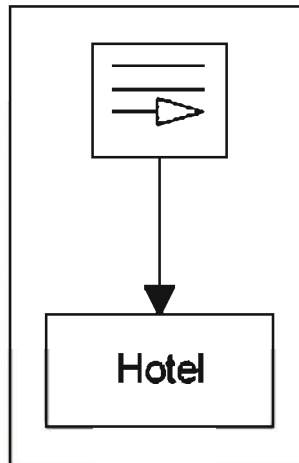
**Figura 2.12:** Representación de un Índice Condicional  
**Fuente:** [NAVARRETE96]

Visita guiada condicional (Conditional Guided Tour): de la pantalla inicial se nos envía al primer hotel, y de él se puede pasar al segundo, y así sucesivamente, de forma que para llegar al último hay que pasar por todos los anteriores. Se permite volver a visitar al anterior hotel. En definitiva, es una estructura de acceso secuencial.



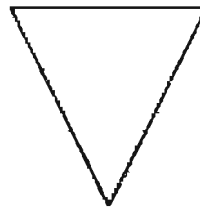
**Figura 2.13:** Representación grafica de una Vista Guiada Condicional  
**Fuente:** [NAVARRETE96]

Visita guiada indexada condicional (Conditional Indexed Guided Tour): es una solución híbrida, donde tenemos un índice para acceder puntualmente a los elementos de la entidad, pero también se nos permite la navegación secuencial una vez seleccionado uno.



**Figura 2.14:** Representación gráfica de una Vista Guiada Indexada Condicional  
Fuente: [NAVARRETE96]

Además, también se define una primitiva de grupo (Grouping) mediante la cual se representa un menú. De una primitiva de grupo colgarán tantas opciones como queremos que haya en el menú.



**Figura 2.15:** Representación gráfica de una Primitiva de Grupo  
Fuente: [NAVARRETE96]

La primitiva de grupo, o a la que a veces también denominaremos de menú, se representa mediante un triángulo invertido del que colgarán todas las opciones del menú. Por ejemplo, veamos un menú que permite acceder a la entidad hotel de las tres maneras diferentes que antes hemos estudiado:

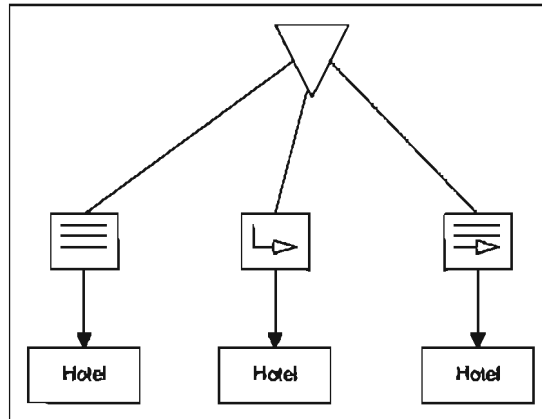


Figura 2.16: Representación gráfica de una Menú  
Fuente: [NAVARRETE96]

Primitivas de datos	Entidad	
	Atributo	atributo
	Slice	
Primitivas de acceso	Índice	
	Visita guiada	
	Visita guiada indexada	
	Grupo	
	Enlace	

Tabla 2.1: Primitivas del RMM  
Fuente: [NAVARRETE96]



## 2.1.2.2 ETAPAS DE LA METODOLOGIA

### **ETAPA 0**

Como toda metodología debe comenzar con un estudio de factibilidad y un análisis de los requerimientos (tanto de la información como de la navegación). También debe hacerse una selección del hardware y software que se necesitará.

### **ETAPA 1: DISEÑO ENTIDAD-RELACION**

En esta etapa se confecciona un diagrama entidad-relación típico, desglosando las relaciones N:M en dos relaciones 1:N, como muestra el ejemplo:

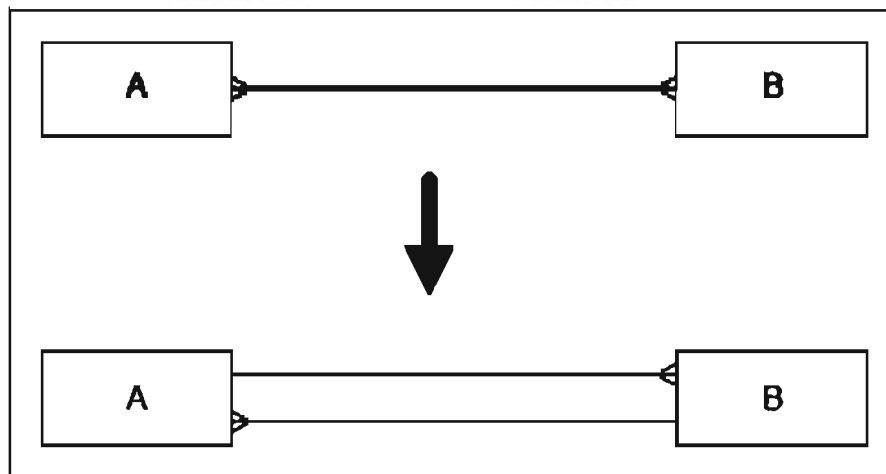


Figura 2.17: Etapa 1 del RMM  
Fuente: [NAVARRETE96]

Como vemos las relaciones se representan con el símbolo:

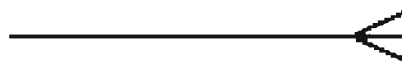


Figura 2.18: Representación de una relación en RMM  
Fuente: [NAVARRETE96]

El objetivo de esta fase es explicitar todos los enlaces entre objetos. Más tarde, las relaciones darán lugar a la navegación. Así, una relación especificará un camino en la navegación.

Dentro de esta fase habría también que especificar los atributos de cada entidad.

### ***ETAPA 2: DISEÑO DE SLICES***

Esta etapa consiste en dividir una entidad en fragmentos significativos y organizarlos en la red de navegación. Esta división se hace según la semántica de los atributos.

Cada slice agrupará uno o más atributos de una entidad, de tipos muy diferentes. Cada entidad tendrá su head, o slice principal, que se marca con un asterisco y que es slice al que, por defecto, se accede a través de los mecanismos de navegación.

Entre los diferentes slices están los llamados enlaces estructurales, que nada tienen que ver con las relaciones, ya que al atravesar un enlace estructural, no se produce ningún cambio de contexto.

Una vez determinados los slices hay que establecer los enlaces estructurales.

### ***ETAPA 3: DISEÑO NAVEGACIONAL***

Como que cada relación del diagrama entidad-relación da lugar a un enlace de navegación, en esta fase sustituimos las relaciones por primitivas de acceso RMDM.

En general, preferiremos una visita guiada a un índice cuando el número de instancias sea pequeño (menor de 10) y no exista un campo índice que pueda ayudar a los usuarios.

Por contra, si el número es grande, usaremos índices. Las visitas guiadas indexadas son un híbrido, usado frecuentemente cuando hay un índice y se desea una navegación entre las instancias.

Además en esta fase hay que elegir a qué slice se accede a través de una primitiva de acceso. Por defecto es el slice principal (head). En caso contrario, debe especificarse, etiquetando el nombre de la estructura de acceso.

Por último, en esta fase se establece una jerarquía de menús, utilizando la primitiva de grupo o menú. Como regla general, evitar grandes profundidades en la jerarquía, ya que desorientan al usuario.

El resultado final de esta etapa es el diagrama RMDM.

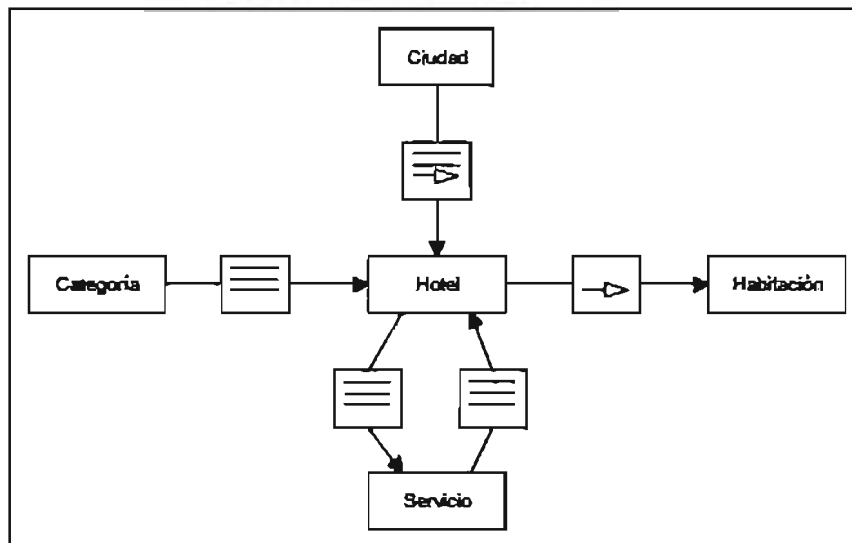


Figura 2.19: Diagrama RMDM  
Fuente: [NAVARRETE96]

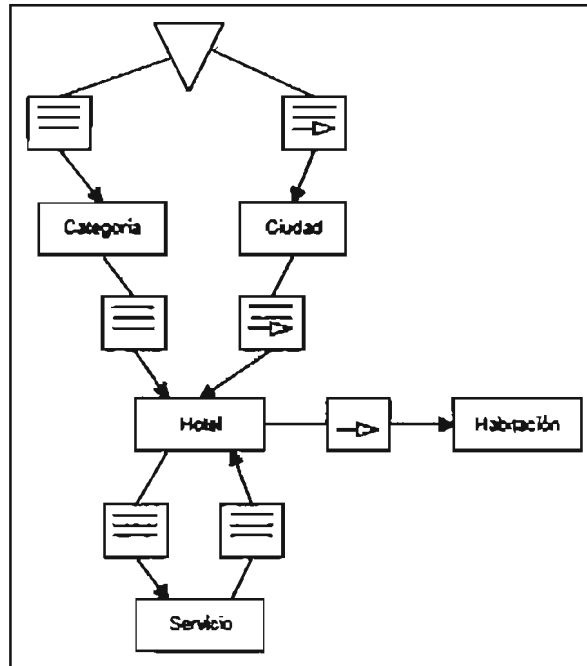


Figura 2.20: Diagrama RMDM con menú  
Fuente: [NAVARRETE96]

#### ***ETAPA 4: DISEÑO DE PROTOCOLOS DE CONVERSION***

En esta etapa se escriben unos protocolos por los cuales se transforma cada elemento del RMDM en un objeto en la plataforma elegida.

#### ***ETAPA 5: DISEÑO DE LA INTERFAZ DE USUARIO***

Diseño gráfico de todas las pantallas correspondientes a cada uno de los slices que hemos obtenido en la etapa 2.

#### ***ETAPA 6: DISEÑO DE COMPORTAMIENTO EN TIEMPO DE EJECUCION***

Se decide qué mecanismos se utilizarán para guardar la historia, hacer backtrackings, permitir enlaces transversales, etc.

## **ETAPA 7: CONSTRUCCION Y PRUEBAS**

Construcción de la aplicación y pruebas, testeando cuidadosamente todos los paths de la navegación.

### **2.1.2.3 DEFICIENCIAS DEL MODELAMIENTO CONCEPTUAL DE RMM Y PROPUESTA DE SOLUCIÓN: MODELO DRM [ROJAS2000]**

El Modelo DRM (Dynamic Relationship Management Model), mantiene las características del modelo RMM original y agrega otras funcionalidades que permiten modelar un dominio mayor de aplicaciones hipermediales. Está orientado, principalmente, a sitios Web que presentan un alto grado de interacción con el usuario, con conexiones a bases de datos.

Las siguientes son las deficiencias encontradas en el modelo original, con sus correspondientes mejoras incluidas en el modelo DRM:

- La primera estructura de modelamiento conceptual de RMM, el Esquema Entidad-Relación, no permite incorporar atributos de interrelaciones, debido a que éstas se reemplazan, posteriormente, por links directos (en el caso de cardinalidad 1:1) o por índices (1:n).

La solución a este problema será recurrir a otros mecanismos para suplir la ausencia de tales atributos.

Se puede crear una nueva entidad, que contenga atributos identificadores de las entidades interrelacionadas, además del o los atributos que se deseaban incluir en tal interrelación.

- RMM no considera las cardinalidades 0:1 y 0:N para las interrelaciones. Los problemas cuyo modelamiento requiere este tipo de interrelaciones son muy frecuentes.

Se propone incluir la interrelación con cardinalidad 0:N. En la siguiente figura, se muestra la notación gráfica para este caso:

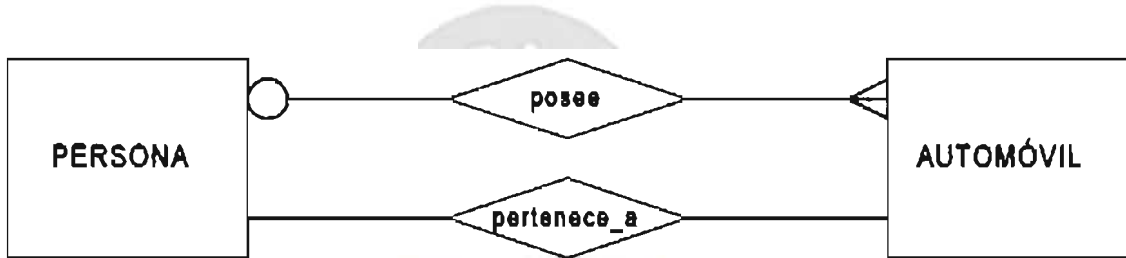


Figura 2.21: Cardinalidad 0:N  
Fuente: [ROJAS2000]

En este ejemplo, la interrelación "posee" tiene cardinalidad 0:N, mientras que "pertenece\_a" es 1:1, lo que representa el hecho que una persona puede tener 0 ó varios automóviles y un automóvil siempre pertenece a una sola persona.

También se agregará la interrelación 0:1, representada gráficamente en la figura siguiente, por la interrelación "seleccionado\_de".

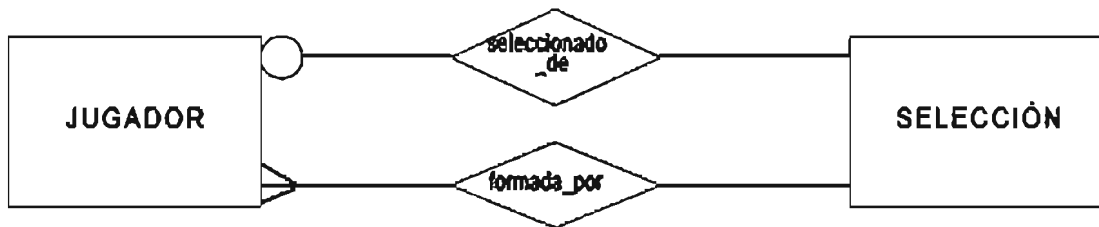


Figura 2.22: Cardinalidad 0:1  
Fuente: [ROJAS2000]

La figura se interpreta de la siguiente forma: un jugador puede ser seleccionado de a lo más una selección, la que está formada por varios jugadores (no se especifica mínimo ni máximo).

- Las consultas a realizar a la base de datos subyacente se limitan a las que se puedan contestar mediante el poblamiento de las interrelaciones definidas en el Esquema Entidad-Relación, no soportando aún otro tipo de consultas que requieran mayor procesamiento. En definitiva, todas las opciones de navegación están previamente definidas e implementadas.

Se sugiere introducir la estructura "query" para manejar tal problema donde "search" sea un botón que active un proceso de consulta a la base de datos, generando una m-slice que modela la unidad de presentación al usuario, con los resultados de la consulta.

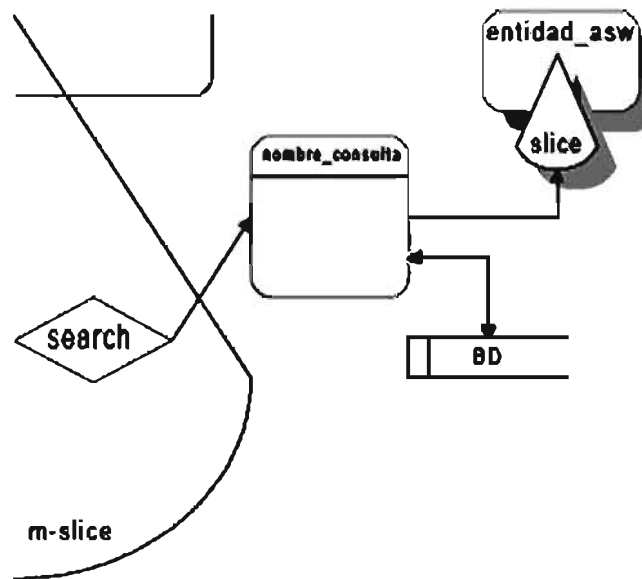


Figura 2.23: Representación grafica de un M-Slice  
Fuente: [ROJAS2000]

El botón "search" representa el formulario de entrada de los parámetros de la consulta, que corresponden a atributos de alguna(s) entidad(es), además de algún otro tipo de parámetro compatible con el (los)

atributo(s) consultados. La compatibilidad de los dominios de los argumentos debe ser chequeada por el proceso de consulta.

Si bien es cierto la m-slice resultante siempre puede atribuírsele a alguna entidad que ya existe, su generación "on-the-fly" y su naturaleza volátil (nace y muere con la consulta) la hacen una entidad distinta, por lo que se le asigna otro nombre (ejemplificado en la figura por "entidad\_asw").

- En el caso de la estructura m-slice, existen deficiencias en la notación gráfica y de lenguaje:
  - No está definida la notación para la acción Download (bajar un archivo)

Solución:



**Figura 2.24:** Representación gráfica de la acción Download  
Fuente: [ROJAS2000]

- No está definida notación de lenguaje de Link Externo (link a una aplicación externa al software que se está modelando).

Solución:

**\*<m-slice ancla>**

**Figura 2.25:** Representación de un Enlace Externo  
Fuente: [ROJAS2000]

- M-slice considera elementos auxiliares, como texto fijo, que no aportan información sobre las entidades. Su notación es compleja



y se transforma en un distractor en la etapa de Diseño Conceptual del software. Es recomendable postergar la inclusión de tales elementos a una etapa posterior de Diseño Sintáctico de Interfaz de Usuario.

- En la gran mayoría de los sistemas hipermediales, existe una característica de navegación muy particular, que se refiere a la acción de volver a la página o pantalla anterior. En el diseño de la aplicación, se debe decidir a cuál página es a la que se retorna: a la última que se revisó durante la navegación o a la anterior definida según una secuencia preestablecida. Las m-slices cubren, implícitamente, sólo este último caso.

Se propone la siguiente estructura, last-visited, que lleva a la última página visitada, comportándose de la misma forma que los botones "Back" de los navegadores de Internet.

La notación de last\_visited es la siguiente:



**Figura 2.26:** Representación gráfica de una Última Visita  
Fuente: [ROJAS2000]

Al desplegar atributos de tipo audio y video, tenemos múltiples alternativas para hacerlo: desplegarlos directamente al ingresar a la página, desplegarlos según instrucción del usuario, utilizar controles como stop, play, pause, etc. Este problema no es cubierto por las m-slices.

Se sugiere, entonces, permitir la definición de nuevos medios, como nuevos tipos de datos. Así, no se contará sólo con el tipo de dato "video", sino, por ejemplo, "video directo", "video con controles" (especificando qué controles), etc. Para esto, se deberá definir una notación de lenguaje que permita estas definiciones.



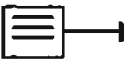
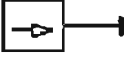
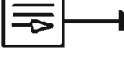



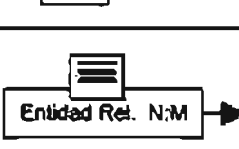
	Hiperenlace
	Hiperenlace jerárquico inferido
	Índice
	Visita guiada
	Visita guiada indexada
	Índice múltiple
	Visita guiada múltiple
	Visita guiada indexada múltiple
	Ejemplo de acceso a partir de una relación N:M, en este caso utilizando un índice (podría ser cualquier otra primitiva de acceso simple)

Tabla 2.2: Primitivas de acceso con DRM  
Fuente: [ROJAS2000]

### 2.1.3 CALIDAD DEL SOFTWARE

Todas las metodologías y herramientas tienen un único fin *"producir software de gran calidad"* [CUEVAS99].

Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad

Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad

Existen algunos requisitos implícitos o expectativas que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo el deseo de un buen mantenimiento) que también pueden implicar una falta de calidad.

El objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso a la hora de la entrega y del uso por parte de los usuarios.

*"Es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible".*

#### 2.1.3.1 DEFINICIONES DE CALIDAD DE SOFTWARE

"Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente" R. S. Pressman (1992).

“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas” ISO 8402 (UNE 66-001-92).

### 2.1.3.2 DIFERENTES ASPECTOS DE LA CALIDAD

- **Interna:** medible a partir de las características intrínsecas, como el código fuente
- **Externa:** medible en el comportamiento del producto, como en una prueba
- **En uso:** durante la utilización efectiva por parte del usuario

### 2.1.3.3 ISO/IEC 9126: TECNOLOGIAS DE LA INFORMACION – CALIDAD DE LOS PRODUCTOS SOFTWARE [ISO9126]

- **Parte 1:** Modelo de Calidad
- **Parte 2:** Métricas Externas
- **Parte 3:** Métricas Internas
- **Parte 4:** Métricas de Calidad en Uso

Un modelo de calidad tiene en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquirentes y usuarios finales (ISO, 2001).

Pueden ser utilizados para construir mejores productos y asegurar su calidad.

Se han desarrollado varios modelos de calidad para diferentes productos y procesos software.

