

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN TECNOLÓGICA

SISTEMA DE ADMINISTRACIÓN DE VENTAS DE PAQUETES TURÍSTICOS

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ANÁLISIS DE SISTEMAS INFORMÁTICOS**

**FLORES LAGUATE CONSUELO GRIMANESA
QUISHPE VILLALBA GLADYS ENCARNACIÓN**

DIRECTOR: ING. ROSA NAVARRETE MSc.

Quito, Octubre de 2006

CAPITULO I

INTRODUCCIÓN

1.1 AMBITO

La empresa "HERUMA Ltda. "es una operadora de turismo que vende paquetes turísticos a empresas organizaciones, grupos y personas naturales a nivel nacional, proporcionando además servicio de transporte, alimentación, alojamiento y guía turístico. Está ubicada en la calle Colón y Diego de Almagro, Centro Comercial Alcaza en la ciudad de Quito.

1.2 PROBLEMA

Dentro del ámbito de su operación, se han detectado entre otros los siguientes problemas:

- ◆ Dificultad en el proceso de administración de la venta de paquetes turísticos.
- ◆ Deficiencia en el control de ventas y administración de información.
- ◆ Lentitud en la atención al cliente.
- ◆ Falta de información para la toma de decisiones oportunas.

Se propone desarrollar un Sistema Informático que permita la gestión de todos los datos de clientes, vendedores, guías turísticos y paquetes turísticos; según requerimiento de los interesados y proporcione información a los directivos para apoyo a la toma de decisiones.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Apoyar las actividades de administración de paquetes turísticos de la empresa "HERUMA LTDA.", mediante el desarrollo y posterior implementación de un sistema informático.

1.3.2 OBJETIVOS ESPECÍFICOS

Los objetivos que se plantean, para el Sistema de administración de paquetes turísticos, en acuerdo con nuestros clientes, son los siguientes:

- ◆ Administrar datos de clientes, vendedores, paquetes turísticos y guías turísticos.
- ◆ Calcular comisiones, tanto de vendedores como guías turísticos, de acuerdo a las ventas que hayan realizado en caso de vendedores; y los tours que hayan dirigido, al tratarse de guías; siempre aplicando sus respectivas restricciones.
- ◆ Administrar reservaciones de paquetes turísticos que se efectúen en forma directa con la empresa, así como las ventas realizadas por los vendedores de la empresa que las realicen mediante visitas a sus clientes.
- ◆ Generar estadísticas e informes apoyándonos en datos reales que estén siendo manejados por el sistema.
- ◆ Registrar la asistencia de vendedores y guías turísticos.

1.4 ALCANCE Y LIMITACIONES

1.4.1 ALCANCE

- ◆ Respaldo de los datos de los clientes, vendedores, guías turísticos, paquetes turísticos y ventas.
- ◆ Controlar la asistencia de vendedores y guías turísticos
- ◆ Generar diferentes reportes de acuerdo a las necesidades de los directivos de la empresa.
- ◆ Facilitar el cálculo de comisiones a vendedores y guías turísticos.
- ◆ Llevar un control de las ventas de paquetes turísticos.
- ◆ Brindar un mejor servicio a los clientes al realizar las ventas de paquetes turísticos.

1.4.2 LIMITACIONES

- ◆ No calcula rol de pagos de vendedores y guías turísticos.
- ◆ No incluye interfaz con la contabilidad.

CAPITULO II

MARCO TEÓRICO

2.1 INGENIERÍA DE SOFTWARE

La Ingeniería de software es la disciplina que orienta al desarrollo de software de calidad.

El proceso de ingeniería de software se define como un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, es decir, la obtención de un producto de software de calidad.

Las necesidades de usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo.

El Proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición.

(<http://www.austral.edu.ar/web/ingenieria/design/transparencias/7>)

2.1.1 CALIDAD DEL SOFTWARE

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y su existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

La calidad de software es medible y varia de un sistema a otro o de un programa a otro. La calidad de software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas en el diseño, por lo que es indispensable tener un control de calidad durante todas las etapas del ciclo de vida del software. (<http://www.espanol.geocities.com/michelsoftpe/ads.htm>).

2.1.2 SOFTWARE DE CALIDAD

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software, que permitan lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de calidad del software. (Pressman, 24)

2.1.3 CONTROL DE LA CALIDAD DEL SOFTWARE

Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes.

Es necesario definir los parámetros, indicadores o criterios de medición; estos son definidos por innumerables autores, los cuales se denominan y agrupan en formas diferentes.

En el proceso de control se definen los siguientes pasos:

- ◆ Definir el software que va a ser controlado.
- ◆ Seleccionar una medida que pueda ser aplicada al objeto de control.
- ◆ Crear o determinar los métodos de valoración de los indicadores.
- ◆ Definir las regulaciones organizativas para realizar el control.

2.2 ARQUITECTURA DE DESARROLLO CLIENTE/SERVIDOR

El sistema de administración de paquetes turísticos para la empresa “HERUMA LTDA:”, se desarrollará en arquitectura cliente-servidor de dos capas.

La arquitectura cliente/servidor es descrita como aquella que tiene dos o mas capas de acuerdo a como está distribuida la lógica de la aplicación entre el cliente y el servidor. Una arquitectura cliente-servidor debe tener por lo mínimo una capa cliente una capa servidor, este modelo consiste en un cliente y un servidor de base de datos. (Biblioteca de Consulta Microsoft ® Encarta ® 2005. © 1993-2004 Microsoft Corporation)

Bajo este modelo el cliente es el que solicita un servicio, se encarga de la presentación y en parte de las reglas del negocio, y el servidor es el capaz de proporcionar un servicio, se encarga del acceso a los datos y muchas veces de la lógica del negocio mediante procedimientos almacenados.

2.2.1 CLIENTE

Esta compuesta por la interfaz o front-end (parte del software que interactúa con el usuario final) y parte de la lógica del negocio o componentes.

Un cliente se puede comunicar con varios procesos servidores y un servidor se puede comunicar con varios clientes. Los procesos pueden ejecutarse en la misma máquina o en distintas máquinas comunicadas a través de una red. Por lo general, la parte de la aplicación correspondiente al cliente se optimiza para la interacción con el usuario, ejecutándose en su propia máquina, a la que se denomina terminal o cliente, mientras que la parte correspondiente al servidor proporciona la funcionalidad multiusuario centralizada y se ejecuta en una máquina remota, denominada de forma abreviada, simplemente, servidor. (Biblioteca de Consulta Microsoft ® Encarta ® 2005. © 1993-2004 Microsoft Corporation).

Las funciones del cliente son:

- ◆ Para ingresar al sistema podrá utilizar su nombre de usuario y contraseña y se habilitara aquellas áreas a las que el usuario tenga permisos.
- ◆ Manejo de la interfaz del usuario.
- ◆ Captura o validación de los datos de entrada.
- ◆ Consultas y listados de datos.
- ◆ Consultas de asistencia, entre otras.

2.2.2 SERVIDOR

Esta compuesta por la base de datos o back-end (parte que guarda y mantiene la información del negocio) y la mayor parte de las reglas del negocio o componentes contenidas en procedimientos almacenados.

Los servidores tienen fundamentalmente dos funciones, bien “servidores de aplicaciones”, que alojan distintos tipos de programas que pueden llamarse desde y ejecutarse en los terminales, bien “servidores de bases de datos”, que alojan archivos con datos que pueden ser consultados y/o editados y modificados en las máquinas terminales o clientes; también pueden ser servidores de ambos tipos simultáneamente.

Las funciones del servidor son:

- ◆ Gestión de periféricos compartidos.
- ◆ Control de usuarios que pueden acceder al sistema, especificando a que área puede ingresar.
- ◆ El administrador controla el ingreso del cliente.
- ◆ Realizar copia de seguridad de los datos que el sistema maneja.
- ◆ Forzar a un usuario a desconectarse del sistema
- ◆ Enviar mensajes a los usuarios conectados y recibir mensajes desde ellos.

2.2.3 CARACTERÍSTICAS DE LA ARQUITECTURA CLIENTE-SERVIDOR

Una aplicación cliente/servidor típica es un servidor de base de datos al que varios usuarios realizan consultas simultáneamente. El proceso cliente realiza una consulta, el proceso servidor le envía las tablas resultantes de la consulta y el proceso cliente las interpreta y muestra el resultado en pantalla. Los sistemas distribuidos pueden consistir en diversos servidores que alojen datos, de forma que el cliente no tiene por qué conocer exactamente dónde se encuentran, simplemente hace una petición de servicio, y es el sistema servidor el encargado de localizarlos y proporcionar el resultado de la consulta al usuario que hizo la petición (<http://www.academicos.cualtos.udg.mx/info.matica/ceneval2003/ciclo%20de%20vida%20del%20software.doc>).

Las características de la arquitectura cliente servidor son:

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

2.2.4 VENTAJAS DE LA ARQUITECTURA CLIENTE/SERVIDOR

- ◆ Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.

- ◆ El esquema Cliente/Servidor facilita la integración entre sistemas diferentes y comparte información permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. De esta manera, podemos integrar PCs con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
- ◆ Al favorecer el uso de interfaces gráficas interactivas, los sistemas contruidos bajo este esquema tienen mayor interacción más intuitiva con el usuario. El uso de interfaces gráficas para el usuario, el esquema Cliente/Servidor presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- ◆ Una ventaja adicional del uso del esquema Cliente/Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets o el RPC).

La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.

El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante en el ámbito global. (http://fmc.axarnet.es/win2000srv/tema-05/tema_05-1.htm).

2.2.5 DESVENTAJAS DE LA ARQUITECTURA CLIENTE/SERVIDOR

- ◆ Hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos. La seguridad de un esquema Cliente/Servidor es otra preocupación importante. Por ejemplo, se deben hacer verificaciones en el cliente y en el servidor. También se puede recurrir a otras técnicas como el encriptamiento.
- ◆ El desempeño es otro de los aspectos que se deben tener en cuenta en el esquema Cliente/Servidor. Problemas de este estilo pueden presentarse por congestión en la red, dificultad de tráfico de datos, etc.
- ◆ Un aspecto directamente relacionado con lo anterior es el de cómo distribuir los datos en la red. En el caso de una organización, por ejemplo, éste puede ser hecho por departamentos, geográficamente, o de otras maneras. Hay que tener en cuenta que en algunos casos, por razones de confiabilidad o eficiencia, se pueden tener datos replicados, y que puede haber actualizaciones simultáneas.

A otro nivel, una de las decisiones que deben tomar las organizaciones es la de sí comprar o desarrollar los diferentes componentes.

2.3 REQUERIMIENTOS DE INFRAESTRUCTURA

La velocidad de procesamiento está determinada por las características de hardware y software.

2.3.1 HARDWARE

Los requerimientos mínimos de hardware son:

En el servidor se necesita un computador con las siguientes características:

Procesador:	Pentium III de 1.3 Ghz o superior
Sistema Operativo:	Windows 2000 Profesional.
Memoria RAM:	256 Mb.
Disco duro:	60 Gb.
CD ROM	52X
Monitor:	15"

En el cliente se necesita un computador con las siguientes características:

Procesador:	Pentium o superior
Sistema Operativo:	Windows 98 o superior.
Memoria RAM:	64 Mb o superior
Disco duro:	4 Gb mínimo
CD ROM	52X
Monitor:	14"

2.3.2 SOFTWARE

Las herramientas utilizadas durante el desarrollo de nuestro sistema son:

2.3.2.1 Back-end

El software de back-end proporciona el soporte para la administración de los datos en una base de datos.

SQL es un software de back-end que permite procedimientos almacenados (stored procedures) realizados en lenguaje SQL. Se trata de procedimientos que se almacenan semicompilados en el servidor y que pueden ser invocados desde el cliente, estos podrían ser la parte servidor de una aplicación cliente servidor.

Aunque SQL diseñado para funcionar como motor de almacenamiento de datos para miles de usuarios que se conectan a través de una red simultáneamente, puede funcionar también como base de datos independiente directamente en el mismo equipo de una aplicación. Las características de escalabilidad y facilidad de uso de SQL le permiten trabajar eficazmente en un único equipo sin consumir demasiados recursos y sin que sean necesarias tareas administrativas por parte del usuario independiente. Las mismas características permiten a SQL adquirir de forma dinámica los recursos necesarios para admitir miles de usuarios, al tiempo que minimizan la administración y la optimización de bases de datos. El motor de base de datos relacional de SQL se ajusta a sí mismo de forma dinámica para adquirir o liberar la cantidad adecuada de recursos del equipo necesarios para admitir una carga variable de usuarios que tienen acceso a una instancia de SQL en un momento determinado. El motor de base de datos relacional de SQL dispone de características para evitar los problemas lógicos que tienen lugar si un usuario intenta leer o modificar datos que están utilizando otros usuarios.

SQL Server puede manejar perfectamente base de datos de Terabytes con millones de registros y funcionar sin problemas con miles de conexiones, esta

pensado para gestionar tantos clientes simultáneos como admita la potencia de hardware del equipo en el que este instalado.

2.3.2.2 Front-end

Visual Basic 6.0 comercializada por Microsoft para desarrollo de aplicaciones en entornos Windows, como lenguaje de programación de aplicaciones ofimáticas, consultas a bases de datos y creación de páginas de Internet dinámicas.