En general, descomponen la calidad jerárquicamente en una serie de características y subcaracterísticas que pueden usarse como una lista de

comprobación de aspectos relacionados con la calidad y usan como base la ISO 9126

### 2.1.3.4 MODELO DE CALIDAD: PARA CALIDAD INTERNA Y EXTERNA [ISO/IEC9126]

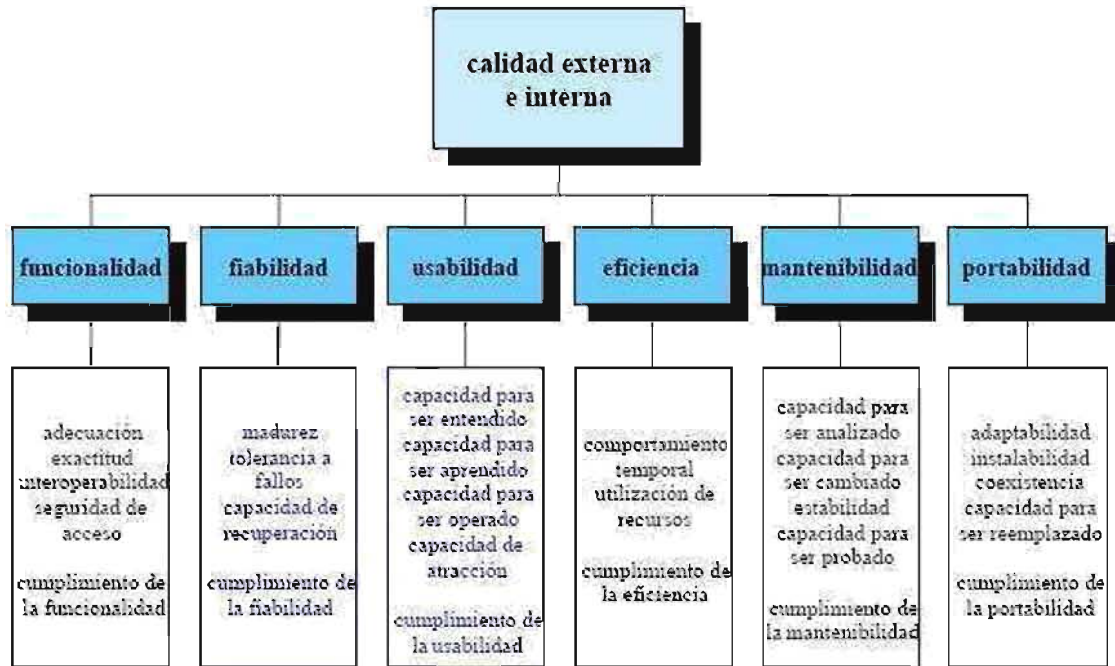


Figura 2.27: Calidad externa e interna  
Fuente: [KOBAYASHI]

## FUNCIONALIDAD

Capacidad del producto software para proporcionar funciones que satisfagan las necesidades especificadas e implícitas.

La funcionalidad se subdivide en cinco subcaracterísticas:

- **Adecuación:** la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.

- **Exactitud:** la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.
- **Interoperabilidad:** la capacidad del producto software para interactuar con uno o más sistemas especificados.
- **Seguridad:** referido a la capacidad del producto software para proteger la información y los datos.
- **Conformidad:** la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad.

## FIABILIDAD

Capacidad del producto software para mantener un nivel especificado de rendimiento.

La fiabilidad se subdivide en cuatro subcaracterísticas:

- **Madurez:** la capacidad del producto software para evitar fallos provocados por errores en el software.
- **Tolerancia a fallos:** la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.
- **Recuperabilidad:** la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.
- **Conformidad:** la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.

## USABILIDAD

La capacidad del producto software de ser entendido, aprendido, utilizado y atractivo al usuario.

La usabilidad se subdivide en cinco subcaracterísticas:

- **Comprensibilidad:** la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.
- **Facilidad de aprendizaje:** la capacidad del producto software para permitir al usuario aprender su aplicación.
- **Operabilidad:** la capacidad del producto software para permitir que el usuario lo opere y lo controle.
- **Atracción:** la capacidad del producto software para atraer al usuario.
- **Conformidad:** la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.

## EFICIENCIA

La capacidad del producto software para proporcionar el rendimiento apropiado, relativo a la cantidad de recursos utilizados.

La eficiencia se subdivide en tres subcaracterísticas:

- **Comportamiento temporal:** la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.

- **Utilización de recursos:** la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.
- **Conformidad:** la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.

## MANTENIBILIDAD

La capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales.

La mantenibilidad se subdivide en cinco subcaracterísticas:

- **Analizabilidad:** Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.
- **Cambiabilidad:** Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.
- **Estabilidad:** Capacidad del producto software de evitar los efectos inesperados de las modificaciones.
- **Facilidad de prueba:** Capacidad del producto software de permitir validar las partes modificadas.
- **Conformidad:** Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.

## PORTABILIDAD

La capacidad del producto software de ser transferido de un entorno a otro.



La portabilidad se subdivide en cinco subcaracterísticas:

- **Adaptabilidad:** la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.
- **Facilidad de instalación:** la capacidad del producto software para ser instalado en un ambiente determinado.
- **Coexistencia:** la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.
- **Reemplazabilidad:** la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.
- **Conformidad:** la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.



<b>Característica</b>	<b>Pregunta Central</b>	<b>Subcaracterística</b>	<b>Pregunta Central</b>
<i>Funcionalidad</i>	Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué ... ?	Adecuación	Tiene el conjunto de funciones apropiadas para las tareas especificadas?
		Exactitud	Hace lo que fue acordado en forma esperada y correcta?
		Interoperabilidad	Interactúa con otros sistemas especificados?
		Conformidad	Está de acuerdo con las leyes o normas y estándares, u otras prescripciones?
		Seguridad de Acceso	Previene accesos no autorizados a los datos y programas?
<i>Confiabilidad</i>	Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?	Nivel de Madurez	Con qué frecuencia presenta fallas por defectos o errores?
		Tolerancia a fallas	Si suceden fallas, como se comporta en cuanto a la performance especificada?
		Recuperabilidad	Es capaz de recuperar datos en caso de fallas?
<i>Usabilidad</i>	El software, es fácil de usar y de aprender?	Comprensibilidad	Es fácil de entender y reconocer la estructura y la lógica y su aplicabilidad?
		Facilidad de Aprender	Es fácil de aprender a usar?
		Operabilidad	Es fácil de operar y controlar?
<i>Eficiencia</i>	Es rápido y minimalista en cuanto a uso de recursos, bajo ciertas condiciones?	Comportamiento con respecto al Tiempo	Cuál es el tiempo de respuesta y performance en la ejecución de la función?
		Comportamiento con respecto a Recursos	Cuántos recursos usa y durante cuánto tiempo?
<i>Mantenibilidad</i>	Es fácil de modificar y testear ?	Analisisabilidad	Es fácil diagnosticar una falla o identificar partes a modificar?
		Modificabilidad	Es fácil de modificar y adaptar?
		Estabilidad	Hay riesgos o efectos inesperados cuando se realizan cambios?
		Testeabilidad	Son fáciles de validar las modificaciones?
<i>Portabilidad</i>	Es fácil de transferir de un ambiente a otro?	Adaptabilidad	Es fácil de adaptar a otros entornos con lo provisto?
		Instalabilidad	Es fácil de instalar en el ambiente especificado?
		Conformidad	Adhiere a los estándares y convenciones de portabilidad?
		Reemplazabilidad	Es fácil de usarlo en lugar de otro software para ese ambiente?

**Tabla 2.3:** Descripción del ISO 9126 (Parte 2 y 3)  
**Fuente:** [KOBAYASHI]

## 2.2 MARCO CONCEPTUAL

### 2.2.1 TECNOLOGIAS WEB

#### 2.2.1.1 INTERNET

Internet es una red mundial de computadoras interconectadas con un conjunto de protocolos, el más destacado, el TCP/IP. Aparece por primera vez en 1960. También se usa este nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que use las mismas tecnologías que Internet, independientemente de su extensión o de que sea pública o privada.

Cuando se dice red de redes se hace referencia a que es una red formada por la interconexión de otras redes menores.

#### 2.2.1.2 PROTOCOLO DE INTERNET

El Protocolo de Internet (IP, de sus siglas en inglés Internet Protocol) es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

#### ***PROTOCOLO TCP/IP***

El Protocolo de Internet (IP) y el Protocolo de Transmisión (TCP), fueron desarrollados inicialmente en 1973 por el informático estadounidense Vinton Cerf como parte de un proyecto dirigido por el ingeniero norteamericano Robert Kahn y patrocinado por la Agencia de Programas Avanzados de Investigación (ARPA, siglas en inglés) del Departamento Estadounidense de Defensa. Internet comenzó siendo una red informática de ARPA (llamada ARPAnet) que conectaba redes de ordenadores de varias universidades y laboratorios en investigación en Estados Unidos. World Wide Web se desarrolló en 1989 por el

informático británico Timothy Berners-Lee para el Consejo Europeo de Investigación Nuclear (CERN, siglas en francés).

### **PROTOCOLO HTTP**

El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con mensajes y otros similares.

#### **2.2.1.3 NAVEGADOR**

Un navegador web, hojeador o browser es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet. Esta red de documentos es denominada World Wide Web (WWW) o Telaraña Mundial. Los navegadores actuales permiten mostrar o ejecutar: gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

#### **2.2.1.4 SERVIDOR WEB**

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear <http://www.ars-wdd.com/> en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página;

el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

#### **2.2.1.5 SISTEMA CLIENTE-SERVIDOR**

Un sistema Cliente/Servidor suele ejecutarse en al menos dos sistemas distintos (uno hace de cliente y el otro de servidor). No obstante, es posible que tanto el cliente como el servidor se encuentren en un único sistema. Generalmente un servidor proporciona servicios a varios sistemas clientes, aunque puede haber un único sistema. La función del cliente suele llevarla a cabo un servidor de archivos, excepto cuando se necesita el máximo rendimiento y debe utilizarse un servidor especializado. A menudo el cliente es un sistema de escritorio conectado en red. Siempre que el usuario necesite recuperar o almacenar información, la parte cliente de la aplicación ejecuta una solicitud, que se envía (generalmente por una red) al servidor. El servidor ejecuta entonces la solicitud y devuelve la información al cliente.

#### **2.2.2 SISTEMA GESTOR DE BASE DE DATOS**

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la Base de datos y el usuario, las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

## **MySQL**

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos propietarios puede comprar a la empresa una licencia más permisiva que les permita ese uso.

Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias propietarias, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

## **CONEXION ODBC**

ODBC son las siglas de Open DataBase Connectivity, que es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS, el propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el DBMS entienda. Para que esto funcione tanto la aplicación

como el DBMS deben ser compatibles con ODBC, esto es que la aplicación debe ser capaz de producir comandos ODBC y el DBMS debe ser capaz de responder a ellos.

Para conectarse a la Base de Datos se crea una DSN dentro del ODBC que define los parámetros, ruta y características de la conexión según los datos que solicite el fabricante.

### **2.2.3 LENGUAJES**

#### ***HTML***

El HTML, acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender.

HTML es una aplicación de SGML conforme al estándar internacional ISO 8879. XHTML es una reformulación de HTML 4 como aplicación XML 1.0, y que supone la base para la evolución estable de este lenguaje. Además XHTML permite la compatibilidad con los agentes de usuario que ya admitían HTML 4 siguiendo un conjunto de reglas.

#### ***JAVASCRIPT***

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, orientado a las páginas web basado en el paradigma prototipo, con una sintaxis semejante a la del lenguaje Java.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien orientado a eventos. Estará listo para actuar en cuanto un evento (un click en un botón, por ejemplo) sea ejecutado. Aún así javascript implementa una sencilla interfaz de objetos/propiedades/métodos.

El lenguaje Javascript se integra dentro del código HTML de las páginas Web.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores de Internet comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

### ***PHP***

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. Las siglas significan "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+.

### ***SQL***

El Lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla.



## ***VISUAL FoxPro***

Visual FoxPro es un lenguaje de programación orientado a objetos y procedural, un Sistema Gestor de Bases de datos o Database Management System (DBMS), y desde la versión 7.0, un Sistema administrador de bases de datos relacionales, producido por Microsoft.

Visual FoxPro ofrece a los desarrolladores un conjunto de herramientas para crear aplicaciones de bases de datos para el escritorio, entornos cliente/servidor, PC o para la Web.

Entre sus características se pueden enumerar:

- Capacidades poderosas y muy veloces para el manejo de datos nativos y remotos.
- Flexibilidad para crear todo tipo de soluciones de bases de datos.
- Lenguaje de programación Orientado a objetos.
- Utilización de sentencias SQL en forma nativa.
- Poderoso manejo de vistas y cursores y control completo de estructuras relacionales.
- Su propio gestor de base de datos incorporado. Sin embargo, también puede conectarse con servidores de base de datos, tales como Oracle, Microsoft SQL Server o MySQL.
- Cuenta con un motor de generación de informes renovado y muy flexible para soluciones más robustas.

### **2.2.4 CONTROL DE ACCESO**

El control de acceso constituye una poderosa herramienta para proteger la entrada a un web completo o sólo a ciertos directorios concretos e incluso a ficheros o programas individuales. Este control consta generalmente de dos pasos:

En primer lugar, la autenticación, que identifica al usuario o a la máquina que trata de acceder a los recursos, protegidos o no.

En segundo lugar, procede la cesión de derechos, es decir, la autorización, que dota al usuario de privilegios para poder efectuar ciertas operaciones con los datos protegidos, tales como leerlos, modificarlos, crearlos, etc.

## **AUTENTICACION**

Se emplean sistemas de autenticación en la mayoría de las actividades diarias de negocios. En las transacciones relacionadas con información personal o financiera, se debe mostrar una prueba de identidad. Cuando deseamos cambiar un cheque en un banco, el cajero nos pide primero que le mostremos un documento de identidad antes de entregarnos el valor del cheque.

La autenticación puede tomarse en dos contextos diferentes:

**Autenticación de Usuario.** Es el proceso de determinar si una persona o una empresa está autorizada para llevar a cabo una acción dada. Algunos sistemas de autenticación incluyen tanto identificación como autorización, mientras que otros solamente incluyen uno u otro. Cuando usted inserta su tarjeta débito en un ATM, se le pide que ingrese su Número de Identificación Personal (conocido como PIN o NIP). El NIP le indica al Cajero Automático que está autorizado para llevar a cabo la transacción solicitada.

De la misma manera, cuando se desea tener acceso a una porción privada de una red, normalmente se debe escribir el nombre de usuario (identificación) y la contraseña (autorización).

**Autenticación de Datos.** En los sistemas de procesamiento automatizado de datos, a menudo es imposible que alguien determine si los datos han sido

modificados. Esta revisión llevaría demasiado tiempo si se tiene en cuenta la enorme cantidad de datos que manejan hoy en día las aplicaciones de procesamiento de datos.



## Capítulo III: Análisis y Diseño

### 3.1 ESTRUCTURA ORGANIZACIONAL

La empresa "BOLIVIAN BUS" tiene la estructura organizacional que se muestra en la Figura 3.1, donde se puede observar los niveles, el de decisión, operativo y la forma en que están relacionados los siguientes componentes de la misma.



**Figura 3.1:** Organigrama de la empresa BOLIVIAN BUS  
Fuente: Elaboración propia

El nivel de decisión, conformado por el presidente de grupo que es la única que da el visto bueno a cambios en el funcionamiento de la empresa, tales como la habilitación de horarios, planeación del rol de salidas diarias, contrataciones del personal, etc.

El nivel operativo, conformado por los propietarios, personal de operaciones y mantenimiento, administradores, personal de boletería, encomiendas, quienes se encargan de llevar a cabo todas las disposiciones del nivel de decisión.

### 3.2 DESCRIPCION DE FUNCIONES

*PROPIETARIOS.* Cada propietario tiene bajo su responsabilidad las siguientes tareas:

- Pago de seguro de pasajeros y del automotor.
- Control de salidas diarias del bus que le pertenece.
- Verificación de pasajes vendidos por salidas, cantidad y precio.
- Control de importes de encomiendas por cobrar.
- Control de liquidación si el importe a cancelar es el correcto.
- Operación y mantenimiento del bus que les pertenece.
- Pago de sueldos.

*GERENTE GENERAL.* Responsable de llevar el control de la empresa, dado que establecen y controlan al personal tanto de la oficina central como de sus sucursales, a través de informes que rinden los administradores de la central y de cada sucursal.

*PERSONAL DE OPERACIONES.* Dentro de esta sección se encuentra incluido todo el personal encargado del manejo de los buses: chóferes y ayudantes.

- Atención a pasajeros en el viaje.
- Llevar el bus con pasajeros, equipajes y encomiendas a su destino.
- Carga a buzones de equipajes y encomiendas, y entrega de los mismos en destino.
- Entrega de lista de pasajeros a oficina de tránsito durante el viaje.

*ADMINISTRADOR DE SUCURSAL.* Encargado de recaudar todos los ingresos de venta de pasajes y envío de encomiendas, a través de rendición de cuentas diarias que se realiza el personal de boletería de las oficinas a su cargo.

- Estado de cuentas de la oficina a su cargo.
- Cancelación de liquidación por salida a propietarios.
- Recepción de ingresos por venta de pasajes.
- Recepción de ingresos por envío de encomiendas.
- Pago de gastos administrativos (teléfono, luz, alquileres, etc.)
- Sueldos del personal de boletería.

*VENDEDOR DE PASAJES.* Cumplen con las siguientes funciones:

- Venta, reserva y devolución de pasajes.
- Recepción, entrega de equipajes y encomiendas.
- Lista de pasajeros que realizan el viaje.
- Manifiesto de equipajes y encomiendas.
- Recepción y entrega de equipajes.
- Recepción y entrega de encomiendas.

### **3.3 ANALISIS DEL SISTEMA A DESARROLLAR**

#### **3.3.1 MODELO AMBIENTAL**

##### **3.3.1.1 DECLARACION DE PROPOSITO**

El propósito del Sistema de Control y Administración SARIRI para la empresa de transporte de pasajeros internacional "BOLIVIAN BUS", es recoger Y almacenar información necesaria para facilitar y centralizar los procesos administrativos de la empresa. Esto incluye venta de pasajes a los clientes, generación de boletos e informes sobre utilidades, control de salidas y llegadas de los buses, asignación de rutas y destinos así como de horarios.

### 3.3.1.2 DIAGRAMA DE CONTEXTO

#### Sistema de Control y Administración SARIRI - Context Diagram

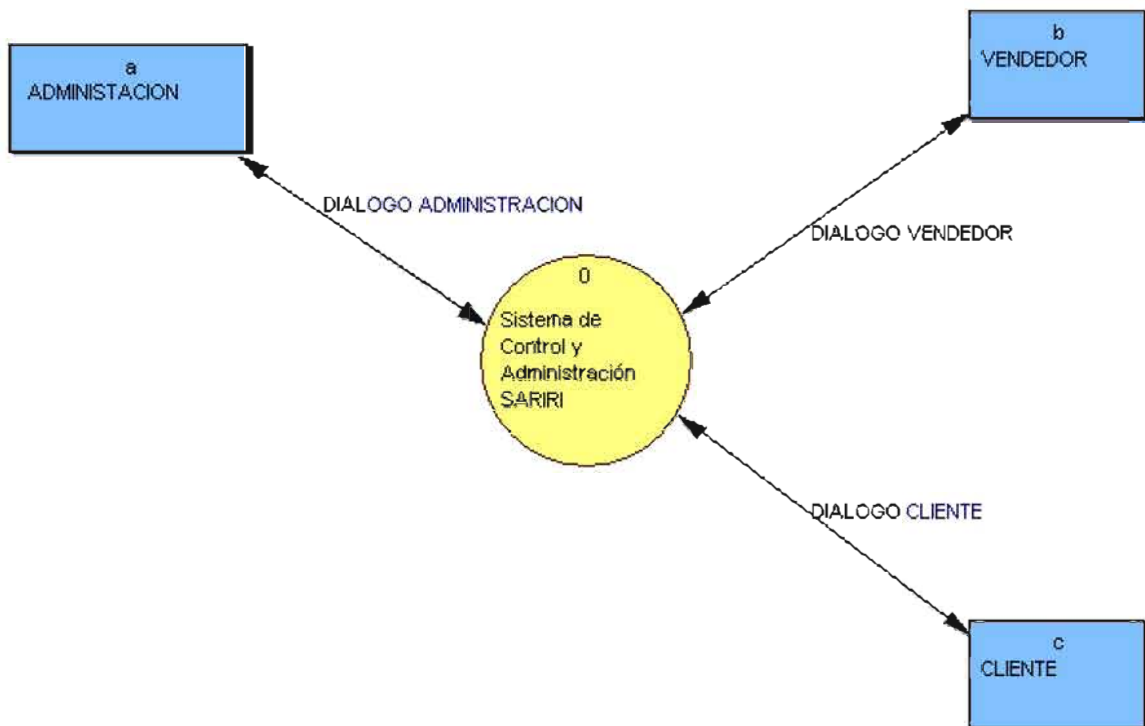


Figura 3.2: Diagrama de Contexto  
Fuente: Elaboración Propia

### 3.3.1.3 LISTA DE ACONTECIMIENTOS

1. La administración asigna destino de buses.
2. La administración elimina información de buses.
3. La administración elimina información de propietarios.
4. La administración elimina información de rutas.
5. La administración elimina información de sucursales.
6. La administración elimina información de vendedores.
7. La administración elimina viajes.
8. La administración modifica destino de buses.
9. La administración modifica información de buses.
10. La administración modifica información de propietarios.
11. La administración modifica información de sucursales.