Es la herramienta mas productiva para crear aplicaciones de alto rendimiento empresariales y basadas en la Web, la integración de Visual Database, Tools y un entorno RAD aumenta la productividad, mientras que la compilación a código nativo proporciona rápidas aplicaciones, proporciona un conjunto completo de herramientas para integrar bases de datos con cualquier aplicación

Visual Basic 6.0 proporciona un conjunto completo de herramientas para integrar bases de datos con cualquier aplicación. Entre las características de bases de datos se incluyen herramientas de diseño para crear y modificar bases de datos de SQL Server. ([http://urania.dis.afit.edu.co/cursos/st725/material/lectintro01.p df](http://urania.dis.afit.edu.co/cursos/st725/material/lectintro01.pdf)).

2.3.2.3 Reportes

Seagate Crystal Reports.- es un poderoso control ActiveX para diseño en el entorno de Visual Basic permite crear fácil y rápidamente informes administrativos comprensibles, personalizados y atractivos. Seagate Crystal Reports también se refiere a informes como facturas, cartas modelo, etiquetas de correo, y otros elementos relacionados que requieren la organización y producción de datos, provee un modelo de objeto completo, que permite diseñar interfaces personalizadas para entrada de datos que controlan los datos del informe.

Un informe es simplemente una presentación organizada de datos. Como una herramienta de administración, el informe proporciona a la gerencia de la empresa la perspectiva necesaria para dirigir efectivamente la organización. (<http://www.seagatesoftware.com/crystalreports/techsupp>).

2.3.2.4 Herramientas CASE de soporte al desarrollo

Rational Rose.- es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Es la herramienta líder en el mundo para el modelado de sistemas complejos y de tiempo real. Ofrece la capacidad de modelar y visualizar procesos de negocios y destacar oportunidades para aumentar la eficiencia.

Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Desarrollo Iterativo

Rational Rose utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

Trabajo en Grupo

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades.

Generador de Código

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

Ingeniería Inversa

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

Mejora la comunicación entre usuario y desarrolladores, cuando se modela casos de uso en Rational Rose, asegura que la solución sea creada con el usuario en mente.

Rational Rose unifica a los analistas de negocios, sistemas y datos al permitirles crear y administrar modelos en una herramienta con un solo lenguaje de modelado. (http://urania.dis.eafit.edu.co/cursos/st725/material/lect_intro01.pdf).

CAPITULO III

ASPECTOS METODOLOGICOS

3.1 CUADRO DE RESUMEN

PARADIGMA ESPIRAL INCREMENTAL ORIENTADO A OBJETOS	METODOLOGÍA OMT	DIAGRAMAS UML	
Análisis	Modelo Estático	Descripción de Actores	
		Diagrama de Casos de Uso	
		Descripción de Casos de Uso	
		Diagrama de Clases	
		Diccionario de Clases	
Diseño	Modelo Dinámico	Diagrama de Interacción	Diagrama de Secuencia
			Diagrama de Colaboración
	Modelo funcional	Diagrama de Actividades	
Construcción	Herramientas de Software	Rational Rose 2000	
		Visual Basic 6.0	
		SQL	
Pruebas	Prueba Funcional	Descripción de la Prueba	
		Procedimiento de la Prueba	
Mantenimiento	La documentación del desarrollo y seguimiento de los modelos facilita el futuro mantenimiento		

FUENTE: Pressman, R. (2003). Ingeniería de Software

ELABORACIÓN: Flores C, Quishpe G.

Aplicando el Paradigma Espiral Incremental resulta imposible poner barreras entre las etapas de Análisis y Diseño.

3.2 PARADIGMA DE DESARROLLO

Paradigma es un modelo de proceso de software que se utiliza como guía para el desarrollo de Sistemas. Cada uno de los esquemas formales en que se organizan las palabras nominales y verbales para sus respectivas flexiones. (Pressman, 24)

3.2.1 PARADIGMA ESPIRAL INCREMENTAL ORIENTADO A OBJETOS

Es un modelo de proceso de software evolutivo que empareja la naturaleza iterativa del prototipado con los aspectos controlados sistemáticos del modelo secuencial lineal. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software, este es desarrollado en series incrementales. Durante las iteraciones iniciales, la versión incremental puede ser un modelo en papel o un prototipo, durante las iteraciones finales, se producen versiones más completas del sistema. (Pressman, 28)

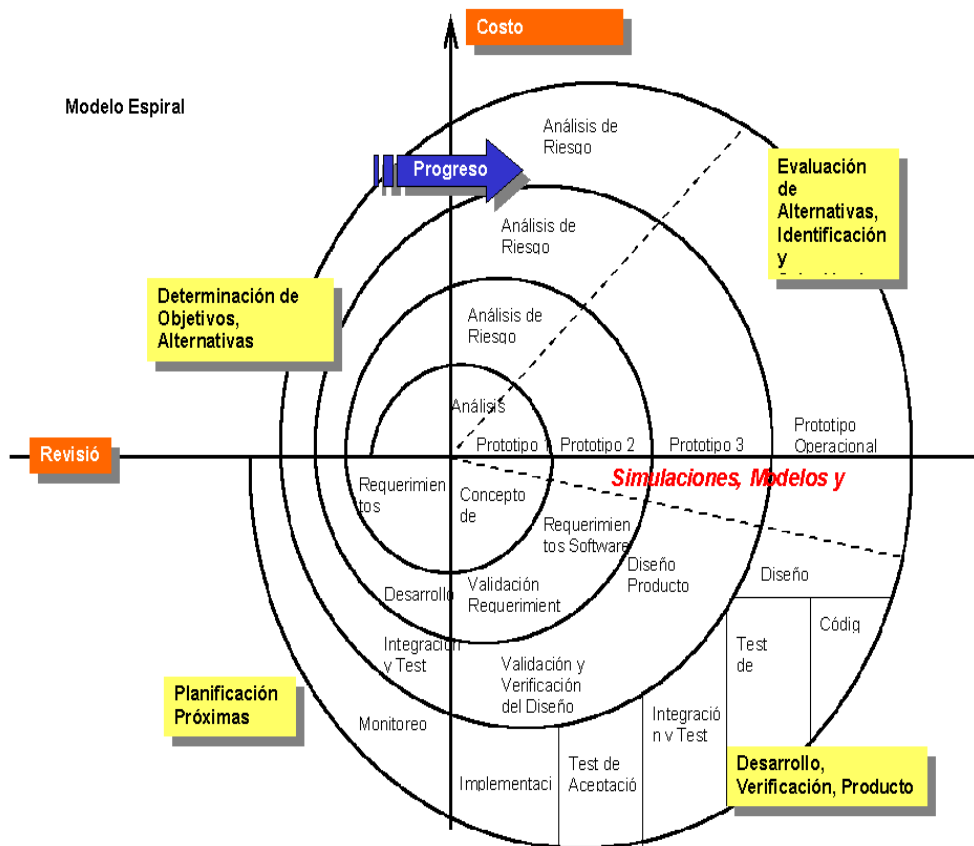


Figura 1. Modelo Espiral Incremental
(<http://www.itlp.edu.mx/publica/tutoriales/analisis/23.htm>)

3.2.1.1 ACTIVIDADES DEL MODELO ESPIRAL INCREMENTAL

El modelo espiral incremental está dividido en un número de actividades, también llamadas regiones de tareas, normalmente entre tres o seis regiones, el número aumenta según la complejidad del programa a desarrollar.

- Comunicación con el cliente.
- Planificación se definen las líneas de tiempo, los recursos y otra información del proyecto.
- Análisis de riesgos

- Ingeniería son tareas requeridas para construir representaciones de la aplicación.
- Construcción y desarrollo son tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- Evaluación del consumidor son tareas requeridas para obtener la opinión del cliente, basada en la evaluación de las representaciones del software durante la etapa de ingeniería e implementadas durante la etapa de construcción y desarrollo.

Al comenzar el desarrollo de software, el equipo de desarrollo se mueve alrededor de la espiral incremental en la dirección de las agujas del reloj, comenzando en el centro. Cada pasada por la región de planificación el plan del proyecto se reajusta, basándose en la evaluación del cliente, mas aun, el gestor del proyecto reajusta el número de iteraciones planeadas para la conclusión del proyecto.

En el eje se separa la evaluación del cliente y la comunicación con el cliente es donde comienza cada nuevo ciclo de la espiral, que puede ser utilizado para crear nuevas versiones del programa o desarrollar nuevos productos que tienen como base el que ya existía, en cualquier caso el comienzo del desarrollo está aquí, por lo tanto la espiral permanece en funcionamiento hasta que el software queda obsoleto. (Pressman, 180)

3.2.1.2 VENTAJAS

- Es una aproximación realista para el desarrollo de grandes proyectos.
- Como es software evoluciona a la vez que el proyecto progresa, el desarrollador y el usuario reaccionan antes a los riesgos en cada nivel.
- Usa el prototipado como un sistema de reducción de riesgos, y puede ser utilizado en cualquier etapa.

- Mantiene el sistema por etapas del ciclo de vida clásico, pero lo incluye en un marco de trabajo que se aproxima más a la realidad. (Ruble, 54)

3.3 METODOLOGÍA DE DESARROLLO

Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software.

3.3.1 METODOLOGÍA DE DESARROLLO ORIENTADO A OBJETOS (OMT)

La esencia del desarrollo orientado a objetos es la identificación y organización de conceptos del dominio de la aplicación y no tanto de su representación final en un lenguaje de programación.

3.3.1.1 Modelo Estático

Define las clases y objetos que intervienen en el desarrollo del sistema y su relación estructural entre cada uno de estos. Mediante este modelo podemos analizar la relación que tiene la estructura de una clase y los posibles cambios que pueden existir al momento de hacer variaciones en el diseño original.

3.3.1.2 Modelo Dinámico

Analiza el comportamiento de los objetos en un tiempo determinado, cada objeto puede considerarse como un ente con estados. Además que podemos concluir la capacidad de un estado, su comportamiento al recibir mensajes de otros objetos y saber con quien y cuando nos comunicamos.

3.3.1.3 Modelo de Procesos

Nos permite conocer como se procesan, transforman y ejecutan los datos, es decir, define las funciones que se ejecutan sobre los datos y como estará estructurado el sistema y cuales son dichos procesos. (RUMBAUGH, 337)

3.3.2 FASES DEL CICLO DE VIDA DEL SOFTWARE

Divide el ciclo de vida del Software en 4 fases: análisis de objetos, diseño del sistema, diseño de objetos e implementación.

Fase 1: Análisis

En primer lugar, se describe el problema: Se obtienen unos requisitos que no den lugar a dudas. En toda la fase de análisis se describe el comportamiento del sistema como una “caja negra”.

En segundo lugar se hacen los diagramas de objetos con un diccionario de datos. Así se obtiene el **modelo de objetos**. En él se define la estructura de los Objetos y Clases así como las relaciones, que les unen. Comprende tanto un diagrama de clases como un Diccionario de Datos que las explique.

A continuación se acomete la creación de un **modelo dinámico** para describir los aspectos de control y evolución del sistema. Incluye un diagrama de Eventos del sistema y un Diagrama de Estados por cada clase que tenga un comportamiento dinámico. (Pressman, 526)

Después se crea un **modelo funcional** que describa las funciones, los valores de entrada y salida, e imponga las restricciones pertinentes. Se suele utilizar los tan discutidos Diagramas de Flujo de Datos Orientados a Objetos.

Por ultimo se verifican todos los modelos creados y se itera para conseguir un refinamiento de los 3 modelos.

<http://www.gnu.org/software/toutx/doc/user/es/x41.html>)

Resumiendo los pasos a seguir son:

- Obtener y escribir una descripción inicial del problema.
- Construir un **modelo de objetos**.
- Construir un **modelo dinámico**.
- Construir un **modelo funcional**.
- Verificar, iterar y refinar los tres modelos.

Fase 2: Diseño del sistema

Comprende la arquitectura básica. En esta fase se tomarán las decisiones estratégicas (a alto nivel) de diseño (estructura global del sistema).

Pasos a seguir:

- Organizar el sistema en subsistemas (conjunto de etapas horizontales)
- Identificar la concurrencia inherente al problema.
- Colocar los subsistemas a sus procesos y tareas.
- Elegir la estrategia básica para almacenar datos.
- Identificar los recursos globales y determinar mecanismos de control de acceso a ellos.
- Elegir un método de implementación del control de software.
- Considerar las condiciones límite.
- Establecer las prioridades.

Fase 3: Diseño de objetos

Es el último paso antes de implementar, y sirve para definir completamente todas las características de los objetos. Se detallan los 3 modelos ya descritos en el análisis de objetos de cara a poder implementarlo, y optimizar el programa (acceso a datos, iteraciones, control, recursos,...). Todo esto ha de hacerse con independencia del lenguaje o entorno en que finalmente codifiquemos y ejecutemos la aplicación.

Pasos a seguir:

- Obtener operaciones para el modelo de objetos a partir de los otros modelos.
- Diseñar algoritmos para implementar operaciones.
- Optimizar los accesos a datos.
- Implementar el control de software mediante el sistema elegido durante el diseño del sistema.
- Ajustar las estructuras de las clases para aumentar la herencia.
- Diseño de la implementación de las asociaciones.
- Determinar la representación exacta de los atributos de los objetos
- Compactar las clases a asociaciones en módulos, ocultando en la parte privada toda la información que deba estar oculta.

Fase 4: Implementación del sistema

Durante la implementación se codifican tanto las estructuras en el dominio de la solución. La base que la sustenta en el diseño de objetos.