12. La administración modifica información de vendedores.
13. La administración modifica información sobre rutas.
14. La administración registra nuevas rutas.
15. La administración registra nuevas sucursales.
16. La administración registra nuevos buses.
17. La administración registra nuevos propietarios.
18. La administración registra nuevos vendedores.
19. La administración requiere de un listado de viajes. (T)
20. La administración requiere de un reporte de venta de pasajes. (T)
21. Se verifica la disponibilidad de un asiento. (C)
22. Un cliente desea comprar un pasaje.
23. Un cliente desea confirmar su reserva.
24. Un cliente desea devolver su pasaje.
25. Un cliente pide la cancelación del envío de su encomienda.
26. Un cliente pide la reserva de un asiento.
27. Un cliente quiere enviar/recoger una encomienda.
28. Un cliente solicita el canje de su pasaje por otro.
29. Un cliente solicita la anulación de su reserva.
30. Un vendedor anula la reservación de un asiento.
31. Un vendedor anula un pasaje.
32. Un vendedor anula una encomienda.
33. Un vendedor confirma la reservación de un cliente.
34. Un vendedor modifica la información de un pasaje vendido.
35. Un vendedor modifica los datos de una encomienda registrada.
36. Un vendedor registra la entrega de una encomienda a destinatario.
37. Un vendedor registra menores de edad.
38. Un vendedor registra una compra de pasaje de parte de un cliente.
39. Un vendedor registra una encomienda para su envío.
40. Un vendedor registra una reservación de asiento de parte de un cliente.
41. Un vendedor requiere de la nomina de pasajeros. (T)
42. Un vendedor requiere de un reporte de venta de pasajes. (T)



3.3.2 MODELO DE COMPORTAMIENTO

Sistema de Control y Seguimiento SARIRI (Level 1 Diagram)

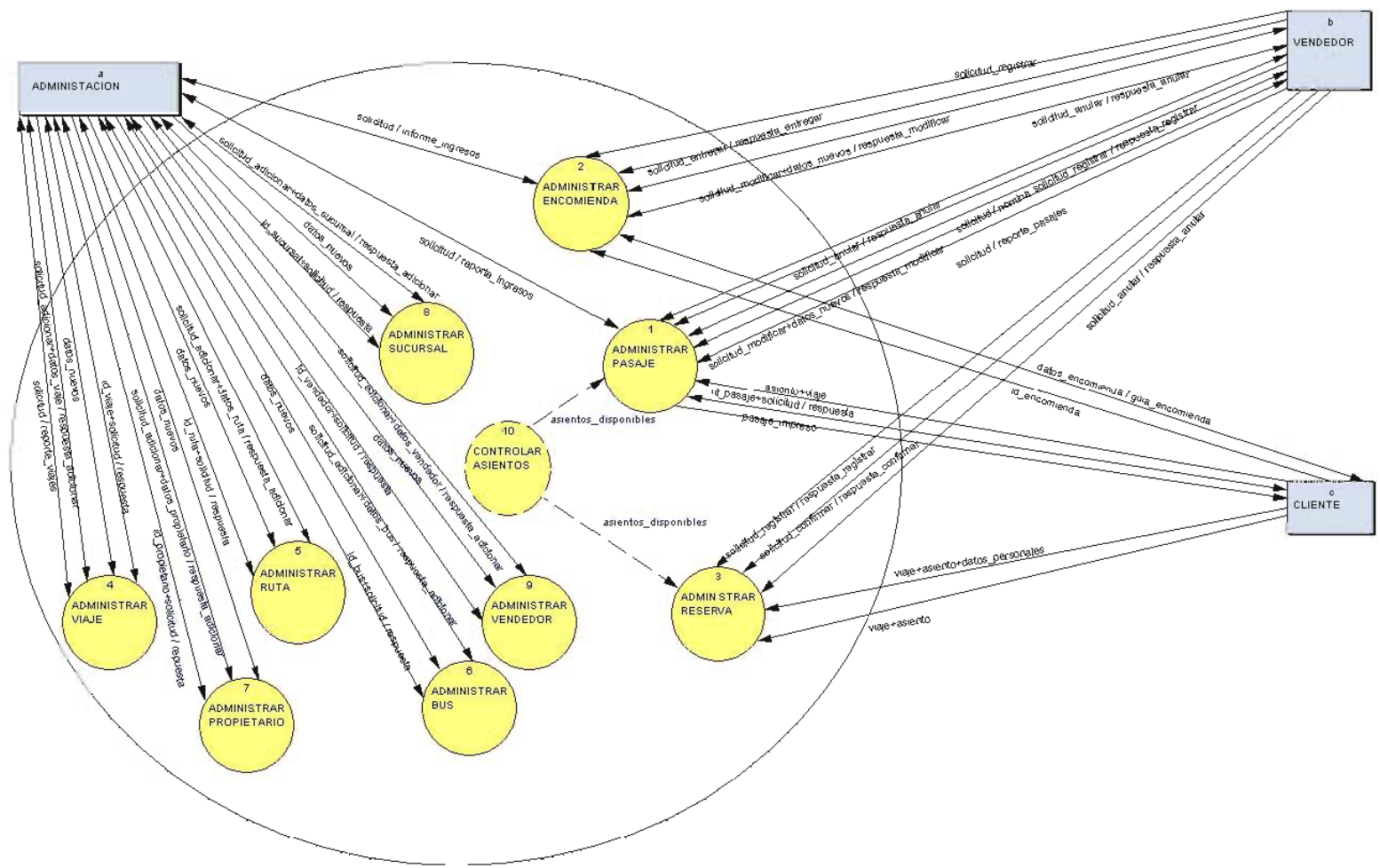
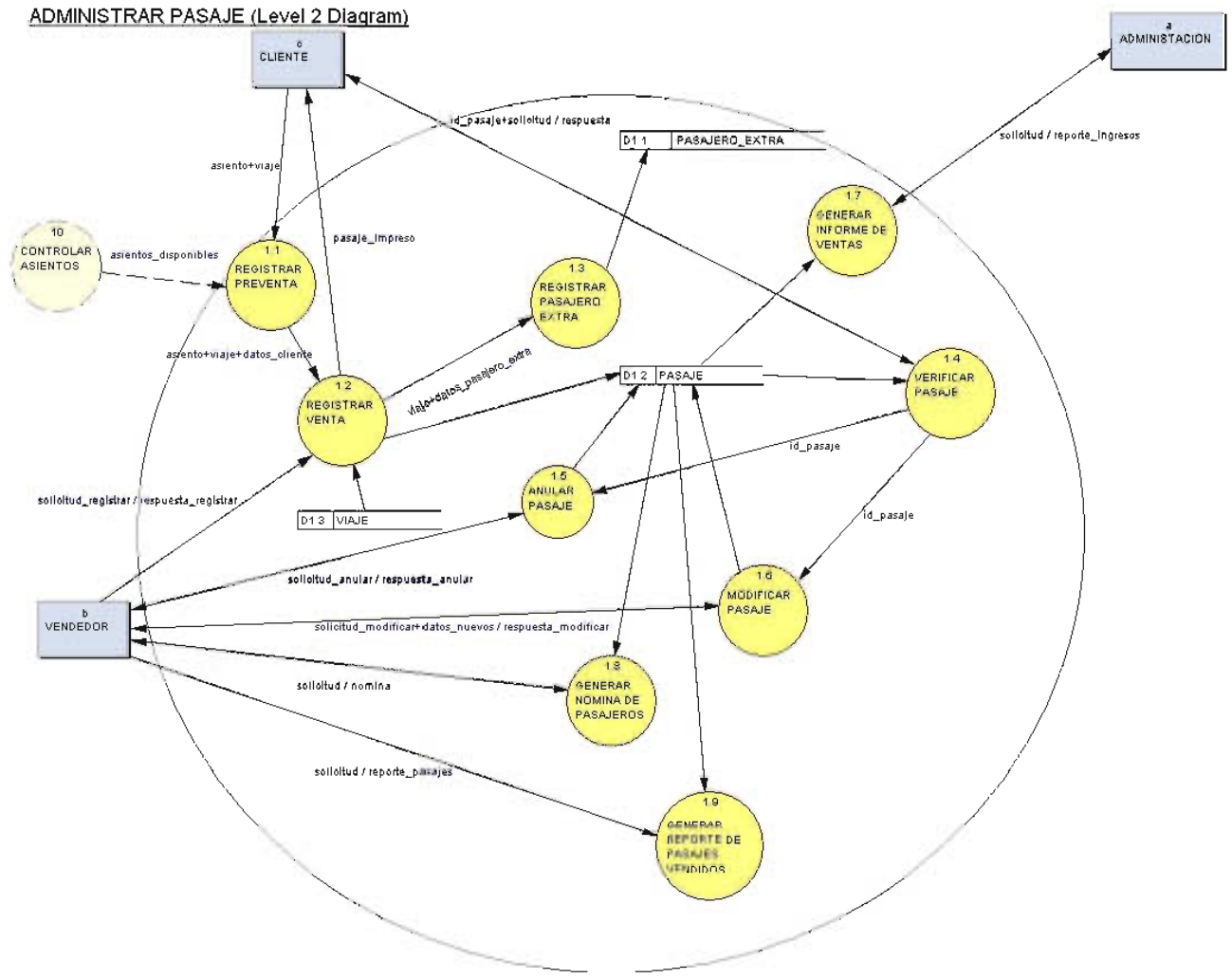
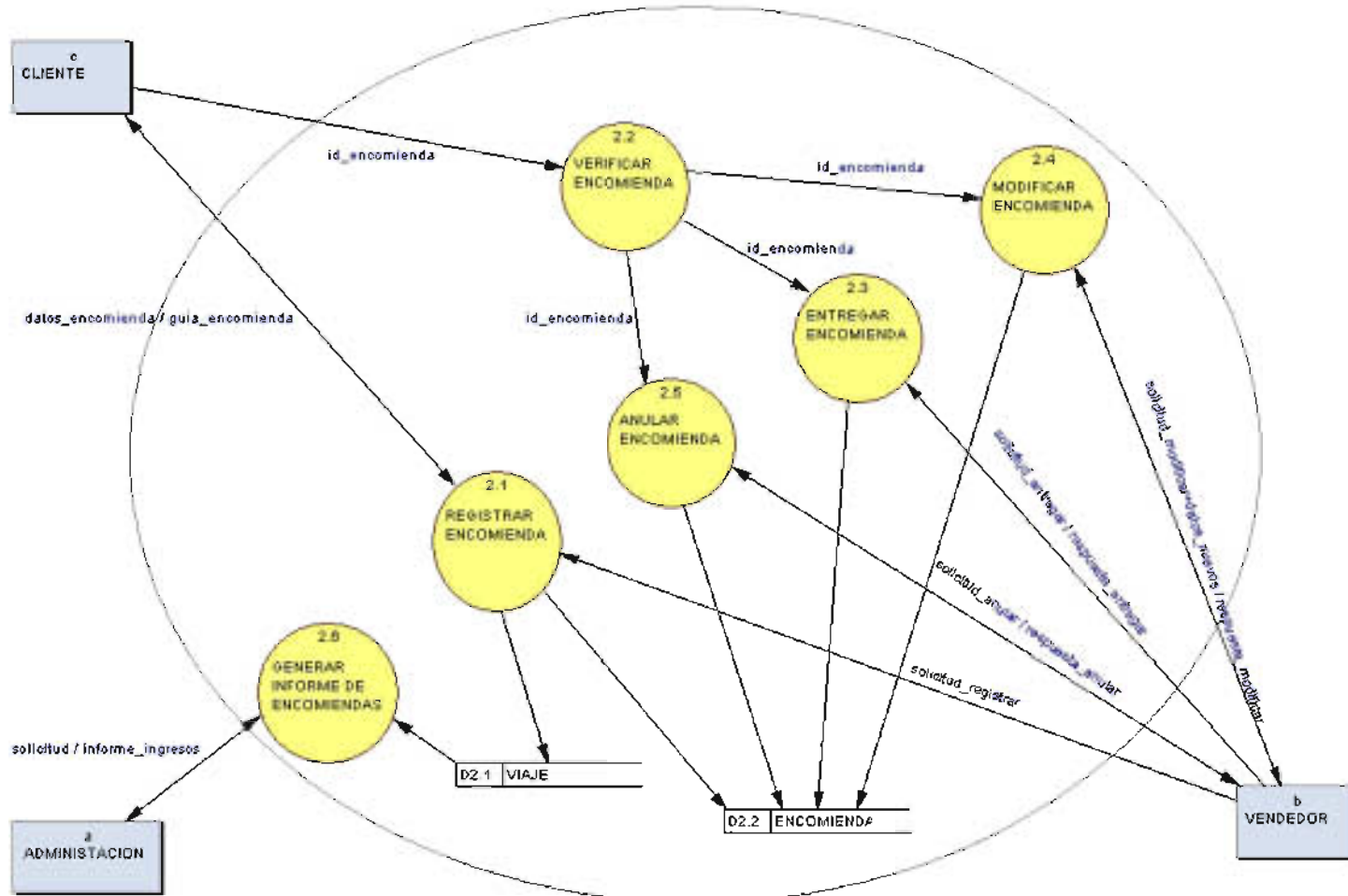


Figura 3.3: Diagrama de Flujo de Datos (Nivel 1)  
Fuente: Elaboración Propia



**Figura 3.4:** Diagrama de Flujo de Datos – Administrar Pasaje (Nivel 2)  
Fuente: Elaboración Propia

**ADMINISTRAR ENCOMIENDA (Level 2 Diagram)**



**Figura 3.5:** Diagrama de Flujo de Datos – Administrar Encomienda (Nivel 2)  
**Fuente:** Elaboración Propia

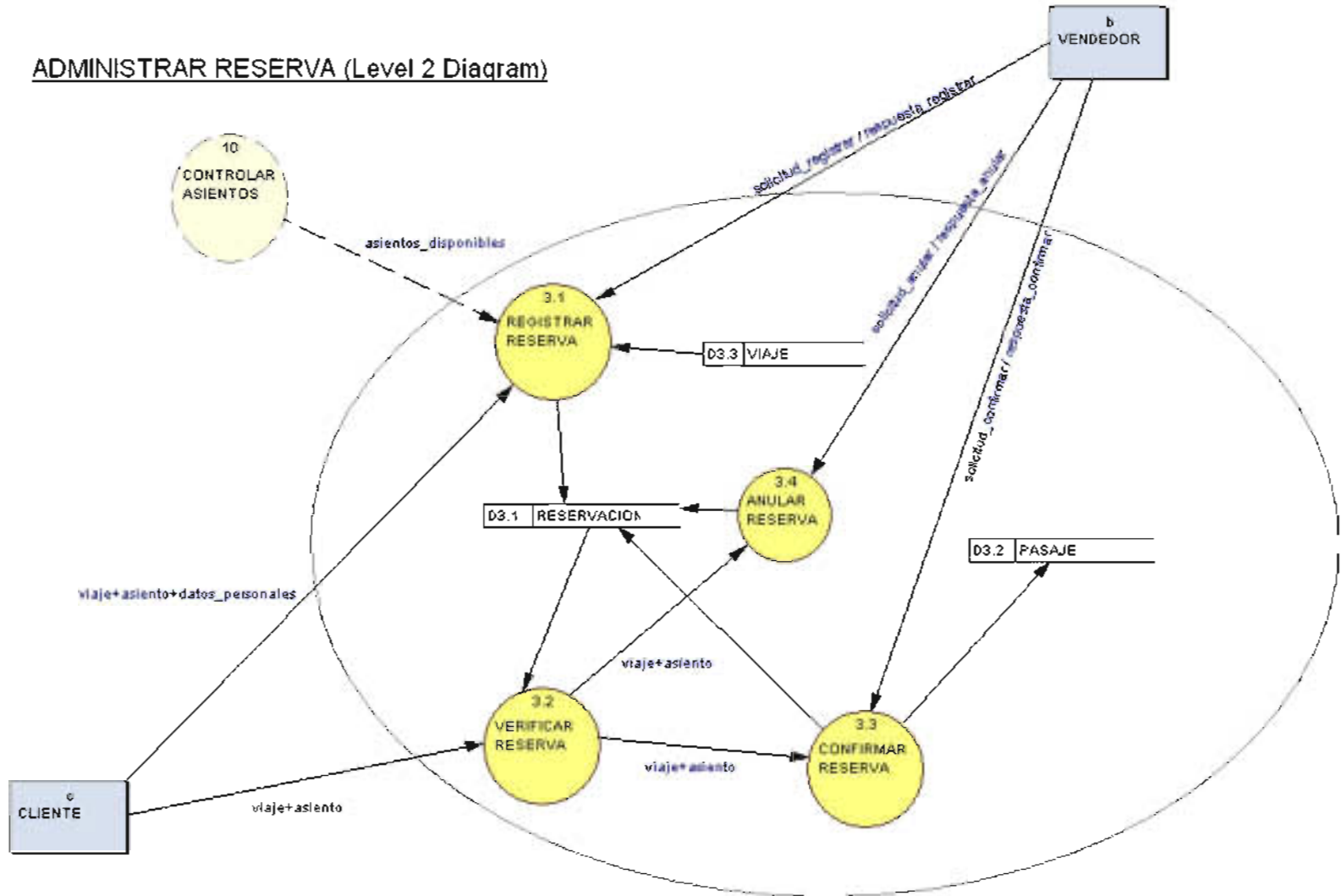


Figura 3.6: Diagrama de Flujo de Datos – Administrar Reserva (Nivel 2)  
Fuente: Elaboración Propia

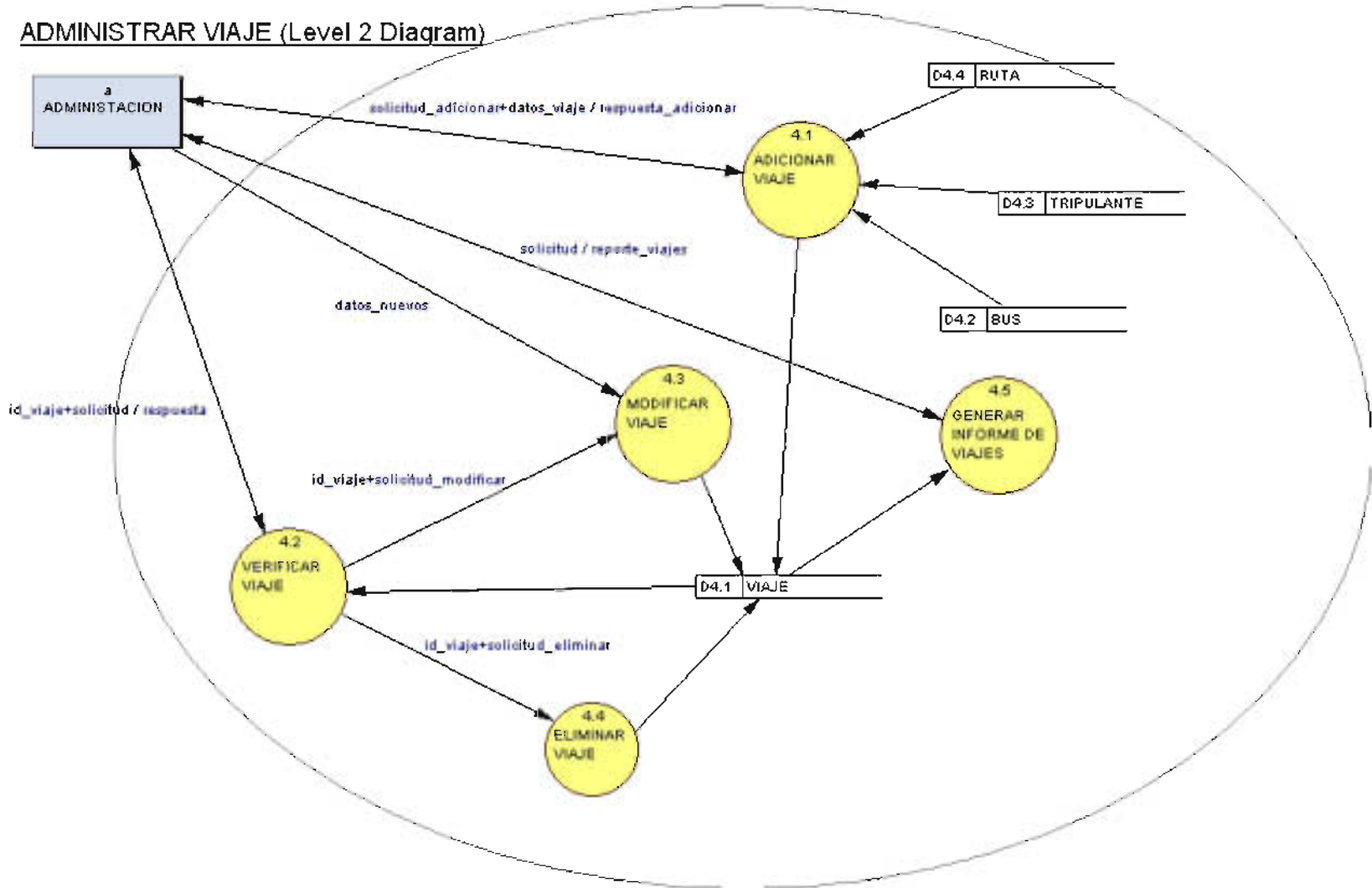


Figura 3.7: Diagrama de Flujo de Datos – Administrar Viaje (Nivel 2)  
Fuente: Elaboración Propia

### ADMINISTRAR RUTA (Level 2 Diagram)

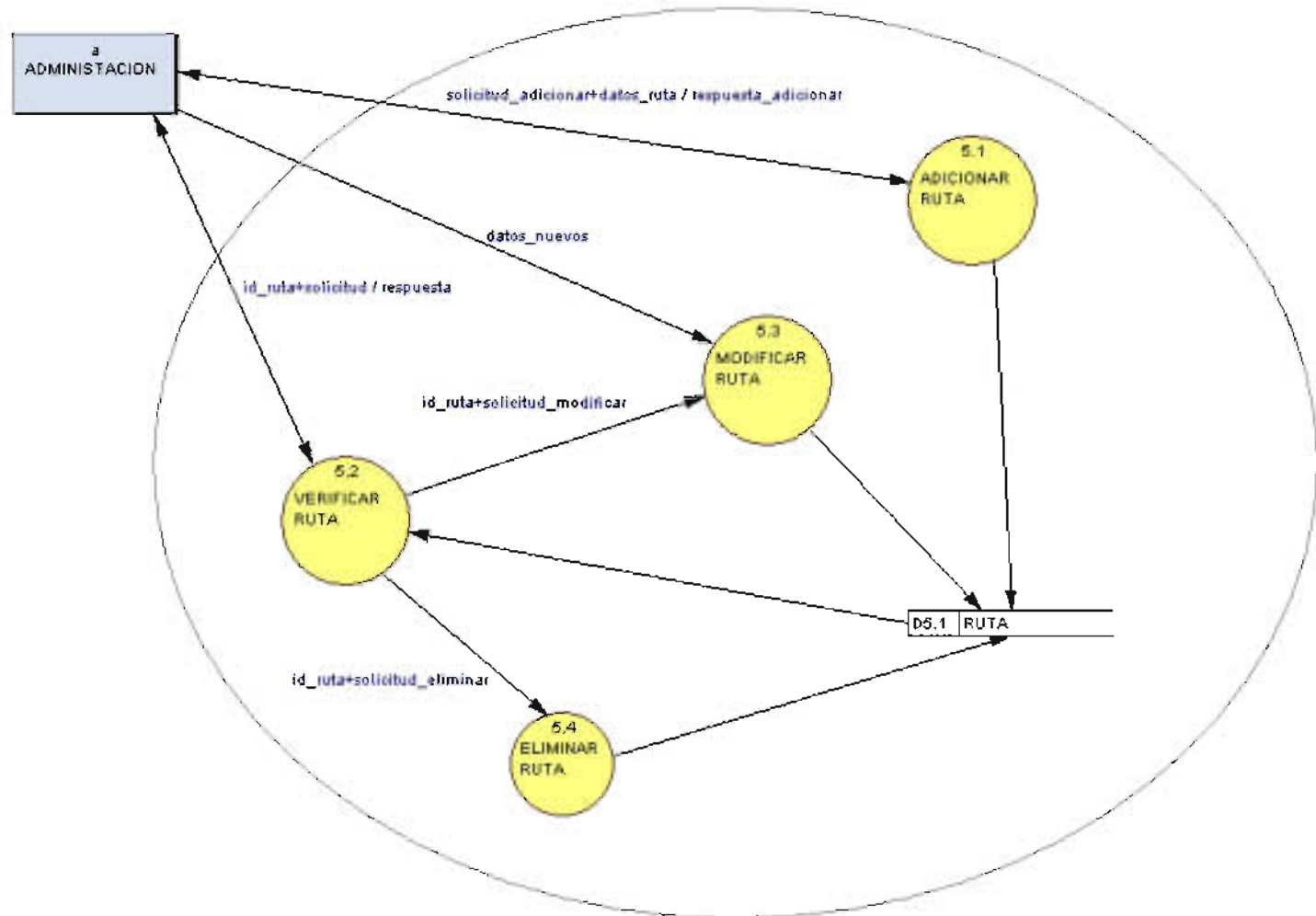


Figura 3.8: Diagrama de Flujo de Datos – Administrar Ruta (Nivel 2)  
Fuente: Elaboración Propia

### ADMINISTRAR BUS (Level 2 Diagram)

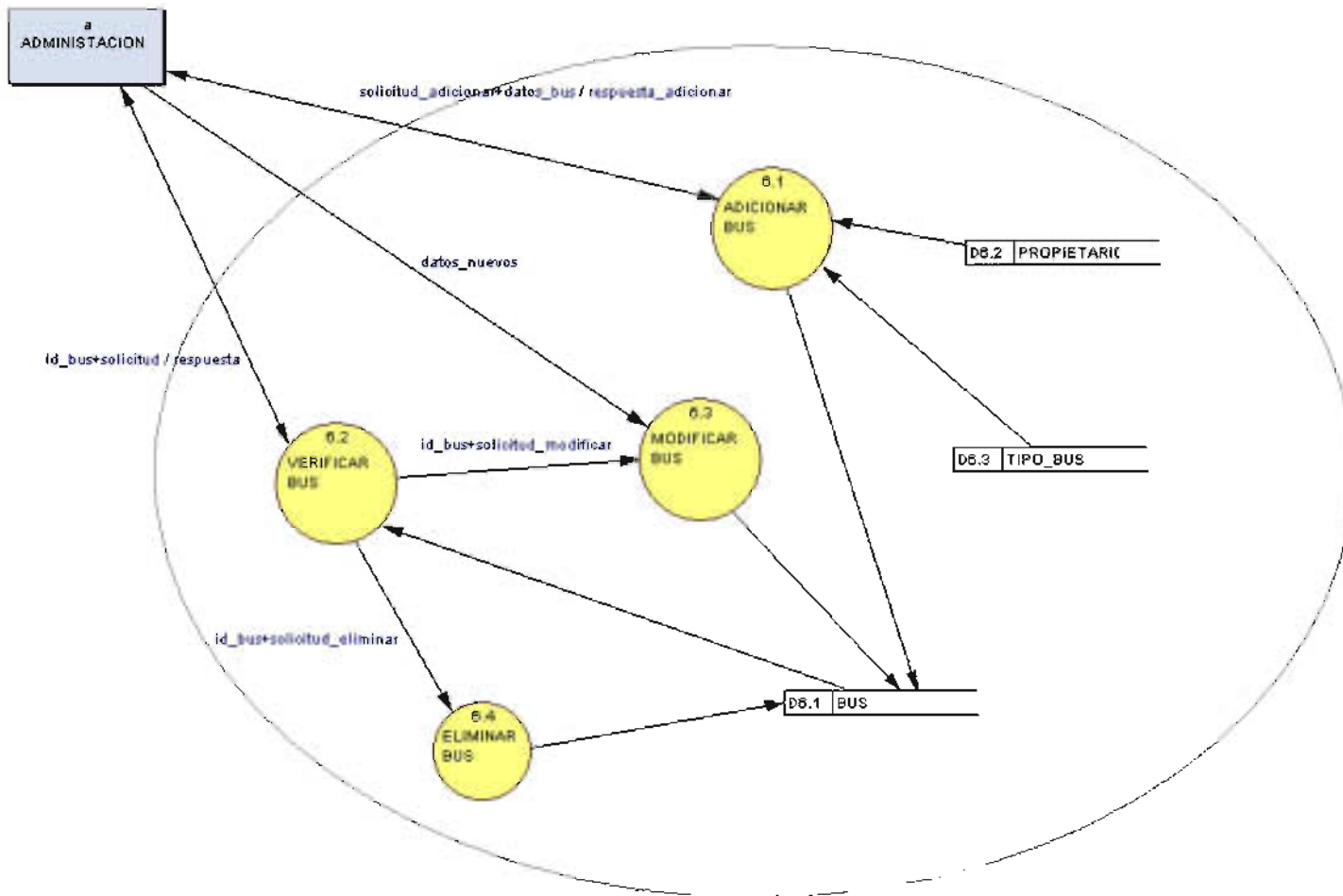


Figura 3.9: Diagrama de Flujo de Datos – Administrar Bus (Nivel 2)  
Fuente: Elaboración Propia

ADMINISTRAR PROPIETARIO (Level 2 Diagram)

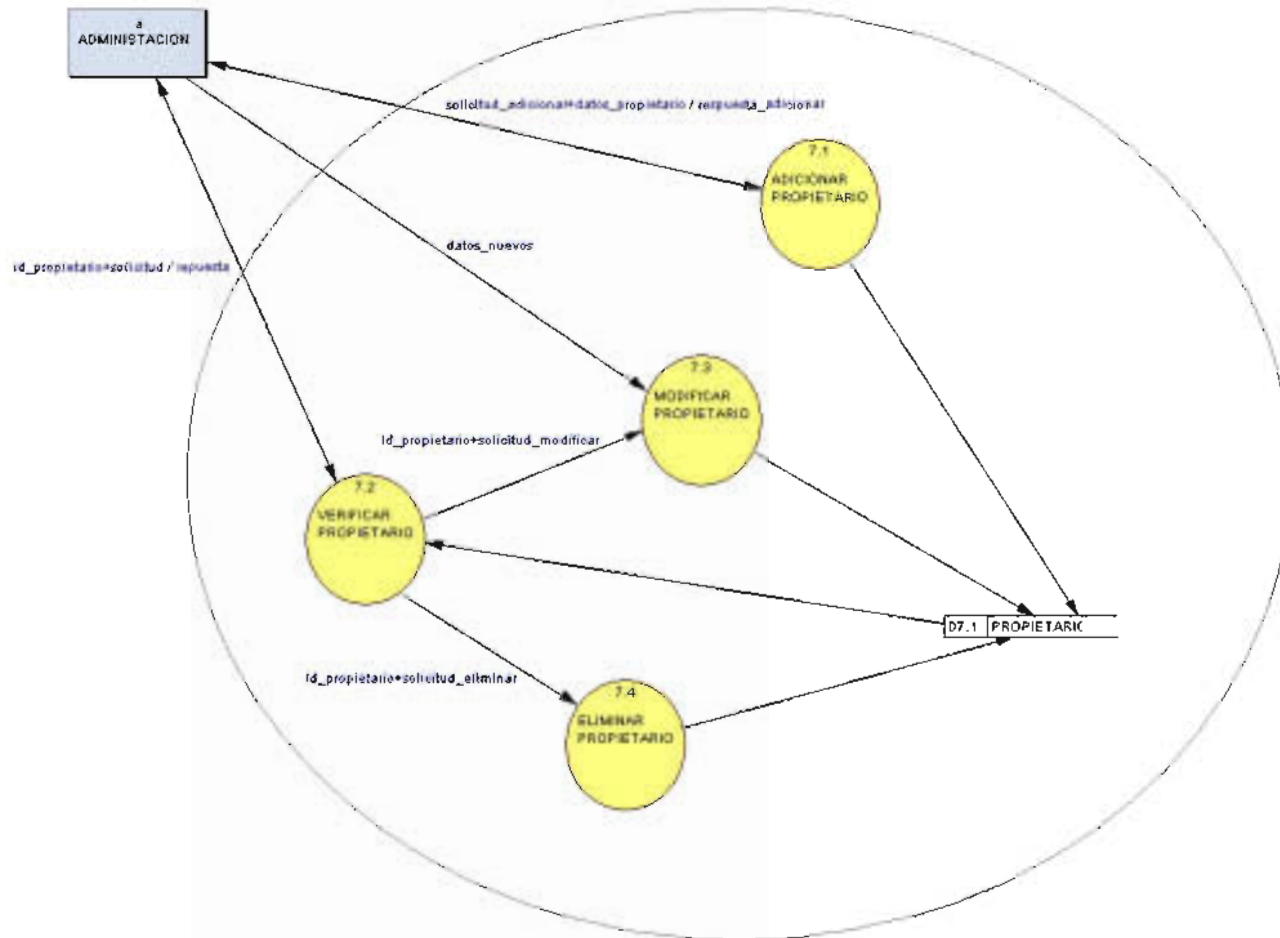


Figura 3.10: Diagrama de Flujo de Datos – Administrar Propietario (Nivel 2)  
Fuente: Elaboración Propia



ADMINISTRAR SUCURSAL (Level 2 Diagram)

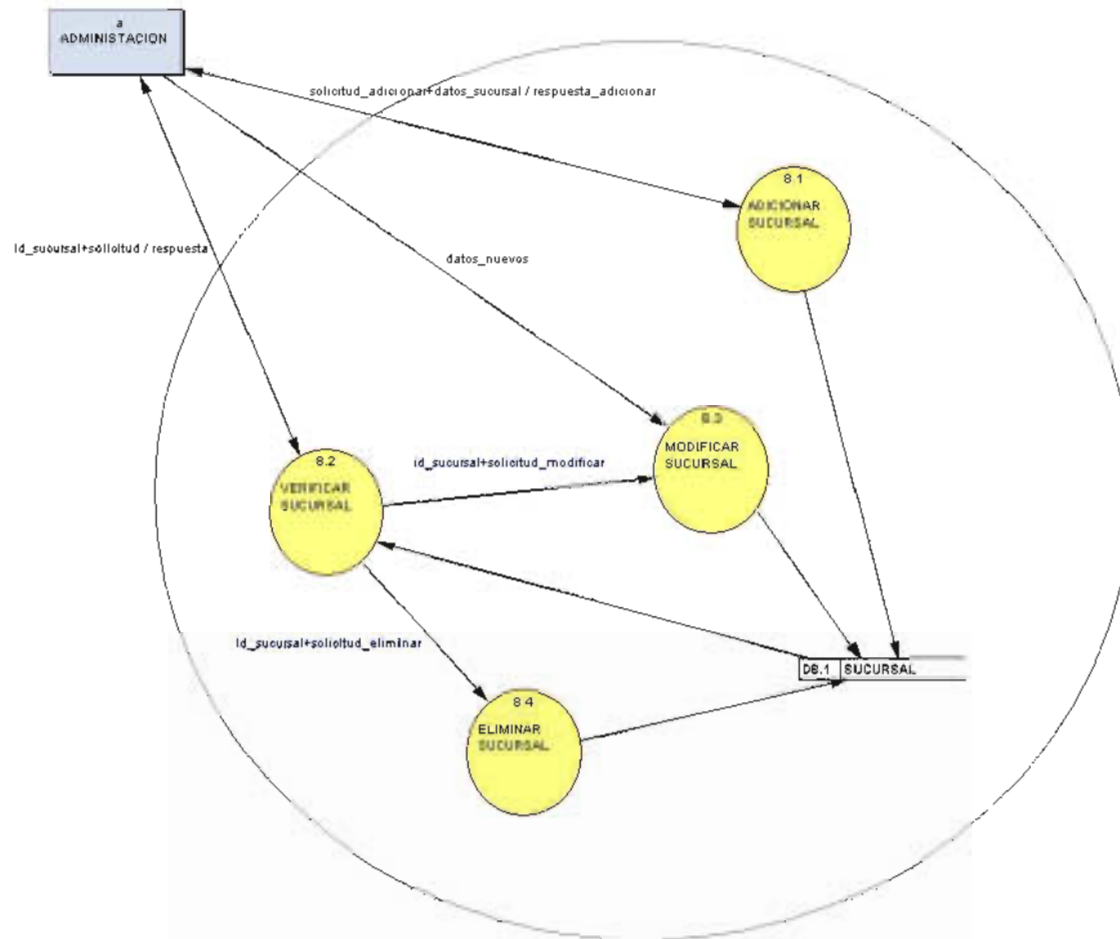


Figura 3.11: Diagrama de Flujo de Datos – Administrar Sucursal (Nivel 2)  
Fuente: Elaboración Propia

ADMINISTRAR VENDEDOR (Level 2 Diagram)

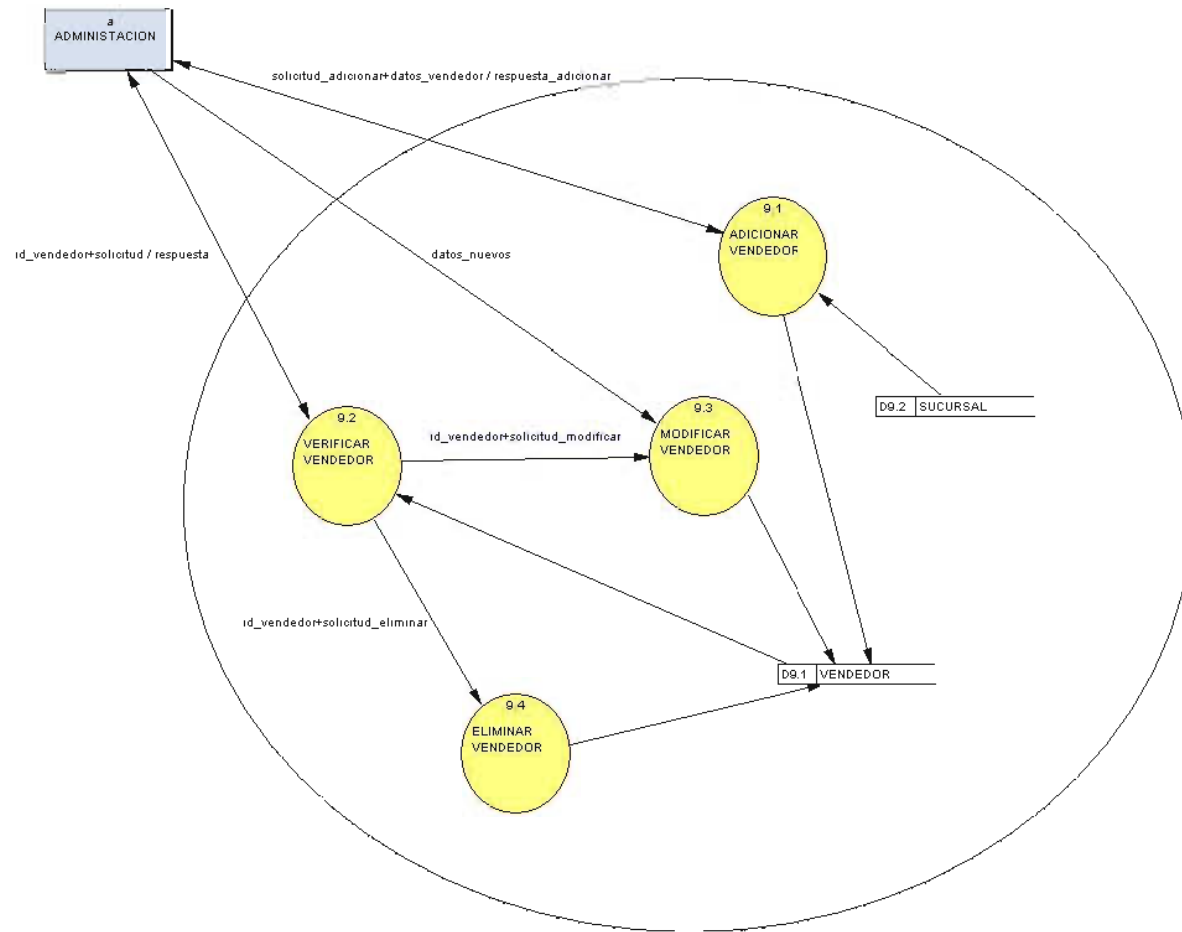


Figura 3.12: Diagrama de Flujo de Datos – Administrar Vendedor (Nivel 2)

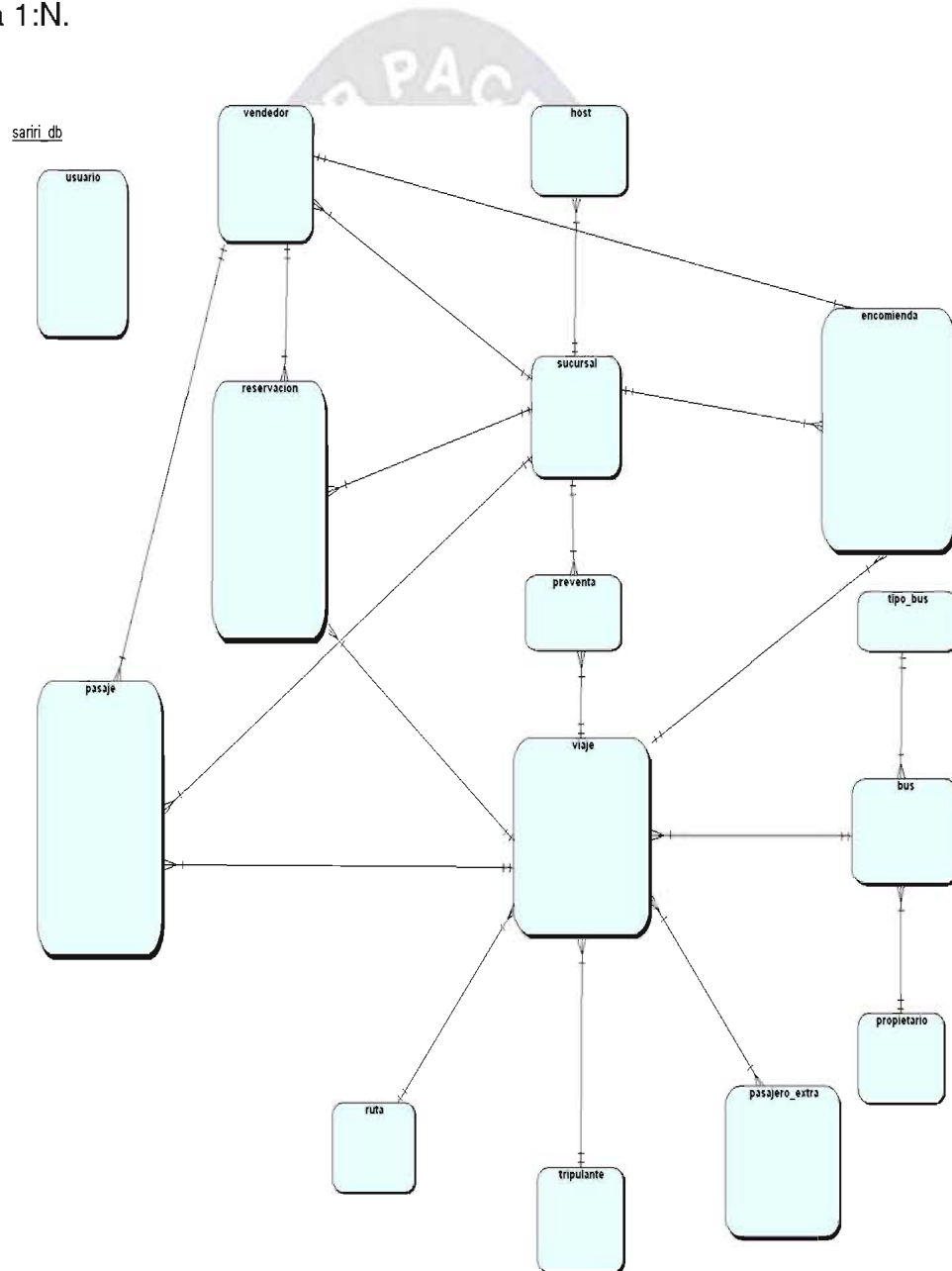
Fuente: Elaboración Propia



### 3.5 MODELO WEB DEL SISTEMA

#### Etapa 1: Diseño Entidad-Relación

En base al diagrama mostrado en el apartado 3.4.1, el diagrama de la figura 3.14 corresponde al diagrama entidad relación con todas las relaciones de la forma 1:N.