- En esta parte se discute los detalles específicos de la implementación de un nuevo sistema utilizando lenguajes, no orientados a objetos y sistemas de administración de base de datos.
- El lenguaje de definición de datos se utiliza para declarar la estructura de los datos el lenguaje de consulta puede estar limitado a un sencillo acceso a registro basado en campos que actúen como claves, o bien podría ofrecer una mayor cantidad de características propias del lenguaje de programación de propósito general.
- Las técnicas de diseño y especificación escritas son útiles para blancos de implementación que vayan mas allá de los lenguajes y de las bases de datos.

Con frecuencia el diseño de una aplicación para una computadora implica una implementación que baya a consistir en una mezcla de software y algunos otros. (Pressman, 26)

3.4 UNIFIED MODELING LANGUAGE (UML)

El lenguaje unificado de modelado, UML, es una notación estándar para el modelado orientado a objetos, resultado de una propuesta de estandarización promovida por el consorcio OMG (Object Management Group). UML se ha extendido de tal manera en la industria y en el mundo académico que actualmente se ha convertido en un estándar “de facto” en el modelado orientado a objetos. (Pressman, 252)

3.4.1 OBJETIVOS DEL UML

- Es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad.
- No pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos para casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería del software, como son la encapsulación y componentes. (Pressman, 253)

3.4.2 ÁREAS CONCEPTUALES DE UML

Los conceptos y modelos de UML pueden agruparse en las siguientes áreas conceptuales:

Estructura estática

En cualquier modelo preciso debe primero definir su universo. Este conjunto de construcciones es la estructura estática. La estructura estática se expresa con diagramas de clases y puede usarse para generar la mayoría de las declaraciones de estructuras de datos en un programa.

Comportamiento dinámico

Hay dos formas de modelar el comportamiento, una es la historia de la vida de un objeto y la forma como interactúa con el resto del mundo y la otra es por los patrones de comunicación de un conjunto de objetos conectados. La visión de un objeto aislado es una maquina de estados, muestra la forma en que el objeto responde a los eventos en función de su estado actual. La visión de la interacción de los objetos se representa con los enlaces entre objetos junto con el flujo de mensajes y los enlaces entre ellos. (Pressman, 280)

Construcciones de implementación

Los modelos UML tienen significado para el análisis lógico y para la implementación física. Un componente es una parte física reemplazable de un sistema y es capaz de responder a las peticiones descritas por un conjunto de interfaces.

Organización del modelo

La información del modelo debe ser dividida en piezas coherentes, para que los equipos puedan trabajar en las diferentes partes de forma concurrente. El conocimiento humano requiere que se organice el contenido del modelo en paquetes de tamaño modesto.

3.4.3 MECANISMOS DE EXTENSIÓN

UML tiene una limitada capacidad de extensión pero que es suficiente para la mayoría de las extensiones que requiere al día a día sin la necesidad de un cambio en el lenguaje básico. Un estereotipo es una nueva clase de elemento de modelado con la misma estructura que un elemento existente pero con restricciones adicionales.

3.4.4 DESCRIPCIÓN DE LOS DIAGRAMAS UML

A continuación daremos a conocer cada uno de los siguientes diagramas:

3.4.4.1 Modelo Estático

3.4.4.1.1 Diagramas de caso de uso

Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje, es decir, son requerimientos funcionales que describen de una manera detallada el comportamiento del sistema con los distintos Actores que interactúan con él.

- Los casos de uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema.

- Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.
- Los Casos de Uso son descripciones de la funcionalidad del sistema independientes de la implementación.
- Los Casos de Uso cubren la carencia existente en métodos previos en cuanto a la determinación de requisitos.
- Los Casos de Uso particionan el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo.

Actores

Hay cuatro grandes categorías de actores:

- Principales: interaccionan con el sistema para explotar su funcionalidad. Trabajan directa y frecuentemente con el software. Personas que usan el sistema.
- Secundarios: soporte del sistema para que los primarios puedan trabajar, es decir, las personas que efectúan tareas administrativas o de mantenimiento.
- Externos: los dispositivos que forman parte del dominio de aplicación y que deben ser utilizados.
- Otros sistemas: los sistemas con los que debe interactuar.
- Iniciadores: no interactúan con el sistema pero desencadenan el trabajo de otro actor.

Relación en los diagramas de caso de uso en UML

- Comunicación: Cada uno de estos medios de unión entre dichas cosas
- Inclusión: una instancia del Caso de Uso origen incluye también el comportamiento descrito por el Caso de Uso destino. <<incluye>>.
- Extensión: el caso de uso origen extiende el comportamiento del Caso de Uso destino. <<extend>>.

- Herencia: el Caso de Uso origen hereda la especificación del Caso de Uso destino y posiblemente la modifica y/o amplía.

Parámetros en la construcción de un caso de uso

Un caso de uso debe ser simple, claro y conciso. Generalmente hay pocos actores asociados a cada Caso de Uso. A continuación detallamos las preguntas que se deben realizar en la construcción de un caso de uso:

- ¿Cuáles son las tareas del actor?
- ¿Qué información crea, guarda, modifica, destruye o lee el actor?
- ¿Debe el Actor notificar al sistema los cambios externos?
- ¿Debe el Sistema informar al actor de los cambios internos?

Descripción del caso de uso comprende

- El inicio: ¿Cuándo y qué actor lo produce?
- El fin: ¿Cuándo se produce y qué valor devuelve?
- La interacción actor - caso de uso: ¿Qué mensajes intercambian ambos?
- Objetivo del caso de uso: ¿Qué lleva a cabo o intenta?
- Cronología y origen de las interacciones
- Repeticiones de comportamiento: ¿Qué ejecuciones alternativas se presentan en el Caso de Uso? (<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x332.html>)

3.4.4.1.2 Diagrama de Clases

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc.). Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). El diagrama de clases de más alto nivel (main class diagram), será lógicamente un dibujo de los paquetes que componen el sistema. A su vez cada paquete tendrá un main class diagram que muestra las clases del paquete. (<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x332.html>).

Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, su cardinalidad (cuantos objetos intervienen en la relación) y su opcionalidad (cuando un objeto es opcional el que intervenga en una relación). La descripción de clases complejas se puede documentar con diagramas de estados.



Figura 2. Símbolos en Rose para visibilidad privada, protegida
(<http://www ldc.usb.ve/ternuel/ci3711/aoo/aoo1.html>)

La sintaxis de una operación en UML es:

visibility name (parameter-list): return-type-expression {property-string}

Donde cada uno de los parámetros en parameter-list se denota igual que un atributo. Los demás elementos son los mismos que encontramos en la notación de un atributo.

Otro detalle importante que se puede especificar en los atributos y operaciones de una clase es su alcance. El alcance de propiedad de una característica especifica si la propiedad aparece en cada instancia de la clase o si sólo hay una instancia la propiedad para todas las instancias de la clase. En UML se pueden especificar dos tipos de alcance de propiedad:

- ◆ instancia: Cada instancia de la clase tiene su propio valor.
- ◆ clasificador: Sólo hay un valor para todas las instancias del clasificador.