**Figura 3.14:** Diagrama de Entidad-Relación

**Fuente:** Elaboración Propia

*Etapa 2: Diseño de slices:*

A continuación se muestran los slices (definidos en el modelo RMM) y los m-slices (definidos en el modelo DRM) definidos para la aplicación.

```
VIAJE.principal: m-slice
begin
  id;
  fecha;
  hora;
  hora_llegada;
  hora_salida;
  id_bus;
  observaciones;
  [con] VIAJE.ingreso ;
end;
```

```
RUTA.principal: slice
begin
  origen;
  destino;
  via;
end;
```

```
TRIPULANTE.principal: slice
begin
  paterno;
  materno;
  nombre;
  licencia;
end;
```

```
BUS.principal: m-slice
begin
  placa;
  asiento;
  [tiene_un] PROPIETARIO.principal;
  [es] TIPO_BUS.descripcion
end;
```

```
PROPIETARIO.principal: slice
begin
  *id index buses begin
    Relation: [tiene];
    Content: BUS.principal;
  index end
  paterno;
  materno;
  nombre;
end;
```



```
SUCURSAL.principal: slice
begin
  *id      index vendedores begin
            Relation: [tiene];
            Content: VENDEDOR.principal;
            index end
  pais;
  ciudad;
  direccion;
  telefono;
end;
```

```
VENDEDOR.principal: m-slice
begin
  paterno;
  materno;
  nombre;
end;
```

```
PASAJE.ingreso: m-slice
begin
  id;
  precio;
  id_viaje;
  observacion;
  [tiene] BUS.id;
end;
```

```
ENCOMIENDA.ingreso: m-slice
begin
  id;
  precio_envio;
  id_viaje;
  observacion;
  [tiene] BUS.id;
end;
```

```
VIAJE.ingreso: m-slice
begin
  id;
  [contiene] PASAJE.ingreso;
  [contiene] ENCOMIENDA.ingreso;
end;
```



Etapa 3: Diseño navegacional

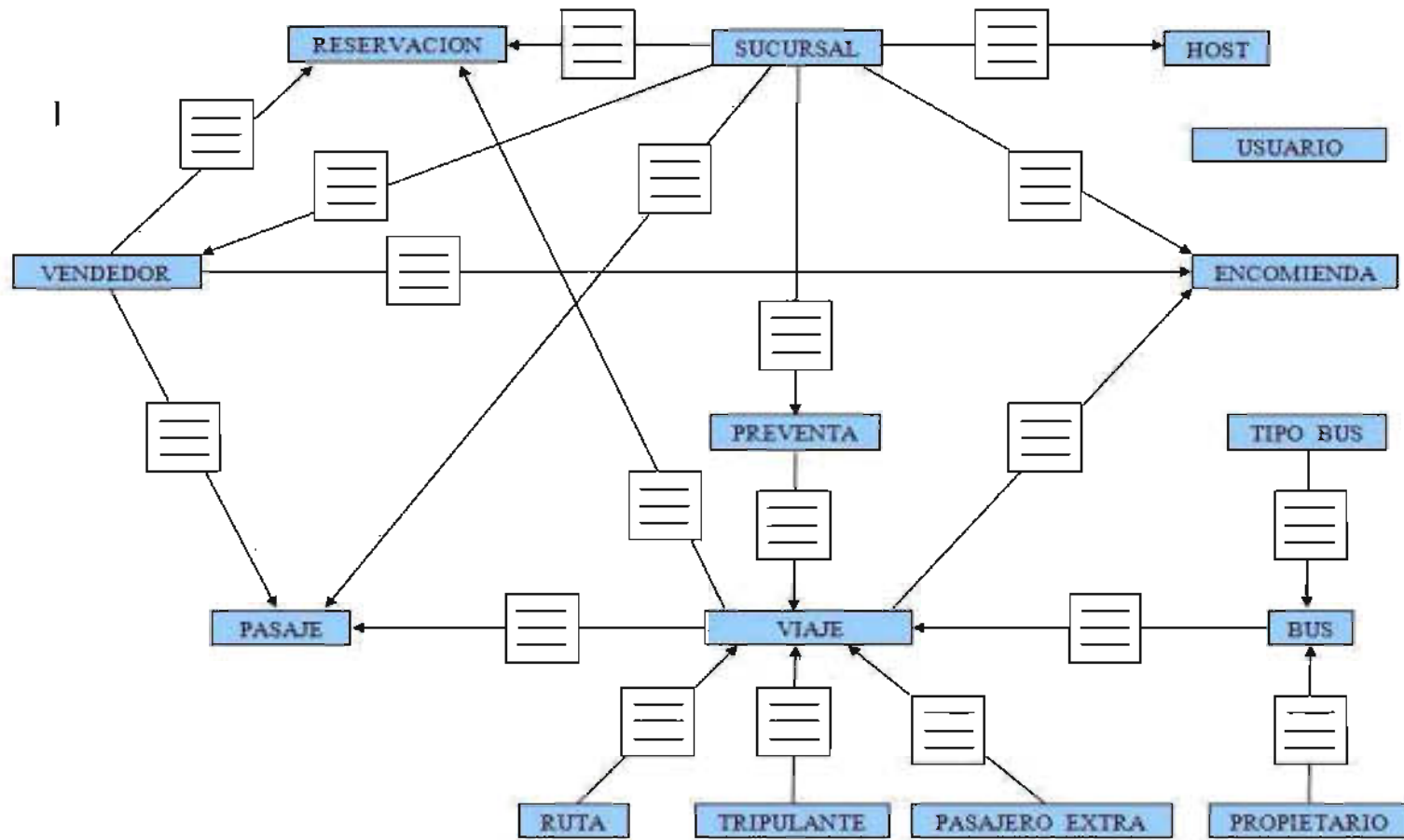


Figura 3.15: Diagrama RMDM  
Fuente: Elaboración Propia

Etapa 3: Diseño navegacional con menú

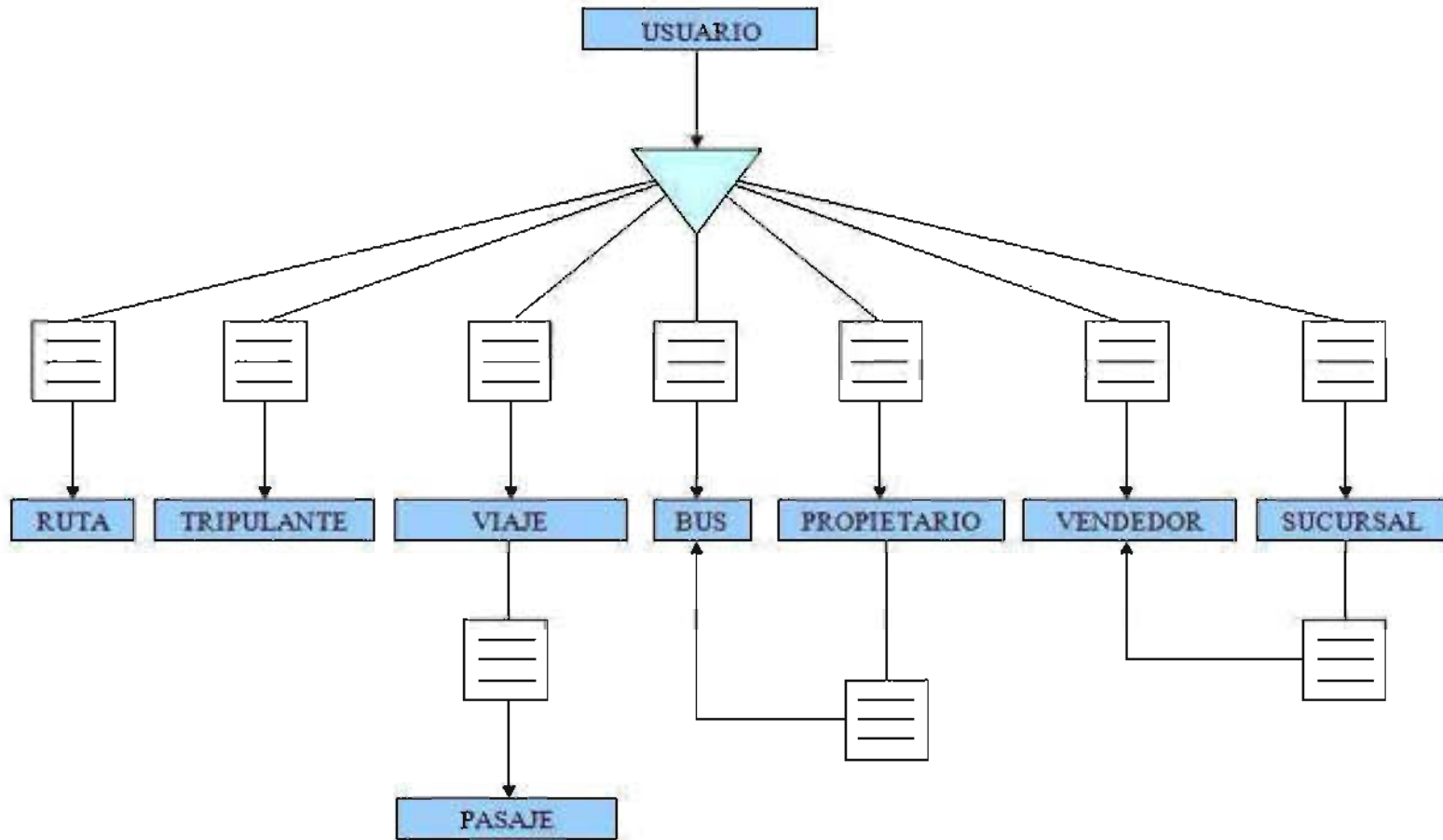


Figura 3.16: Diagrama RMDM con menú  
Fuente: Elaboración Propia



*Etapa 4: Diseño de protocolos de conversión*

En este caso hay una base de datos conectado al servidor web, las entidades se convierten en tablas de la base de datos y los atributos en columnas de esas tablas. En cuanto a las estructuras de acceso, los índices se convierten en resultados de consultas a la base de datos (son variables según el contenido de la misma).

**3.6 RECURSOS DE HARDWARE Y SOFTWARE**

**3.6.1 RECURSOS DE HARDWARE**

Para implantar el sistema de Control y Administración SARIRI (tanto la aplicación win32 como la aplicación web), se requiere componentes de hardware, los cuales deben cumplir con algunas características específicas mínimas debido a las exigencias en cuanto a recursos de hardware que necesita el sistema. A continuación se tiene los componentes mínimos que se necesita para la implementación del sistema desarrollado:

APLICACION WIN32
<i>Procesador Pentium III de 1,7 Ghz                  Memoria RAM de 128 Mb                  Disco Duro de 20 Gb                  Monitor de 17"                  Tarjeta de Red                  Teclado                  Ratón                  Impresora</i>
APLICACION WEB
<i>Servicio de Hosting con las siguientes características:                  Procesador Intel Celeron 2.26 GHz                  Memoria RAM DDR 256 MB                  5 GB de Espacio en disco                  30 GB de transferencia</i>

**Figura 3.17:** Requerimientos mínimos de Hardware  
**Fuente:** Elaboración Propia

### 3.6.2 RECURSOS DE SOFTWARE

Para la etapa de implementación consideramos el siguiente software, el cual es capaz de responder a múltiples requerimientos.

#### *Aplicación WIN32*

La plataforma de soporte es Windows XP, debido a la gran difusión de este sistema operativo.

El lenguaje que se utilizó para el desarrollo de la aplicación es Visual FoxPro 9.0 Service Pack 1, debido a que nos ofrece un conjunto de herramientas para crear aplicaciones de bases de datos tanto para escritorio como para entornos cliente/servidor.

Se utilizó una conexión ODBC (Driver MyODBC 3.51) para poder interactuar con el servidor de bases de datos.

#### *Aplicación Web*

Esta aplicación fue desarrollada en función a las características del servicio de Hosting contratado por la institución, por lo tanto, podemos mencionar lo siguiente:

La plataforma de soporte es Linux, ya que es un sistema operativo estable y seguro y el más utilizado por servidor a nivel mundial.

El software de servidor será Apache, por ser un producto de alto rendimiento y utilizado por la mayoría de los servidores que ofrecen servicio de hosting.

El lenguaje que se utilizó para la construcción del sistema es PHP, por ser un poderoso lenguaje de programación en aplicaciones web, con soporte de funciones nativas para acceder a diferentes motores de bases de datos.

Además se utilizó HTML, Javascript, Hojas de estilos para poder estructurar la aplicación, así mismo para darle más forma y funcionalidad a la misma.

Tanto la Aplicación Web como la Aplicación WIN32 utilizan el servidor de bases de datos MySQL, por ser un motor con gran capacidad de almacenamiento.

Para la parte gráfica se utilizó Adobe Photoshop CS tanto para la aplicación web como para la aplicación WIN32, esto con el fin de realizar una interfaz más amigable para los usuarios finales.

En cuanto a recursos de Internet se necesitará una conexión permanente de 128 kbps (mínimo).

### **3.7 INTERFAZ DE USUARIO**

El diseño de la interfaz de usuario fue desarrollado de manera transparente al usuario, de tal manera que puede ser utilizado incluso por personas que no tengan conocimiento del área, esto con un mínimo de capacitación.

A continuación se presentan las pantallas más significativas del sistema.

**APLICACION WIN32**



**Figura 3.18:** Aplicación WIN32 – Autenticación de Usuario  
**Fuente:** Elaboración Propia

PASAJES			N° DE BOLETO	PRECIO	
N° DE VIAJE 74		BUS N° 0			
			1737	5000.00	
			1738	5000.00	
			1739	6000.00	
			1740	6000.00	
			1741	7000.00	
			1742	7000.00	
			1743	6000.00	
			1744	6000.00	
			1745	6000.00	
			1746	6000.00	
			1747	5000.00	
			1748	5000.00	
			1749	8000.00	
			1750	6000.00	
			1751	6000.00	
			1752	5000.00	
				TOTAL PARCIAL	93000.00
N° DE VIAJE 75		BUS N° 0			
			1753	10000.00	
			1754	10000.00	
			1755	10000.00	
			1756	7000.00	
			1757	7000.00	
			1758	6000.00	
			1759	6000.00	
			1760	6000.00	
			1761	6000.00	
			1762	10000.00	
			1763	10000.00	
			1764	10000.00	
			1765	10000.00	
			1766	7000.00	
			1768	7000.00	
				TOTAL PARCIAL	122000.00
N° DE VIAJE 79		BUS N° 101			
			1767	7000.00	
				TOTAL PARCIAL	7000.00
				SUBTOTAL	222000.00

**Figura 3.19:** Aplicación WIN32 – Reporte de Ingresos  
**Fuente:** Elaboración Propia

1 - SARIRI - Bolivian Bus - VENDEDOR: MALDONADO JAIME

ChilebusArica S.A. INTERNACIONAL Bolivian Bus

CONDUCTOR: BALTAZAR CONDORI EDWIN LICENCIA: 4052145 ID VIAJE: 74  
 CONDUCTOR: VARGAS ZARATE RONALD LICENCIA: 4904378 ID BUS: 103  
 VIA: CHUNGARA - TAMBO QUEMADO Fecha / Hora de Salida: 26/03/2007 12:40:34  
 ORIGEN: ARICA DESTINO: LA PAZ PLACA: 1604-KEN

ASTO	HOMBRE Y APELLIDOS	N° BOL.	DESTINO	NAC.	F. NAC.	N° DOC. ID.	OBSERVACIONES
2	ISIDORA CHAMBILLA RAMOS	1878	LA PAZ	BOLIVIANA	10/01/2048	354478	
3	MARIA BALINAS SANCHEZ	1741	LA PAZ	CHILENA	1903/1978	13357371-2	
4	EMETERIO MURANA CALCINA	1742	LA PAZ	BOLIVIANA	03/03/1977	4014437	
5	TELESFORO NAJALLO TOLA	1745	LA PAZ	BOLIVIANA	04/01/1973	6079159	
6							
7							
8	JORGE MACHICAO TAMAYO	1726	LA PAZ	CHILE	07/10/1954	14619924-9	
9	ISRAEL PEÑALOZA GONZALEZ	1732	LA PAZ	BOLIVIA	18/11/1992	5207475	
10							
11	HUMBERTO QUERRERO GARIDO	1728	LA PAZ	CHILENO	18/09/1946	6023011-2	
12	PEDRO BARRAHONA VALDIVIA	1730	LA PAZ	CHILENO	28/08/1926	3614723-6	
13	ZADIA JIMENEZ TOCO	1737	LA PAZ	BOLIVIANA	04/03/1978	5253446	
14	ZENON JIMENEZ UGARTE	1738	PATACAMAY	BOLIVIANA	28/03/1956	3004048	
15	LUCIA MOLLO JIMENEZ	1752	PATACAMAY	BOLIVIA	23/09/1959	2316263	
16	ALEJANDRO MENDOZA ZARARE	1731	LA PAZ	BOLIVIA	30/09/1977	4744637	
17	VICTORIA HILARIA CONTRERAS	1748	LA PAZ	BOLIVIANA	13/09/1992	7300822	
18	ELENA SEPULVEDA MORA	1709	LA PAZ	CHILE	20/03/1944	6053285-1	
19	ELENA SEPULVEDA MORA	1709	LA PAZ	CHILE	02/10/1927	3052314-2	
20	ELVIRA SEPULVEDA MORA	1704	LA PAZ	CHILE	28/02/1941	6048201-9	
21	LUIS QUINTANILLA MARCA	1740	LA PAZ	BOLIVIA	04/03/1971	4945945	
22							
23							
24	VICTORIA OLIN	1739	LA PAZ	INGLESA	28/02/1974	700113126	
25	OLGA PINILLA BALLADOS	1733	LA PAZ	CHILENA	14/07/1991	5064078-2	
26	DIEGO SANTANDER PINILLA	1734	LA PAZ	CHILENO	28/02/1976	12692249-6	
27	NIEVES TAPIA MOLLO	1743	PATACAMAY	BOLIVIANA	04/08/1975	5749189	

Figura 3.20: Aplicación WIN32 – Nómina de Pasajeros  
 Fuente: Elaboración Propia

1 - SARIRI - Bolivian Bus - VENDEDOR: MALDONADO JAIME

ChilebusArica S.A. INTERNACIONAL Bolivian Bus

Venta de Pasajes

Origen: ARICA Destino: LA PAZ CCHUWANDA N° de Boleto: 2324

Viaje: 74 | 1604-KEN | 42 | ARICA | LA PAZ

N° de Asientos: 28 / 30

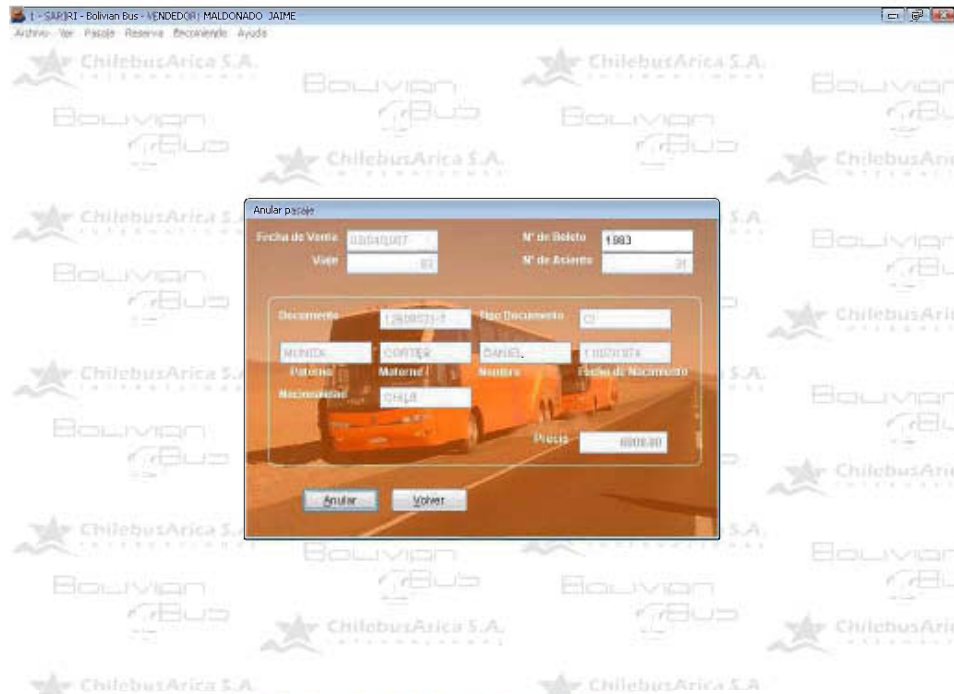
Documento: 4757226 Tipo de Documento: CI

Nombre: ANDRES SANCHEZ ARANA Fecha de Nacimiento: 12/12/1981

Nacionalidad: BOLIVIA Destino: LA PAZ Precio: 8000.00

Registrar Volver

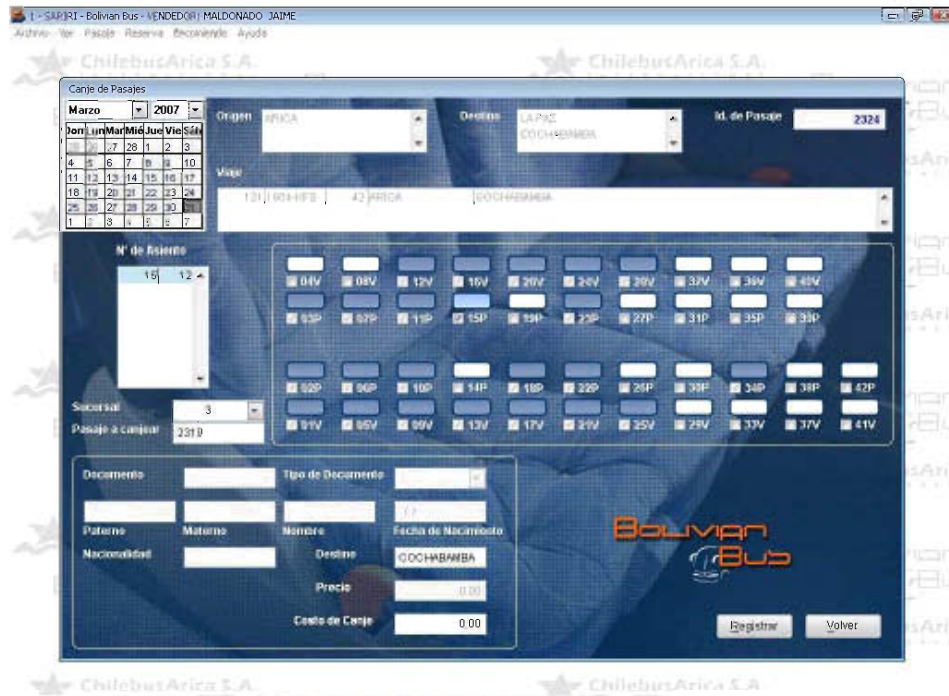
Figura 3.21: Aplicación WIN32 – Formulario de venta de Pasajes  
 Fuente: Elaboración Propia



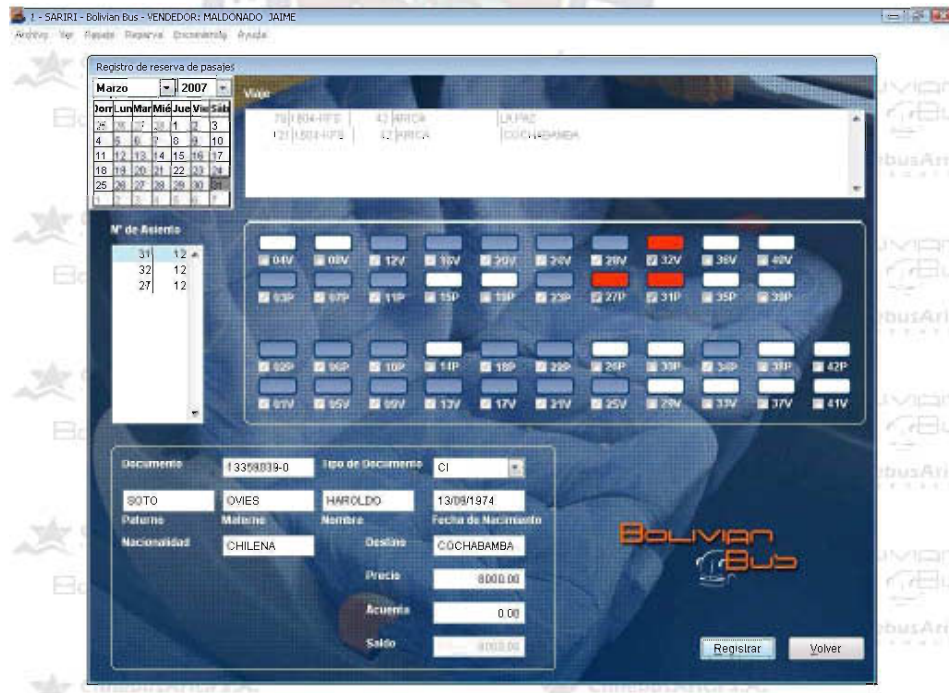
**Figura 3.22:** Aplicación WIN32 – Formulario de anulación de Pasajes  
**Fuente:** Elaboración Propia



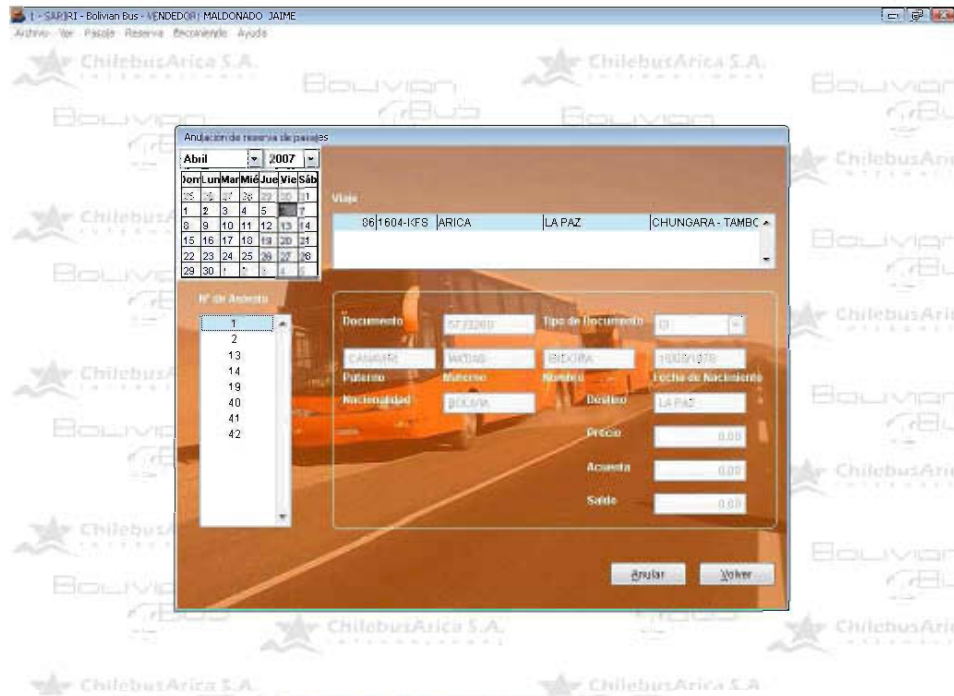
**Figura 3.23:** Aplicación WIN32 – Formulario de duplicado de Pasajes  
**Fuente:** Elaboración Propia



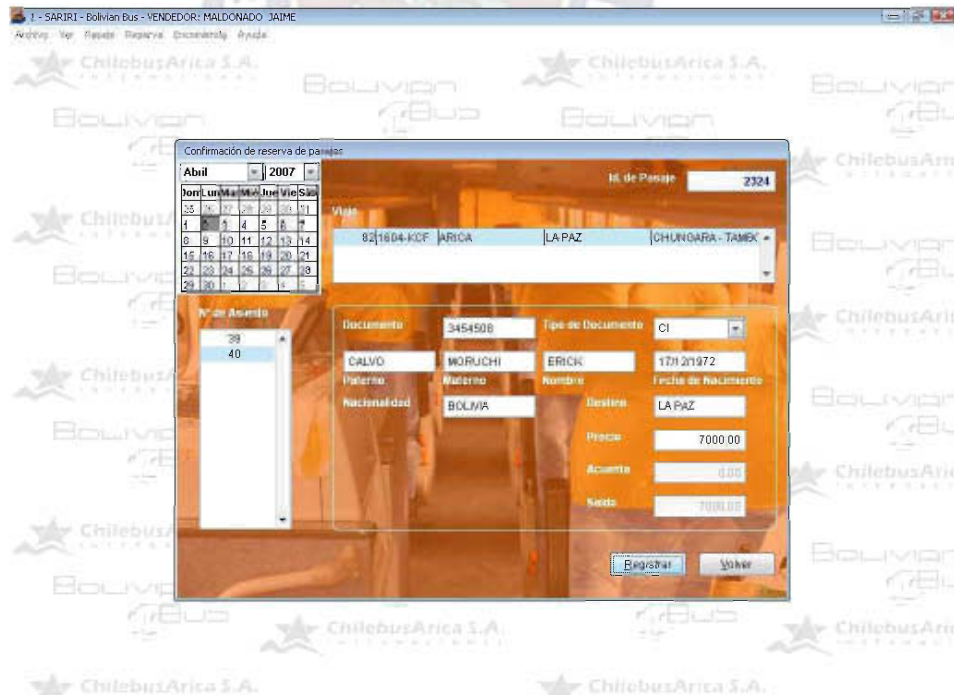
**Figura 3.24:** Aplicación WIN32 – Formulacion de canje de Pasajes  
**Fuente:** Elaboración Propia



**Figura 3.25:** Aplicación WIN32 – Formulario de registro de Reservas  
**Fuente:** Elaboración Propia

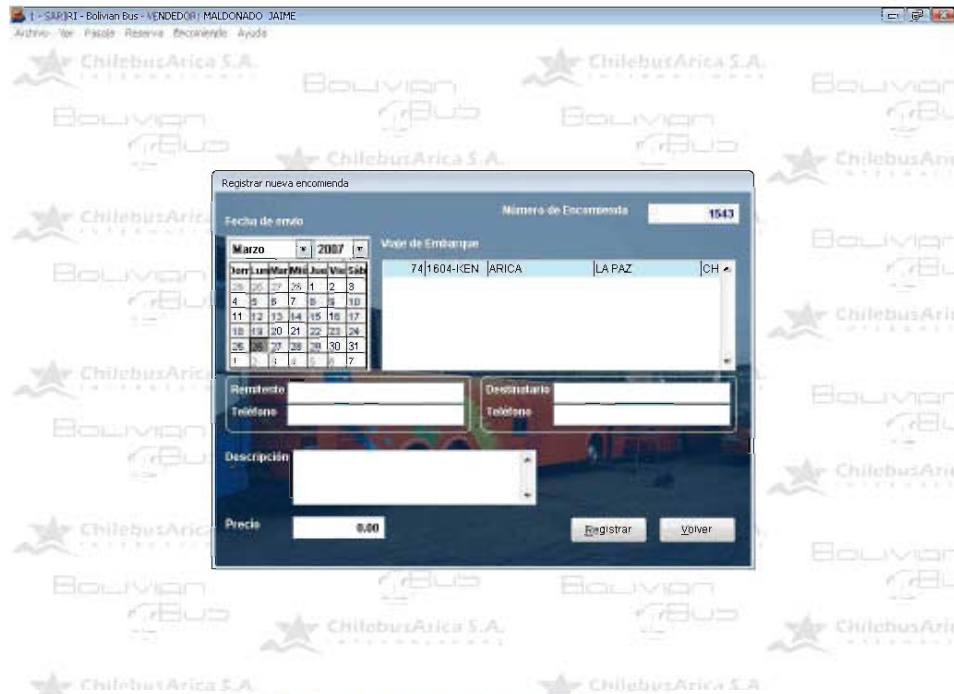


**Figura 3.26:** Aplicación WIN32 – Formulario de anulación de Reservas  
**Fuente:** Elaboración Propia

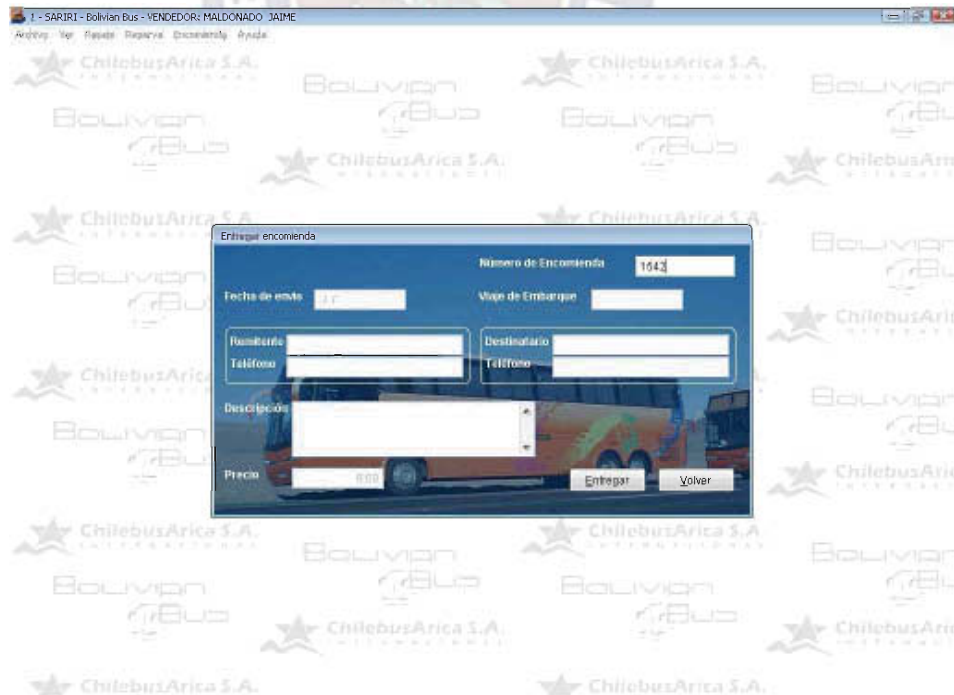


**Figura 3.27:** Aplicación WIN32 – Formulario de confirmación de Reservas  
**Fuente:** Elaboración Propia



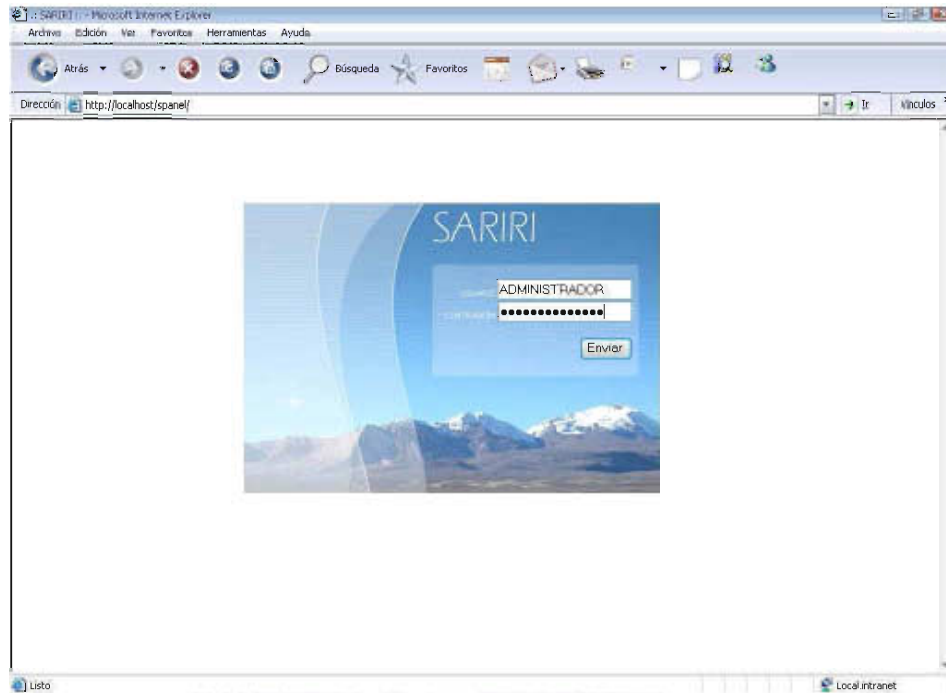


**Figura 3.28:** Aplicación WIN32 – Formulario de registro de Encomiendas  
**Fuente:** Elaboración Propia

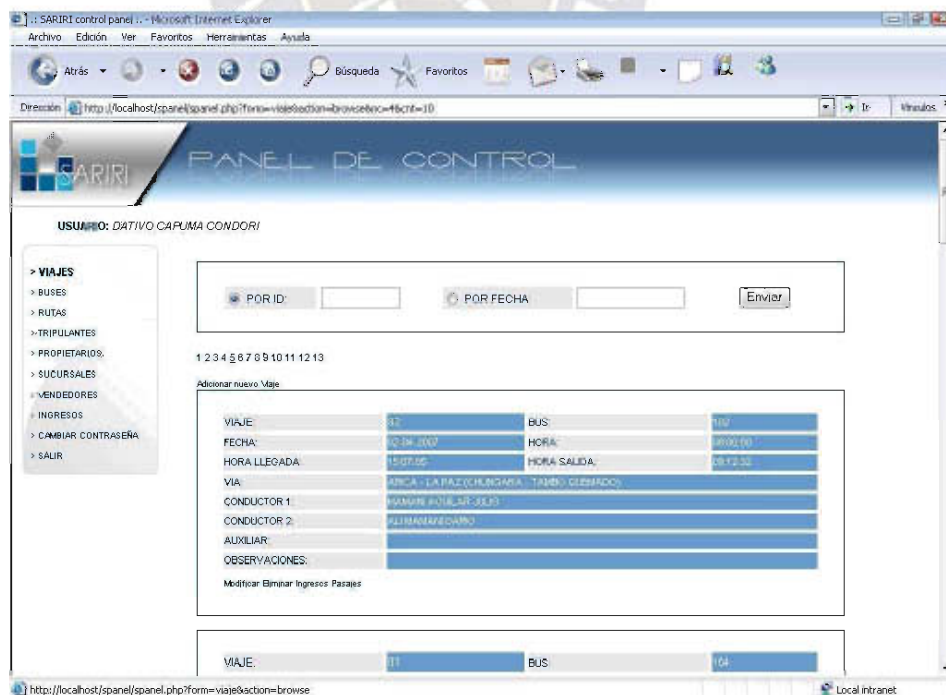


**Figura 3.29:** Aplicación WIN32 – Formulario de entrega de Encomiendas  
**Fuente:** Elaboración Propia

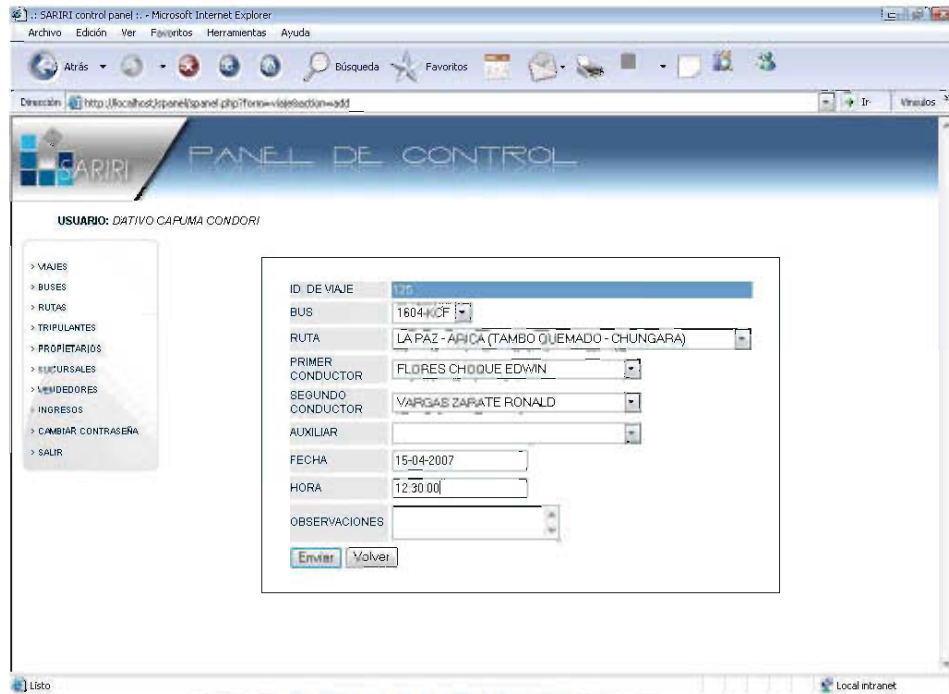
## APLICACION WEB



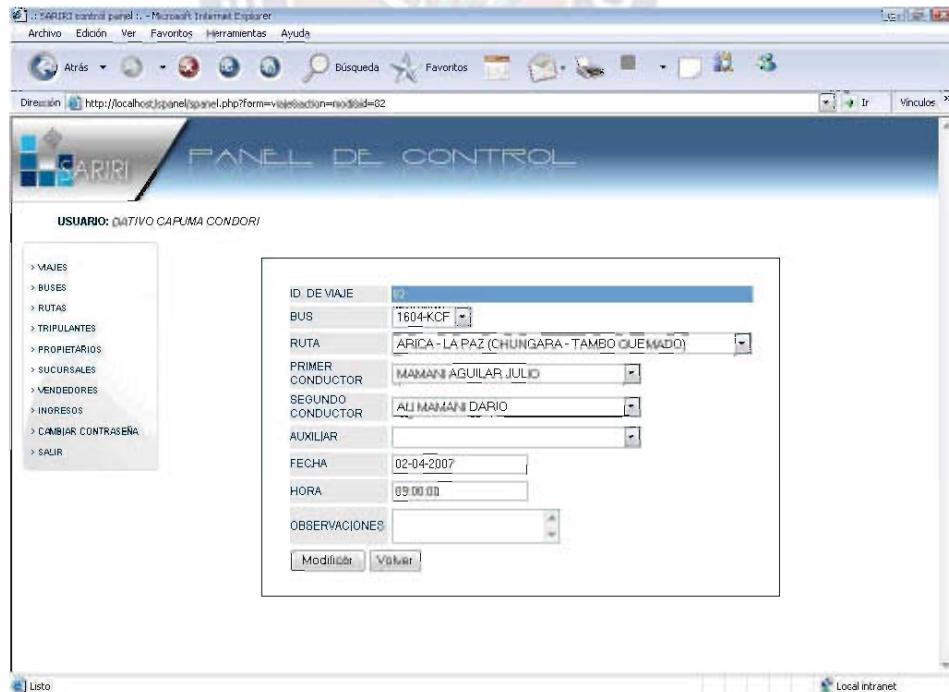
**Figura 3.30:** Aplicación Web – Autenticación de Usuario  
Fuente: Elaboración Propia



**Figura 3.31:** Aplicación Web – Listado de Viajes  
Fuente: Elaboración Propia



**Figura 3.32:** Aplicación Web – Formulario de registro de Viajes  
**Fuente:** Elaboración Propia



**Figura 3.33:** Aplicación Web – Formulario de modificación de Viajes  
**Fuente:** Elaboración Propia

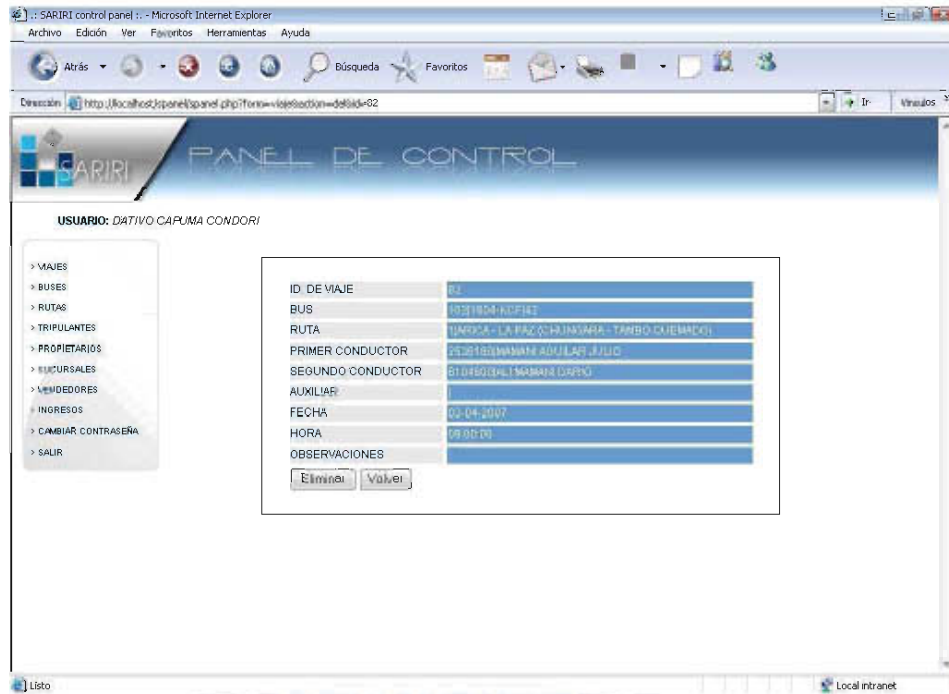


Figura 3.34: Aplicación Web – Formulario de eliminación de Viajes  
Fuente: Elaboración Propia

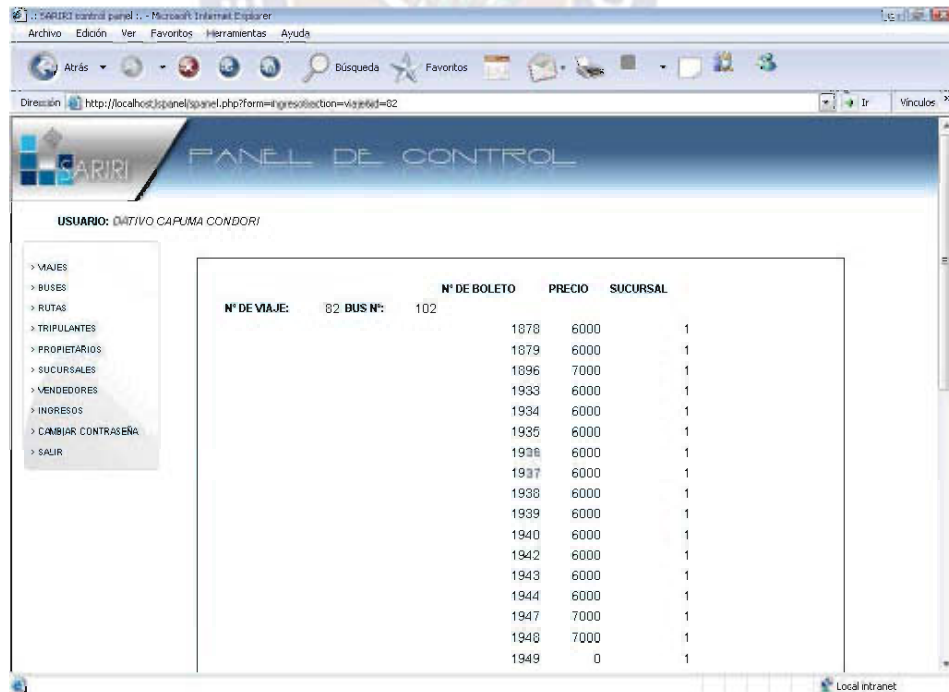


Figura 3.35: Aplicación Web – Descripción de Ingresos por viaje  
Fuente: Elaboración Propia