Los elementos adicionales que pueden aparecer en la representación de una relación son:

- ◆ Rol: Identifica con nombres a los elementos que aparecen en los extremos de la línea que denota la relación, dicho nombre describe la semántica que tiene la relación en el sentido indicado. Por ejemplo, la asociación de composición entre Maquina e Ingrediente recibe el nombre de existencias como rol en ese sentido.
- ◆ Multiplicidad: Indica la cardinalidad de la relación. En el ejemplo se utilizan 1, 1 *, 5, *, como indicadores de multiplicidad. (<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x332.html>)

Una asociación binaria se representa mediante una línea sólida que une dos clases, se trata de una relación entre las dos clases no muy fuerte, es decir, no se exige dependencia existencial ni encapsulamiento. Un ejemplo de asociación de este tipo es la que existe entre una compañía y sus empleados. En este caso la asociación binaria recibe el nombre Trabaja Para, y en ella intervienen una o más instancias (objetos) de la clase Persona denominadas empleado. Además cada empleado se asocia a un empleador que en este caso es único.

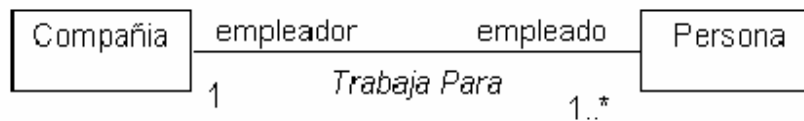


Figura 3. Ejemplo de asociación binaria
(<http://www ldc.usb.ve/ternuel/ci3711/aoo/aoo1.html>)

La agregación (aggregation) es, como ya hemos mencionado, una forma especial de asociación que especifica una relación todo-parte entre el agregado (todo) y una parte que lo compone. Una agregación se representa mediante un rombo en el extremo “todo” de la relación.

Una agregación de composición o simplemente composición (composite aggregation) es una agregación más fuerte que implica:

- ◆ Dependencia existencial: El elemento dependiente desaparece al destruirse el que lo contiene y, si es de cardinalidad 1, es creado al mismo tiempo.
- ◆ Pertenencia fuerte: Se puede decir que el objeto contenido es parte constitutiva y vital del que lo contiene.
- ◆ No compartición: Los objetos contenidos no son compartidos, esto es, no forman parte del estado de otro objeto.

Una clase asociación (association class) es un elemento de modelado que tiene propiedades tanto de asociación como de clase. Una clase asociación puede ser vista tanto como una asociación que también tiene propiedades de clase, o como una clase que también tiene propiedades de asociación. Se representa gráficamente como una clase unida por una línea punteada a una asociación.

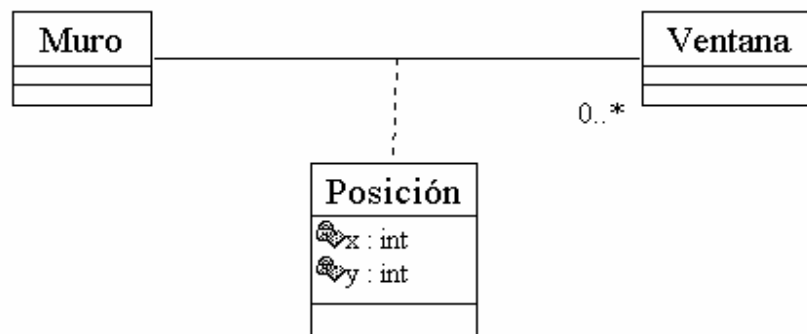


Figura 4. Ejemplo clase de asociación
 (<http://www ldc.usb.ve/ternuel/ci3711/aoo/aoo1.html>)

3.4.4.1.3 Diagrama de objetos

Diagrama utilizado para describir la vista estática de un sistema en un determinado momento.

Los diagramas de objetos modelan las instancias de elementos contenidos en los diagramas de clases. Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. En UML, los diagramas de clase se utilizan para visualizar los aspectos estáticos del sistema y los diagramas de interacción se utilizan para ver los aspectos dinámicos del sistema, y constan de instancias de los elementos del diagrama de clases y mensajes enviados entre ellos. En un punto intermedio podemos situar los diagramas de objetos, que contiene un conjunto de instancias de los elementos encontrados en el diagrama de clases, representando sólo la parte estática de un interacción, consistiendo en los objetos que colaboran pero sin ninguno de los mensajes intercambiados entre ellos. (<http://www.omg.org/technology/documents/formal/uml.htm>)

3.4.4.2 Modelo Dinámico

3.4.4.2.1 Diagrama de colaboración

Diagrama utilizado para describir la organización estructural de los objetos que envían y reciben mensajes.

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas entorno a los objetos y los enlaces entre ellos.

Los diagramas de secuencia proporcionan una forma de ver el escenario en un orden temporal (qué pasa primero, qué pasa después). Los clientes entienden fácilmente este tipo de diagramas, por lo que resultan útiles en las primeras fases de análisis. Por contra los diagramas de colaboración proporcionan la representación principal de un escenario, ya que las colaboraciones se organizan entorno a los enlaces de unos objetos con otros. Este tipo de diagramas se utilizan más frecuentemente en la fase de diseño, es decir, cuando estamos diseñando la implementación de las relaciones.

3.4.4.2.2 Diagrama de secuencia

Diagrama de interacción utilizado para resaltar la ordenación temporal de los mensajes.

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. En aplicaciones grandes además de los objetos se muestran también los componentes y casos de uso. El mostrar los componentes tiene sentido ya que se trata de objetos reutilizables, en cuanto a los

casos de uso hay que recordar que se implementan como objetos cuyo rol es encapsular lo definido en el caso de uso.

3.4.4.3 Modelo Funcional

3.4.4.3.1 Diagrama de estados

Diagrama utilizado para describir los estados por lo que pasa un objeto.

“Los diagramas de estado son una técnica conocida para describir el comportamiento de un sistema. Describen todos los estados posibles en los que puede entrar un objeto, como resultado de los eventos que llegan a él. En la mayor parte de las técnicas Orientadas a Objetos, los diagramas de estados se dibujan para una sola clase, mostrando el comportamiento de un solo objeto durante todo su ciclo de vida”. (Pressman, 178)

3.4.4.3.2 Diagrama de actividades

Diagrama utilizado para mostrar el flujo de control entre actividades.

Elementos: estados de actividad, transiciones, objetos, bifurcación, divisiones y uniones, calles, estado inicial, estado de parada (fin). (Pressman, 285)

Un diagrama de actividades puede considerarse como un caso especial de un diagrama de estados en el cual casi todos los estados son estados acción (identifican una acción que se ejecuta al estar en él) y casi todas las transiciones evolucionan al término de dicha acción (ejecutada en el estado anterior). Un diagrama de actividades puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Permiten representar transiciones internas al margen de las transiciones o eventos externos.

- ◆ Un estado de acción representa un estado con acción interna, con por lo menos una transición que identifica la culminación de la acción (por medio de un evento implícito). No deben tener transiciones internas ni transiciones

basadas en eventos, ya que si fuera este el caso, se representaría con un diagrama de estados. Los estados de acción se representan por un rectángulo con bordes redondeados, y permiten modelar un paso dentro de un algoritmo. (<http://urania.dis.eafit.educo/cursos/st725/mteral/lectintro01.p df>)

3.4.4.4 Pruebas

Una prueba es un conjunto de actividades que se puede planificar por adelantado y llevar a cabo metódicamente para garantizar una buena construcción del sistema desarrollado. Las pruebas deben ser planificadas y llevadas a cabo en su totalidad, tomando en cuenta el tiempo, esfuerzo y recursos.

Hay diferentes tipos de pruebas, entre ellas tenemos:

- Prueba de Unidad.
- Prueba de Integración.
- Prueba de Validación y Verificación.
- Pruebas del Sistema.
- Pruebas de Recuperación.
- Pruebas de Seguridad.
- Prueba funcional.
- Prueba de desempeño.

3.4.4.4.1 Pruebas de unidad

Las pruebas de unidad descubren errores dentro del ambiente de un módulo, para lo cual se debe probar la interfaz del módulo para asegurar que la información fluye en forma adecuada.

Este tipo de prueba permite descubrir una serie de errores, como: tipificación inconsistente, nombres de variables incorrectos, tipos de datos inconsistentes,

incorrecto direccionamiento de memoria, representación incorrecta de una expresión, operadores lógicos incorrectos, ciclos infinitos, validación incorrecta de datos, etc.

Antes de iniciar cualquier otra prueba, es preciso probar el flujo de los datos del módulo, ya que si los datos no ingresan correctamente, todas las demás pruebas no tendrán ningún valor. (Yourdon, 562)

3.4.4.2 Pruebas de integración.

Una vez que los módulos hayan pasado la prueba de unidad y funcionen adecuadamente, es necesario ver si hay problemas en la unión de éstos, ya que los datos se pueden perder al pasar de un módulo a otro, las sub. funciones cuando se combinan, pueden no causar el objetivo deseado, etc. (Yourdon, 261)

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción.

3.4.4.3 Pruebas de validación y verificación.

La verificación es un conjunto de actividades que aseguran que el software implementa correctamente una función específica, en cambio la Validación es un conjunto de actividades que aseguran que el software construido se ajusta a los requisitos y necesidades del usuario. (Yourdon, 482)

3.4.4.4 Pruebas del sistema

La prueba del sistema consiste en la delegación de culpabilidad, esto ocurre cuando en el sistema se descubre algún error y cada uno los creadores del sistema echa la culpa a los otros, para evitar este inconveniente se debe hacer lo siguiente:

Diseñar rutas de manejo de errores que prueben toda la información procedente de otros elementos del sistema. Llevar a cabo una serie de pruebas que simulen la presencia de datos en mal estado.

Registrar los resultados de pruebas como evidencia.

Participar en la planificación y diseño de pruebas para asegurarse de que el software sea probado en forma adecuada. (Yourdon, 484)

3.4.4.5 Pruebas de recuperación

Esta prueba consiste en forzar al sistema para que produzca fallas de software de muchas maneras, para verificar que la recuperación se lleve a cabo.

3.4.4.6 Pruebas de seguridad

Este tipo de prueba verifica que el sistema cumpla con los mecanismos de protección, para evitar el mal uso del mismo y perjudicar al negocio.

Los motivos por los que pueden ingresar a un sistema violando las reglas del mismo son: por joby, venganza, obtener ganancias personales ilícitas, curiosidad. (Yourdon, 488).

3.4.4.4.7 Prueba funcional

Esta es la forma más común de prueba; su propósito es asegurar que el sistema realiza sus funciones normales de manera correcta. Así, los casos de prueba se desarrollan y se alimentan al sistema las salidas (y los resultados de los archivos actualizados) se examinan para ver si son correctos.

Por medio de esta prueba comprobamos si nuestro sistema está funcionando normalmente y de forma correcta. (Yourdon, 484)

3.4.4.4.8 Prueba de desempeño

- ◆ El propósito de este tipo de prueba es asegurarse que el sistema pueda manejar el volumen de datos y transacciones de entrada especificados en el modelo de implantación del usuario, además de asegurar que tenga el tiempo de respuesta requerido. (http://urania.dis.eafit.edu.co/curs_os/st725/material/lect_intro01.pdf).

Esto puede requerir que el equipo que realiza el proyecto simule una gran red de terminales en línea, de manera que se pueda engañar al sistema para que crea que está operando con una gran carga.

Al probar el ingreso de un alto número de datos en el sistema, se comprueba que el sistema está capacitado para manejar dichos datos. (Yourdon, 485).

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- ◆ La metodología empleada es de fácil comprensión y cuenta con amplia documentación, lo cual permite guiar a los desarrolladores que no son expertos en la elaboración de software.
- ◆ Al construir prototipos se controla el sistema en cada fase y se corrigen errores a tiempo, además de realizar cambios, de ser necesarios.
- ◆ El uso de las herramientas de programación Visual Basic 6.0 y la base de datos SQL Server 2000, resultaron adecuadas y el aprendizaje fue sencillo.
- ◆ Al utilizar diagramas UML obtenemos múltiples perspectivas del sistema, el modelo integra estas perspectivas de manera que podemos construir un sistema óptimo y auto consistente.
- ◆ Los conocimientos adquiridos en la carrera fueron de gran utilidad para aplicarlos en la realización de este sistema.

4.2 RECOMENDACIONES

- ◆ Si se desea llevar un estricto control de asistencia de guías y vendedores se debería añadir el módulo de Control de Asistencia de personal y se sugeriría crear un módulo de rol de pagos para todos los empleados.
- ◆ Con la finalidad de reducir el tiempo en cada transacción se puede implementar un servidor como tal con mayor capacidad de almacenamiento y de servicio a los diferentes clientes.
- ◆ Se debe capacitar a los usuarios del Sistema, en cuanto a respaldos y utilización y funcionamiento del mismo, con esto se logrará un aprovechamiento óptimo del sistema.

4.3 BIBLIOGRAFÍA

- ◆ Pressman, Roger S. (1988). Ingeniería de Software. 2° edición. México. Editorial McGraw – Hill. .
- ◆ Ruble, David A.: (1997/1998). Análisis y diseño practico para Sistemas Cliente / Servidor con GUI. Editorial Prentice
- ◆ Yourdon, Edward. (1993). Análisis Estructurado Moderno. México. Editorial Prentice –Hall Hispanoamericana.