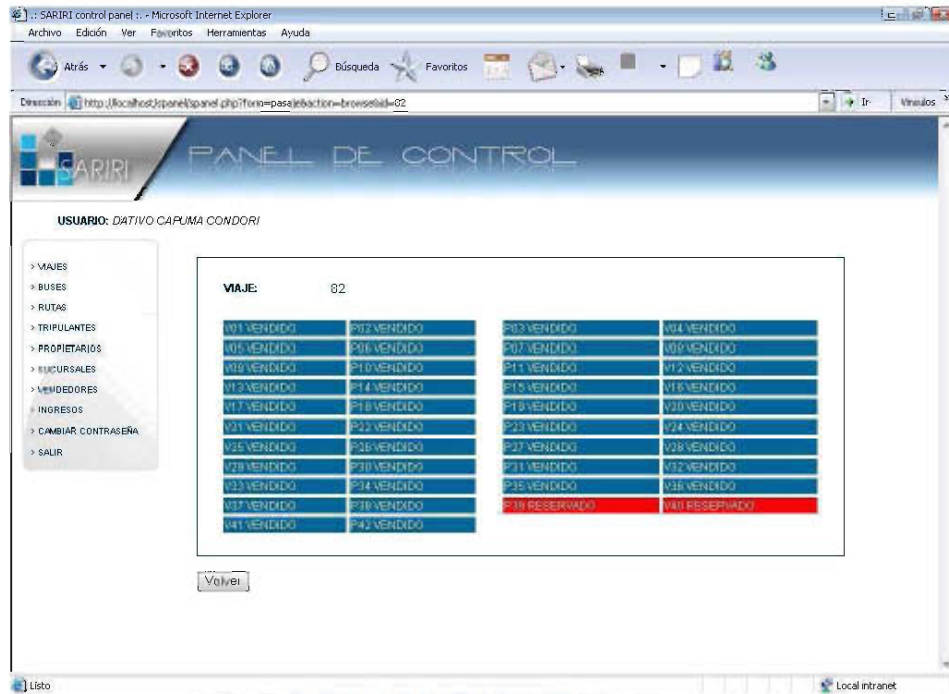


Figura 3.36: Aplicación Web – Vista de asientos por Viajes  
Fuente: Elaboración Propia

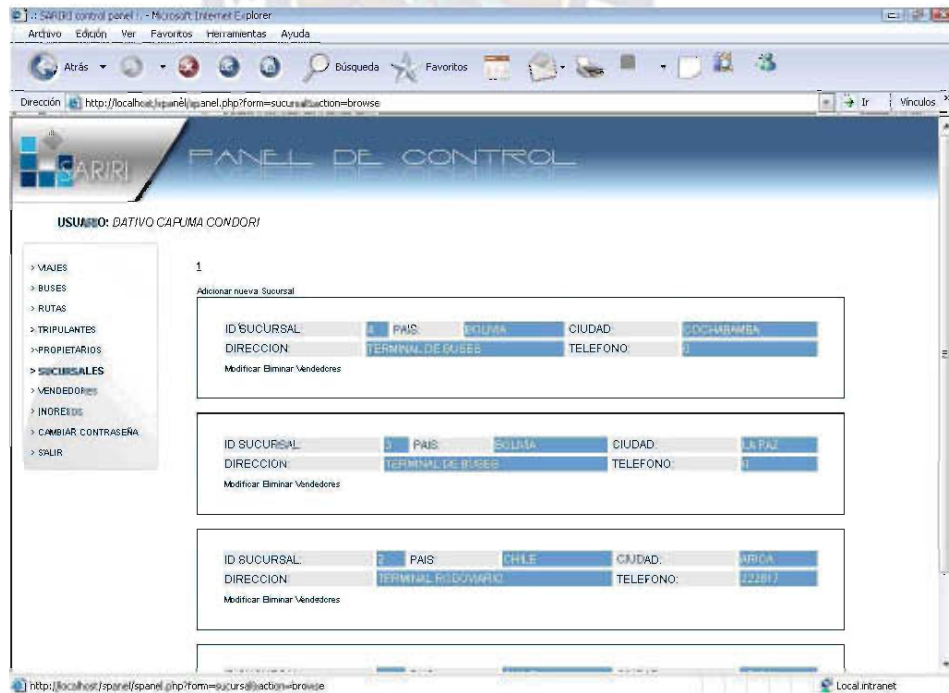
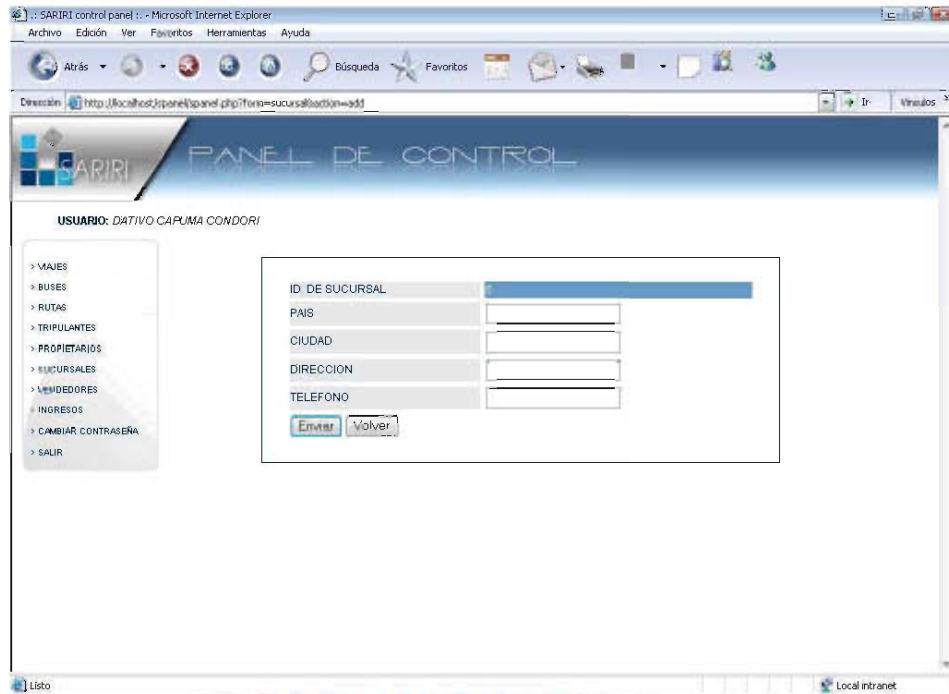
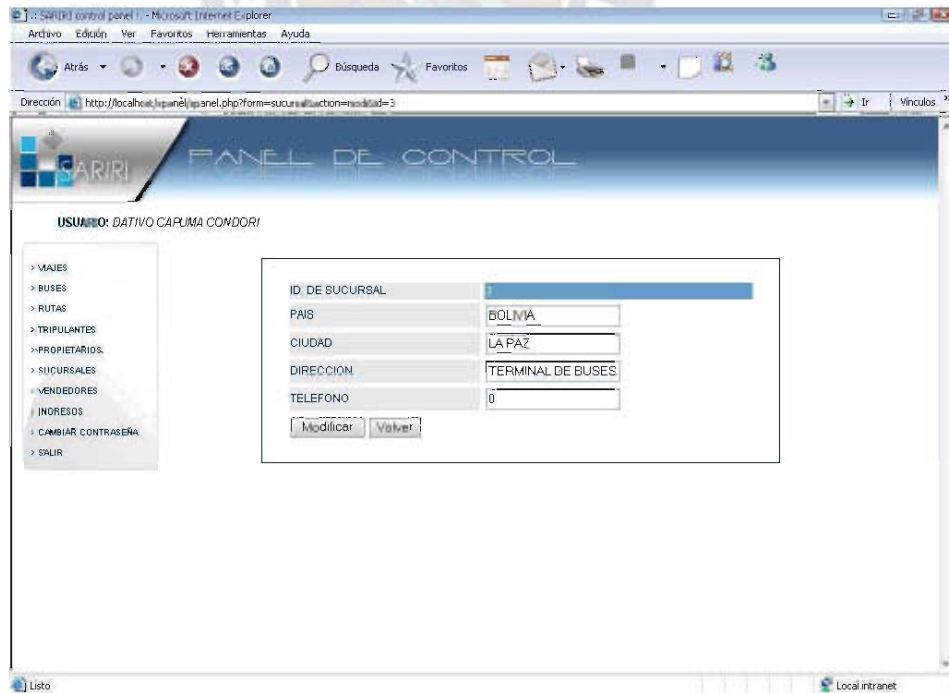


Figura 3.37: Aplicación Web – Listado de Sucursales  
Fuente: Elaboración Propia



**Figura 3.38:** Aplicación Web – Formulario de registro de Sucursales  
**Fuente:** Elaboración Propia



**Figura 3.39:** Aplicación Web – Formulario de modificación de Sucursales  
**Fuente:** Elaboración Propia

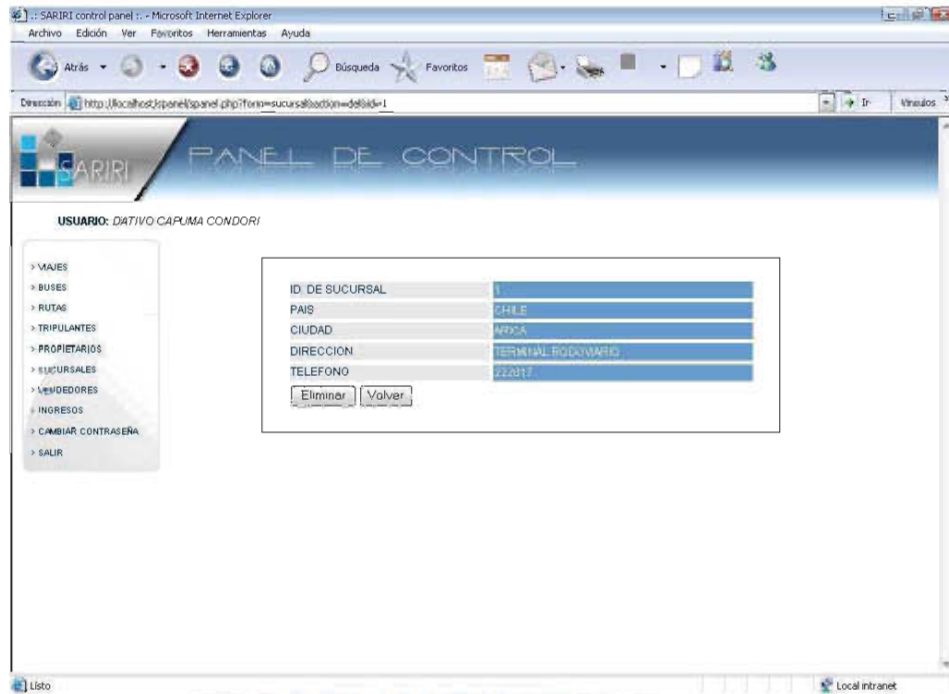


Figura 3.40: Aplicación Web – Formulario de eliminación de Sucursales  
Fuente: Elaboración Propia

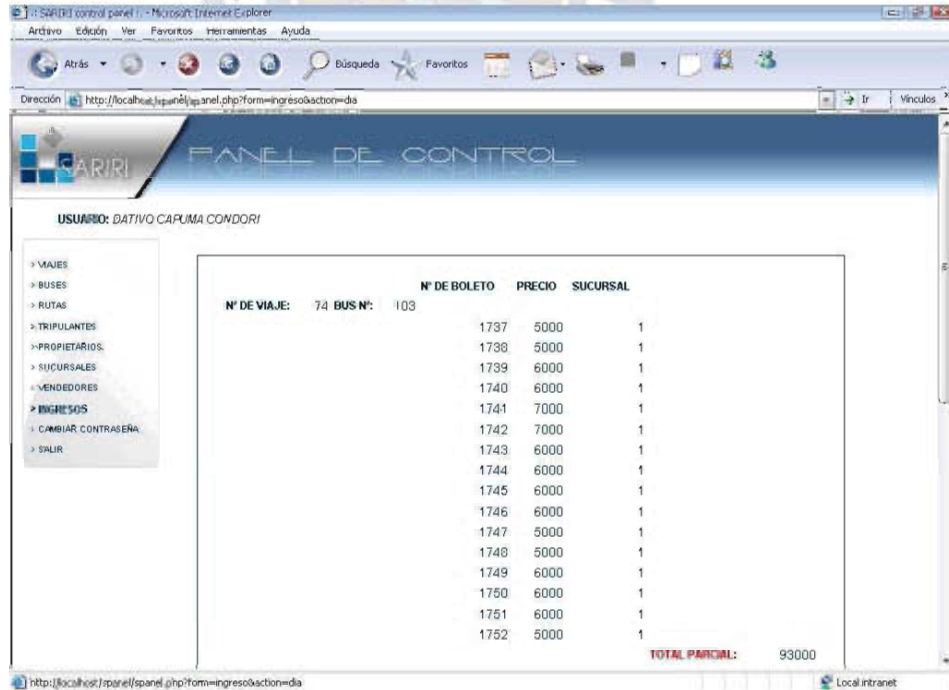


Figura 3.41: Aplicación Web – Descripción de Ingresos del día  
Fuente: Elaboración Propia

### 3.8 MEDIDAS DE CALIDAD

Los puntos utilizados para determinar la calidad de producto de software se encuentran enmarcados en el ISO/IEC 9126, lamentablemente este estándar no es verificable cuantitativamente.

Estas características están determinadas en función a entrevistas directas a los usuarios (vendedores y administradores), pruebas aplicadas y uso del producto de software.

Para dar un sentido más significativo a este apartado se cuantifica cada aspecto en función a un cuestionario de evaluación de software preparado (ver Anexo D) para este efecto, el mismo presenta una parte para usuarios y otra para desarrolladores.

#### 3.8.1 FUNCIONALIDAD

*FUNCIONALIDAD = 0.938 (ver Anexo D, tabla D.1)*

#### ADECUACION

El software desarrollado satisface las necesidades expresadas por los usuarios del mismo (vendedores y administradores), como son la administración de la venta de pasajes, y el manejo de la información sobre encomiendas, la administración de los viajes, control de ingresos entre otros, así mismo para cumplir con éstas necesidades se tuvo que realizar algunas otras acciones como es el diseño de la base de datos para almacenar toda la información que es necesaria para la empresa, una comunicación a través de Internet para una mejor comunicación entre sucursales y el administrador, etc.



## **EXACTITUD**

Toda la información que tiene que ver con la venta de pasajes y envío de encomiendas es registrada en una base de datos remota, gracias a esto el administrador cuenta con información, en tiempo real con un gran nivel de confianza, sobre el estado de los viajes (numero de pasajes vendidos, asientos reservados), ingresos por viaje, ingresos por día de todas las sucursales en las cuales el software esta funcionando. Además también se tienen información confiable sobre el estado de las encomiendas (si ya fue entregado en destino o no).

## **INTEROPERABILIDAD**

El software para sucursales de venta de pasajes esta desarrollado en una plataforma Windows con compatibilidad de sistema operativo Windows 2000 o superior, utiliza un Driver ODBC (MyODBC 3.51) para comunicarse con el servidor de Base de Datos (MySQL), esto debido a que el sistema operativo mencionado es el más conocido por usuario (vendedores).

La aplicación para administradores esta desarrollado en un entorno Web (HTML, Javascript, CSS, PHP) compatible con sistemas operativos Windows y Unix, montado sobre un servidor Apache y se conecta con un servidor de Base de Datos MySQL.

El servidor de Base de Datos es compatible con sistema operativos Windows y Unix.

## **CONFORMIDAD**

La software trata de seguir las especificaciones de Calida de Software ISO/IEC 9126. La aplicación Web cumple con las recomendaciones del W3C (World

Wide Web Consortium) tanto con el lenguaje de marcas como con las hojas de estilos y los enlaces.

## SEGURIDAD DE ACCESO

El software para sucursales de venta de pasajes, se identifica antes de acceder a la base de datos mediante el número MAC de la tarjeta de red del equipo residente, por si fuera poco, también solicita un identificar de usuario y una contraseña para poder iniciarse.

La configuración de la conexión remota se guarda en un archivo, toda esta información esta protegida mediante un código único de 32 caracteres, generados por un algoritmo de encriptación (MD5).

La aplicación para el administrador, utiliza autenticación del usuario mediante identificador y contraseña, tiene implementado una comprobación de sesión de usuario para restringir el acceso de personas no autorizadas.

Dentro la base de datos, todos los datos utilizados para el acceso de usuario se encuentran encriptados en cadenas únicas de 32 caracteres mediante el algoritmo MD5.

### 3.8.2 FIABILIDAD

*FIABILIDAD = 0.37 (ver Anexo D, tabla D.2)*

### NIVEL DE MADUREZ

Durante el primer mes de implementación es software presento 18 errores de pequeña intensidad, solucionados de manera fácil y rápida, durante el segundo mes presentó 3 errores uno grave de consulta a la base de datos y dos de

mucha menor intensidad, los mismos solucionados a la fecha, a partir del tercer mes no se presentó ningún problema o queja con respecto al funcionamiento.

## **TOLERANCIA A FALLAS**

El software desarrollado presenta funcionalidad parcial cuando se produce un error de funcionamiento, siempre y cuando el error no se presente con el servidor de base de datos (en el caso de la aplicación para sucursales como de la aplicación para administradores), con el Driver ODBC (para la aplicación para sucursales), con el servidor web (para la aplicación web).

## **RECUPERABILIDAD**

Para prevenir pérdida de información en caso de falla eléctrica, falla del equipo, pérdida de la conexión, etc. toda acción de registro, modificación y eliminación de información sobre pasajes y encomiendas se la realiza directamente en la base de datos.

### **3.8.3 USABILIDAD**

*USABILIDAD = 0.956 (ver Anexo D, tabla D.3)*

## **COMPENSIBILIDAD**

El software desarrollado presenta una interfaz en base a formularios sencillos y ordenados para hacer que los usuarios puedan ubicarse dentro de los mismos, se encuentran diseñados de forma que el ingreso de la información sigue una secuencia hasta completar su tarea sea esta la de registro, modificación o eliminación de datos, reacciona frente acciones no permitidas emitiendo mensajes de advertencia breves y de lenguaje sencillo.

## FACILIDAD DE APRENDER

El software tiene un entorno familiar para los usuarios con un menú en el orden esperado: Archivo, Ver, etc., en la parte superior derecha tiene los típicos botones para Minimizar, Maximizar, Restaurar y Cerrar (para la aplicación), se hace más sencillo el uso con el soporte del ratón, además tiene implementados rutinas de auto completado de formularios, el sistema esta desarrollado de forma que el usuario tenga que hacer la menor cantidad posible de maniobras con el ratón y utilizar el teclado únicamente para lo esencial (ingreso de información relevante).

## OPERABILIDAD

Los usuarios pueden configurar de manera sencilla la configuración a la base de datos mediante un formulario sencillo.

El software también contiene soporte de accesos directos a formularios mediante teclas abreviadas (para usuarios experimentados).

## ATRACCION

El software esta diseñado en un entorno grafico (utiliza una interfaz grafica de usuario - GUI), combina colores y fondos para hacer más agradable su utilización y estética su presentación.

### 3.8.4 EFICIENCIA

*EFICIENCIA = 0.943 (ver Anexo D, tabla D.4)*

## COMPORTAMIENTO CON RESPECTO AL TIEMPO

El software desarrollado utiliza el Lenguaje Estructurado de Consulta (SQL) para realizar solicitar información al servidor de base de datos, de esta forma la transferencia de información es rápida, además que solo se solicitan los datos necesarios para trabajar, de esta manera el intercambios de información entre el sistema y el servidor no contempla mucho tiempo.

## COMPORTAMIENTO CON RESPECTO A RECURSOS

El software desarrollado realiza las consultas necesarias a la base de datos para un mejor aprovechamiento de los recursos del servidor.

### 3.8.5 MANTENIBILIDAD

*MANTENIBILIDAD = 0.96 (ver Anexo D, tabla D.5)*

### ANALIZABILIDAD

Frente a una falla de código, tanto la aplicación para las sucursales como para el administrador ofrecen mensajes de ubicación de los errores, por un lado, la aplicación para sucursales tiene implementado una rutina de manejo de errores, mostrando un mensaje con la descripción del error, el objeto que lo produce, y la línea de código en la cual se encuentra el error; por otro lado, en la aplicación web el interprete de php se encarga de comunicarnos, si se produce un error, el tipo de error, el archivo en el cual se encontró y la línea en la cual se encuentra. De este modo se pueden identificar la ubicación de los errores para una posterior modificación y rectificación de los mismos.

## **CAMBIABILIDAD**

Gracias a la estructura orientada a objetos del lenguaje de desarrollo del software (Visual FoxPro) la modificación del código fuente del software es sencilla e intuitiva. Además todas las modificaciones del código fuente se adaptan fácilmente al entorno del sistema, siempre y cuando se respeten los protocolos de comunicación con el servidor de base de datos y la estructura de la base de datos.

### **3.8.6 PORTABILIDAD**

*PORTABILIDAD = 0.944 (ver Anexo D, tabla D.6)*

## **FACILIDAD DE INSTALACION**

La instalación de la aplicación para sucursales es fácil y sencilla, utilizando un asistente de instalación que nos guía durante toda la instalación. El instalador contiene tanto la aplicación como el driver ODBC necesario.

La aplicación web se encuentra alojada en un servidor web el cual puede ser accedido desde cualquier parte del mundo.

## **COEXISTENCIA**

Gracias a que la base de datos se encuentra separa del software, el sistema puede compartir este recurso con otras aplicaciones.

## **REEMPLAZABILIDAD**

El software puede ser reemplazado fácilmente por otro sistema que sea compatible con la base de datos.

## Capítulo IV: Conclusiones y Recomendaciones

### 4.1 CONCLUSIONES

Con el uso metodologías y herramientas de análisis, diseño y desarrollo de sistemas de información computarizados, se logró implementar un sistema que satisface las necesidades de los usuarios finales, tanto administradores como vendedores.

La implementación del Sistema de Control y Administración "SARIRI", en la empresa de transporte internacional de pasajeros "Bolivian Bus", permite mejorar ampliamente el desarrollo de las actividades haciéndolas más productivas, eliminando la duplicación de tareas realizadas, efectuando un control riguroso de la información referente a ingresos, emitiendo informes periódicos con información confiable, segura y oportuna, además se mejora cualitativamente la atención a los clientes de la empresa con un mejor servicio.

El Sistema de Control y Administración "SARIRI", puede ser utilizado por otras empresas de transporte que realizan la misma actividad, permitiendo convertirlas en empresas modernas y competitivas.

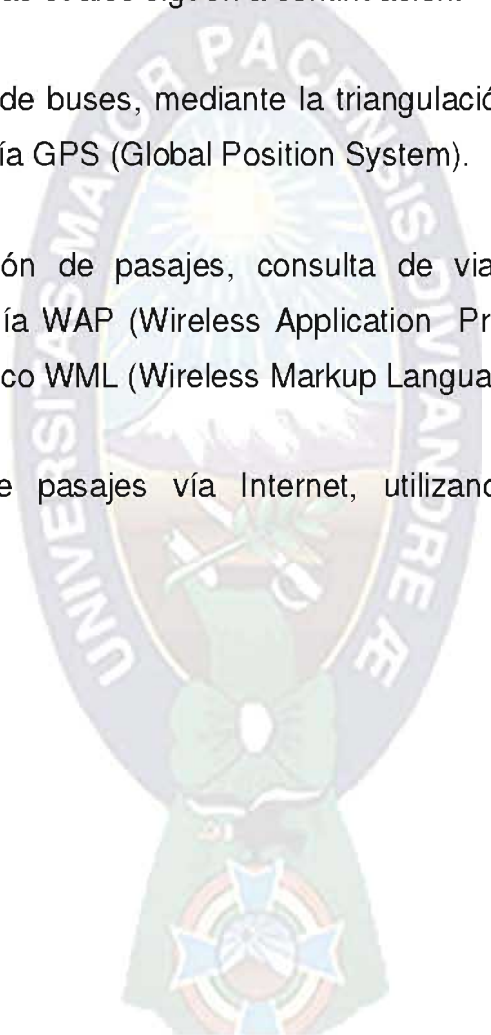
## 4.2 RECOMENDACIONES

Conforme con el avance de la tecnología es evidente la evolución de los sistemas de información computarizados, por ello se plantean algunas sugerencias para una extensión del Sistema de Control y Administración "SARIRI" implementado en la empresa de transporte internacional de pasajeros "BOLIVIAN BUS", las cuales siguen a continuación:

Mejorar el control de buses, mediante la triangulación satelital de condenadas utilizando tecnología GPS (Global Position System).

Realizar reservación de pasajes, consulta de viajes, asientos y destinos, utilizando tecnología WAP (Wireless Application Protocol) con el lenguaje de etiquetas inalámbrico WML (Wireless Markup Language).

Realizar venta de pasajes vía Internet, utilizando e-Comerce (Comercio Electrónico).





## REFERENCIA BIBLIOGRAFICA

[YOURDON93] YOURDON E., 1993. Análisis Estructurado Moderno. Volumen I, 735 págs. 1ª Edición, Prentice-Hall Hispanoamérica, México.

[NAVARRETE96] NAVARRETE A., 1996. Una metodología relacional hipermedia – Estudio en casos Prácticos, 246 págs. Proyecto final de carrera. Palma de Mallorca, España.

[ROJAS2000] ROJAS G., 2000. DRM: Modelamiento Conceptual para Sistemas Hipermediales. III INTERNATIONAL SCHOOL ON INTERNET IMPACT IN SOFTWARE ENGINEERING, 4 págs. Viña del Mar, Chile.

[WIKI07] LA ENCICLOPEDIA LIBRE, WIKIPEDIA. Disponible en: <http://www.wikipedia.es>.

[KOBAYASHI] KOBAYASHI Gutiérrez A., 2007. Aseguramiento de la Calidad de Aplicaciones Web. IV COREIS NORTE UPN 2007. Perú.

[CUEVAS99] CUEVA Lovelle J., 1999. Calidad del Software. Conferencia, 21 de Octubre de 1999. Grupo GIDIS. Universidad Nacional de la Pampa.

[ISO9126] ISO/IEC 9126:1991 Information Technology – Software Product Evaluation – Quality Characteristics and Guidelines for their use, Secretaría General de ISO, Ginebra, 1991.

[ISO/IEC9126-1] ISO/IEC 9126-1:2001 Software Engineering – Product quality. Part 1: Quality Model, Secretaría General de ISO, Ginebra, 2001.

[ISO/IEC9126-2] ISO/IEC 9126-2:2003 Software Engineering – Product quality. Part 2: External Metrics, Secretaría General de ISO, Ginebra, 2003.

[ISO/IEC9126-3] ISO/IEC 9126-3:2003 Software Engineering – Product quality. Part 3: Internal Metrics, Secretaría General de ISO, Ginebra, 2003.

[PHP01] Stig Sæther B. y cols, 2001. Manual de PHP. Editado por Rafael Martínez, 1062 págs. Grupo de documentación de PHP.

[MATEOS97] MATEOS Suarez J., 1997. MANUAL BASICO PARA LA CREACION DE PAGINAS HTML. Versión 1.0, 63 págs. Universidad de Guadalajara, México.

[BUSTAMANTE98] BUSTAMANTE A. Desarrollo de Aplicaciones en: VISUAL FoxPro. 1ra. Edición, 520 págs. GrapPerú, Perú.

## **ENLACES RELACIONADOS**

Guía de Referencia XHTML

<http://www.w3c.es/Divulgacion/GuiasReferencia/XHTML1/>

Validación de Lenguaje de marcas <http://validator.w3.org>

Página de la Calidad de la W3C <http://www.w3.org/QA/Tools/>

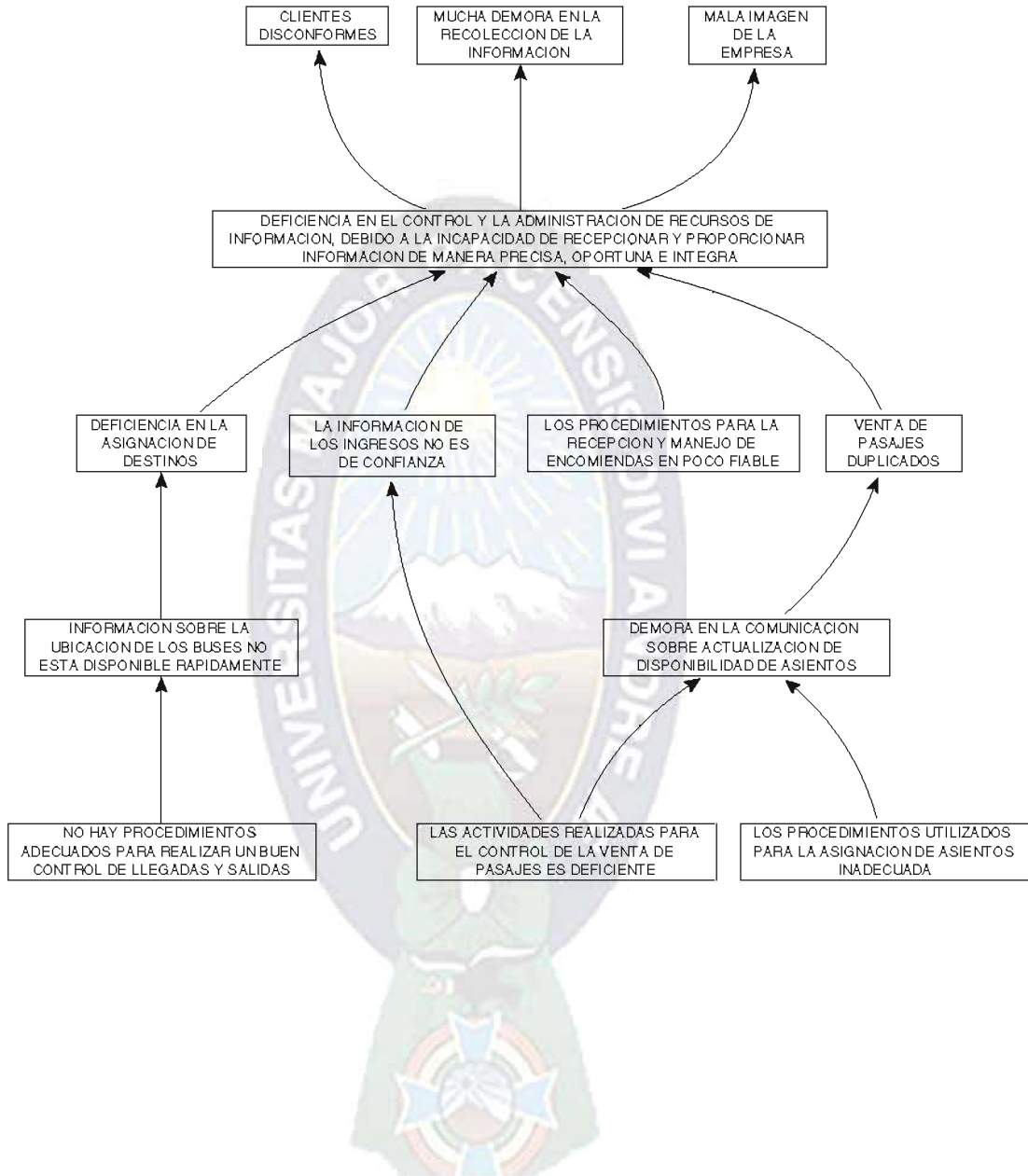
Validación de Hojas de Estilo <http://jigsaw.w3.org/css-validator/>

Validación de Enlaces <http://validator.w3.org/checklink>

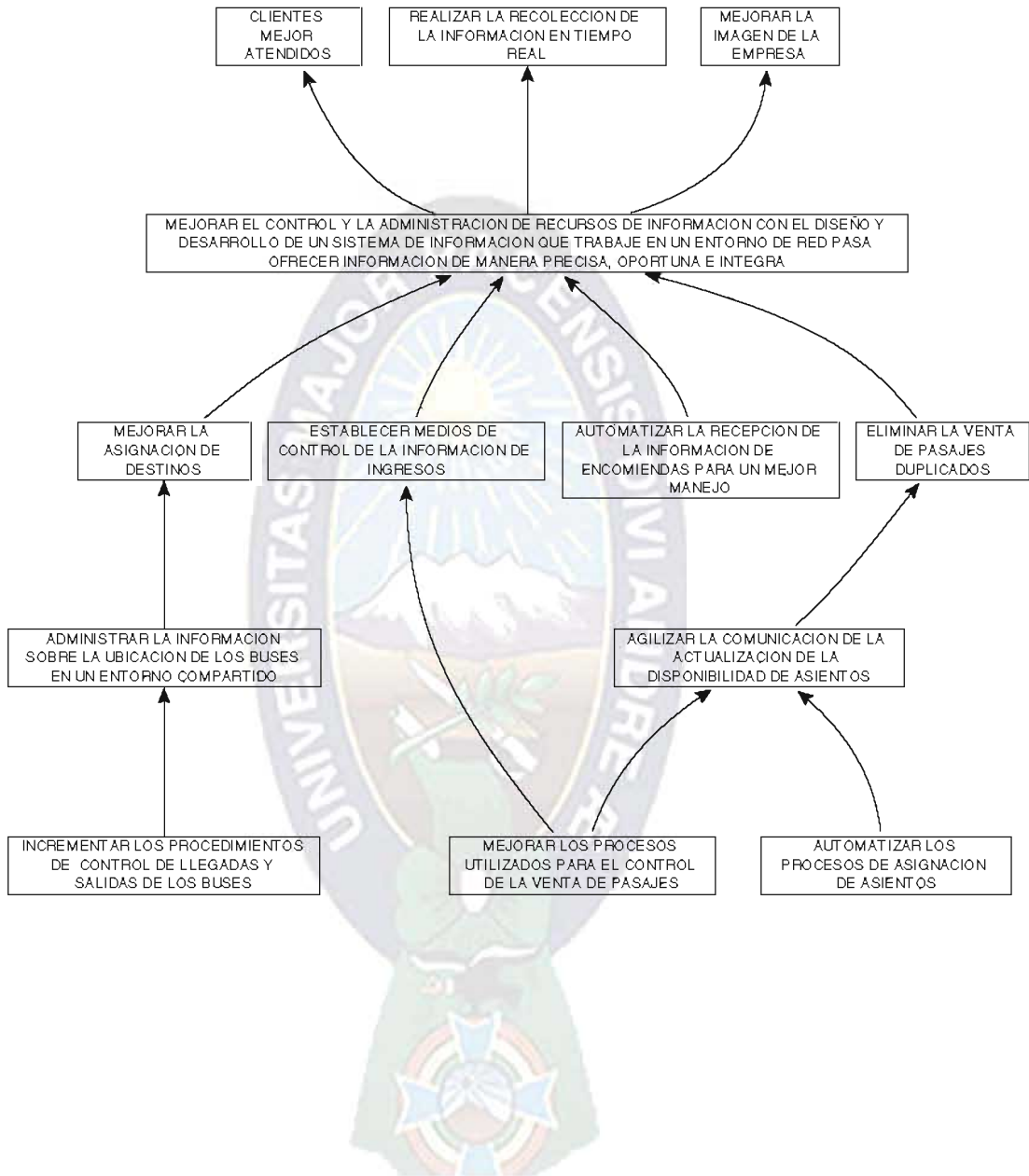
# AneXOS



## Anexo A: Árbol de Problemas



## Anexo B: Árbol de Objetivos



### Anexo C: Matriz del Marco Lógico

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<p><b>Fin:</b> Contribuir a conseguir la confianza total de los datos para mejorar el desarrollo de las actividades de la empresa y procurar una mejor situación financiera de la misma.</p>	<p>Mejorar el control y la administración eliminando en un 100% las tareas duplicadas que son susceptibles a producir errores.</p>	<p>Informes que realiza el administrador mensualmente para los propietarios.</p>	<p>La empresa de transporte "BOLIVIAN BUS" deberá utilizar el sistema y cambiar algunas políticas de administración y control.</p>
<p><b>Propósito:</b> Mejorar el control y la administración de recursos de la Empresa de Transporte Internacional de Pasajeros "BOLIVIAN BUS" con el diseño y desarrollo de un sistema de información que trabaje bajo un entorno de red, utilizando herramientas de análisis y diseño estructurado de sistemas.</p>	<p>Eliminar el 95% de procesos manuales en el manejo de la información para el control y la administración de las actividades de la empresa, minimizando de esta forma el tiempo de estos procesos.</p>	<p>Reportes sobre venta de pasajes realizados diariamente por el administrador.</p> <p>Guía de encomiendas realizada por el personal de ventas de cada sucursal para cada viaje.</p> <p>Lista de pasajeros realizada para cada viaje por el personal de ventas de cada sucursal, en una cantidad de 12 copias por salida.</p> <p>Manifiesto de encomiendas realizado por el personal de ventas de cada sucursal para cada salida.</p>	<p>Se debe contar con ambientes propicios para la implementación del sistema.</p> <p>El sistema deberá ser instalado en todas las sucursales de venta de pasajes y utilizados por todo el personal de ventas.</p> <p>El administrador utilizará el sistema como parte de sus actividades administrativas y de control.</p>
<p><b>Componentes:</b></p> <ul style="list-style-type: none"> <li>• Mejorar las operaciones de venta y control de pasajes.</li> <li>• Mejorar y facilitar el manejo de la información concerniente a las encomiendas.</li> <li>• Mejorar las operaciones de asignación de asientos.</li> </ul>	<p>El sistema funcionará al 100% en abril del 2007.</p> <p>Se contará con un confianza del 99% sobre la integridad de la información.</p> <p>El 95% de la recolección de la información será en tiempo real.</p>	<p>Certificado de conformidad elaborado por el gerente general de la empresa a la implementación del sistema.</p> <p>Reportes mensuales realizados por el administrador basados en toda la información almacenada en la base de datos.</p>	<p>El personal de venta deberá aceptar la implantación del sistema.</p> <p>La información introducida al sistema deberá ser seria y confiable.</p> <p>El personal de ventas deberá tener conocimientos mínimos de ofimática.</p>

<ul style="list-style-type: none"> <li>• Facilitar la elaboración y comunicación de destinos de viajes.</li> <li>• Llevar un mejor control de los horarios de los buses para poder ofrecer un mejor servicio a los clientes.</li> <li>• Centralizar la información necesaria para la elaboración de reportes y listados de manera automatizada.</li> </ul>	<p>Las actividades administrativas serán un 95% de fáciles de realizar.</p> <p>Los resultados de las consultas realizadas a la base de datos tendrán un 98% de utilidad para la toma de decisiones.</p> <p>Los problemas sobre venta de pasajes duplicados no serán mayores al 0.01%.</p>	<p>El registro de la información realizado diariamente en cada sucursal de venta de pasajes.</p> <p>Reportes realizados diariamente por el personal de ventas de cada sucursal.</p>	
<p><b>Actividades:</b></p> <ol style="list-style-type: none"> <li>1. Investigación preliminar del sistema.</li> <li>2. Análisis del sistema.</li> <li>3. Diseño del sistema.</li> <li>4. Desarrollo del sistema.</li> <li>5. Prueba del sistema.</li> <li>6. Puesta en marcha del sistema.</li> <li>7. Capacitación al personal.</li> </ol>	<p><i>Actividad 1:</i> Tiempo: 3 meses Costo / Mes: 300 Usd.</p> <p><i>Actividad 2:</i> Tiempo: 2 meses Costo / Mes: 300 Usd.</p> <p><i>Actividades 3,4,5 y 6:</i> Tiempo: 3 meses Costo / Mes: 300 Usd.</p> <p><i>Actividad 7:</i> Tiempo: 1 mes Costo / Mes: 600 Usd.</p>	<p>Perfil de Proyecto de grado realizado durante los primeros tres meses de la realización del proyecto.</p> <p>Documentación del proyecto al cabo de los cinco meses de iniciado el proyecto.</p> <p>Código fuente del sistema, resultados de pruebas, capacitación del personal y la liberación del sistema concluido para su utilización formal.</p> <p>Manuales técnico y de usuario presentados a la culminación del sistema.</p>	<p>Tener acceso a bibliografía especializada para el análisis y diseño de aplicaciones.</p> <p>Se debe contar con todos los suministros económicos y tecnológicos necesarios.</p>
<p><b>COSTO TOTAL</b></p>	<p>3000 Usd.</p>		

## Anexo D: Modelo de Calidad

# CUESTIONARIO DE EVALUACION DEL SISTEMA

## **"SARIRI"**

Evaluar cada aspecto con un valor del 1 – 10.

### USUARIOS

El software implementado:

1. (.....) Satisface mis necesidades con sus funciones.
2. (.....) Ayuda a realizar tareas específicas con sus funciones.
3. (.....) Ofrece información correcta con un grado de precisión aceptable.
4. (.....) Es capaz de interactuar con otros sistemas específicos (servidores de base de datos, servidores web, etc.).
5. (.....) Protege la información de personas no autorizadas.
  
6. (.....) No incurre en errores cuando se ejecuta.
7. (.....) Cuando el software tiene algún error de funcionamiento, no se ve afectado todo el programa.
8. (.....) Es capaz de recuperar la información que se estuvo manejando antes que produjera un error.
  
9. (.....) Es entendible.
10. (.....) Puede ser utilizado fácilmente.
11. (.....) Es adecuado para mi trabajo.
12. (.....) Lo utilizo para determinadas tareas.
13. (.....) Puede ser aprendido con facilidad
14. (.....) Puede ser operado (cambiar la configuración).
15. (.....) Puede ser controlado (ej.: teclas abreviadas).
16. (.....) Es atractivo a la vista.
  
17. (.....) Tiene un rendimiento de acuerdo con los recursos que utiliza.
18. (.....) Funciona con rapidez cuando realiza sus funciones.
19. (.....) Utiliza solo cantidades y recursos necesarios (ej.: computadoras, impresoras, etc.).



## CUESTIONARIO DE EVALUACION DEL SISTEMA

### **"SARIRI"**

Evaluar cada aspecto con un valor del 1 – 10.

#### DESARROLLADORES

El software implementado:

1. (.....) Puede ser modificado.
2. (.....) Deja identificar las partes que deben ser modificadas.
3. (.....) Permite implementar una modificación especificada.
4. (.....) No presenta efectos inesperados después de las modificaciones.
5. (.....) Permite validar las partes modificadas.
6. (.....) Puede ser transferido de un entorno a otro.
7. (.....) Es capaz de ser adaptado a otros ambientes con facilidad (otras empresas similares).
8. (.....) Es fácil de instalar.
9. (.....) Es capaz de coexistir con otro software y compartir recursos.
10. (.....) Es capaz de reemplazar a otro software similar.

## RESULTADOS DE CUESTIONARIO

PREGUNTA	EMP1	EMP2	EMP3	EMP4	EMP5	EMP6	EMP7	EMP8	EMP9	EMP10	
<b>FUNCIONALIDAD</b>											
1	10	9	9	10	10	9	10	9	8	9	9,3
2	10	10	10	10	9	10	10	9	9	10	9,7
3	8	9	9	10	9	8	10	10	10	8	9,1
4	9	9	10	10	10	9	8	10	7	9	9,1
5	10	10	10	9	10	10	9	10	10	9	9,7
<b>PROMEDIO</b>											9,38

**Tabla D.1:** Resultados del cuestionario para la funcionalidad  
Fuente: Elaboración Propia

PREGUNTA	EMP1	EMP2	EMP3	EMP4	EMP5	EMP6	EMP7	EMP8	EMP9	EMP10	
<b>FIABILIDAD</b>											
6	9	10	10	8	10	10	9	10	10	8	9,4
7	10	9	10	10	10	9	8	10	10	9	9,5
8	10	10	10	9	9	8	8	9	9	10	9,2
<b>PROMEDIO</b>											9,37

**Tabla D.2:** Resultados del cuestionario para la fiabilidad  
Fuente: Elaboración Propia

PREGUNTA	EMP1	EMP2	EMP3	EMP4	EMP5	EMP6	EMP7	EMP8	EMP9	EMP10	
<b>USABILIDAD</b>											
9	9	9	10	10	9	10	8	10	10	9	9,4
10	10	10	10	10	9	8	10	9	9	10	9,5
11	10	10	10	10	10	10	10	10	10	10	10
12	10	10	10	10	10	10	10	10	10	10	10
13	9	9	10	9	9	9	10	10	10	10	9,5
14	10	9	9	9	10	10	8	8	9	9	9,1
15	8	9	9	8	9	10	10	9	10	8	9
16	10	10	10	10	10	10	10	10	10	10	10
<b>PROMEDIO</b>											9,56

**Tabla D.3:** Resultados del cuestionario para la usabilidad  
Fuente: Elaboración Propia

PREGUNTA	EMP1	EMP2	EMP3	EMP4	EMP5	EMP6	EMP7	EMP8	EMP9	EMP10	
<b>EFICIENCIA</b>											
17	8	10	8	9	10	10	8	9	10	9	9,1
18	9	9	9	10	10	10	10	9	10	10	9,6
19	9	10	10	10	8	9	10	10	10	10	9,6
<b>PROMEDIO</b>											9,43

**Tabla D.4:** Resultados del cuestionario para la eficiencia  
Fuente: Elaboración Propia

PREGUNTA	DES1	DES2	DES3	DES4	DES5	
<b>MANTENIBILIDAD</b>						
1	10	9	10	10	10	9,8
2	10	9	10	10	10	9,8
3	10	10	10	10	9	9,8
4	8	9	10	10	9	9,2
5	9	9	10	9	10	9,4
<b>PROMEDIO</b>						9,6

**Tabla D.5:** Resultados del cuestionario para la mantenibilidad  
**Fuente:** Elaboración Propia

PREGUNTA	DES1	DES2	DES3	DES4	DES5	
<b>PORTABILIDAD</b>						
6	10	9	10	8	8	9
7	10	10	10	9	10	9,8
8	9	9	10	10	10	9,6
9	9	9	9	10	10	9,4
10	10	10	9	10	8	9,4
<b>PROMEDIO</b>						9,44

**Tabla D.6:** Resultados del cuestionario para la portabilidad  
**Fuente:** Elaboración Propia