REFERENCIAS DE LA WEB

- ◆ Biblioteca de Consulta Microsoft ® Encarta ® 2005. © 1993-2004 Microsoft Corporation.
- ◆ <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x332.html>
- ◆ <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x332.html>
- ◆ http://fmc.axarnet.es/win2000srv/tema-05/tema_05-1.htm
- ◆ http://urania.dis.eafit.edu.co/cursos/st725/material/lect_intro01.pdf
- ◆ <http://www.academicos.cualtos.udg.mx/Informatica/Ceneval2003/Ciclo%2de%2Vida%2del%2Software.doc>
- ◆ <http://www.austral.edu.ar/web/ingenieria/design/transparencias/7>
- ◆ <http://www.espanol.geocities.com/michelsoftpe/ads.htm>
- ◆ <http://www.gnu.org/software/toutdoux/doc/user/es/x417.html>
- ◆ <http://www.omg.org/technology/documents/formal/uml.htm>

ANEXOS

MANUAL TÉCNICO

1 ANÁLISIS Y REQUERIMIENTOS

1.1 MODELO ESTÁTICO

1.1.1 Identificación de Actores



Secretaria

SECRETARIA.- Es la persona que registra a los vendedores, clientes, paquetes turísticos, ventas, pagos, comisiones y genera los diferentes reportes.



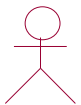
Vendedor

VENDEDOR.- Entrega reporte diarios de venta de paquetes, solicita información sobre clientes e informe de pago de comisiones.



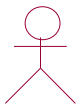
guía turístico

GUÍA TURÍSTICO.- Solicita información sobre tour a dirigir e informe de pago de comisiones.



Cliente

CLIENTE.- Solicita información sobre paquetes turísticos, entrega datos personales.



Gerente General

GERENTE GENERAL solicita informes sobre ventas de paquetes, paquete más vendido, vendedor con más ventas, comisiones por ventas, pagos y asistencia de vendedores.



Gerente de Operaciones

GERENTE DE OPERACIONES.- solicita informes sobre ventas de paquetes, paquete más vendido, vendedor con más ventas, comisiones por ventas, pagos y asistencia de vendedores, realiza venta de paquetes turísticos.

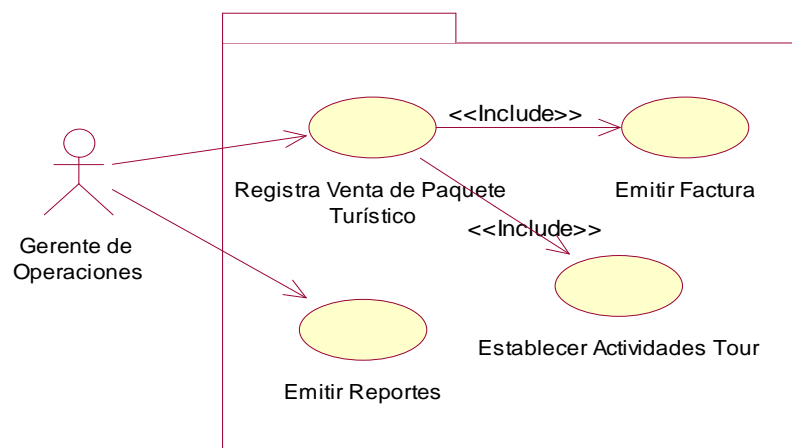
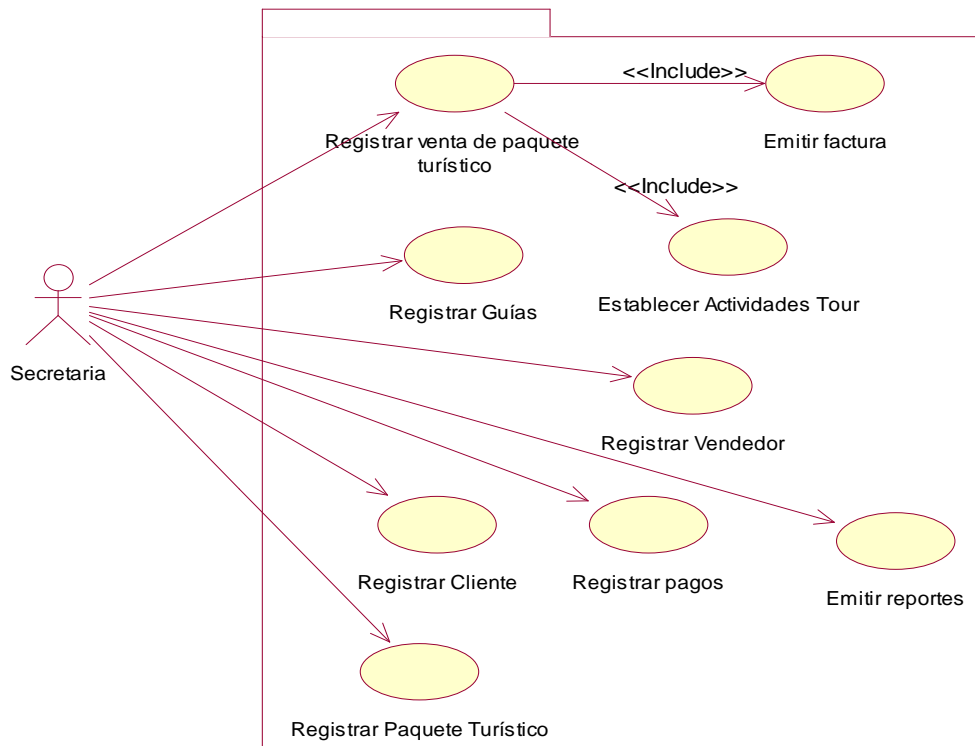


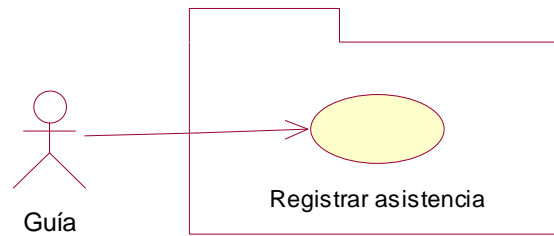
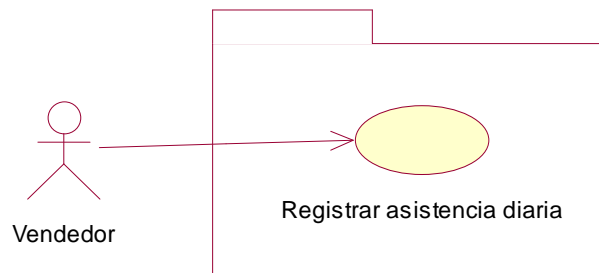
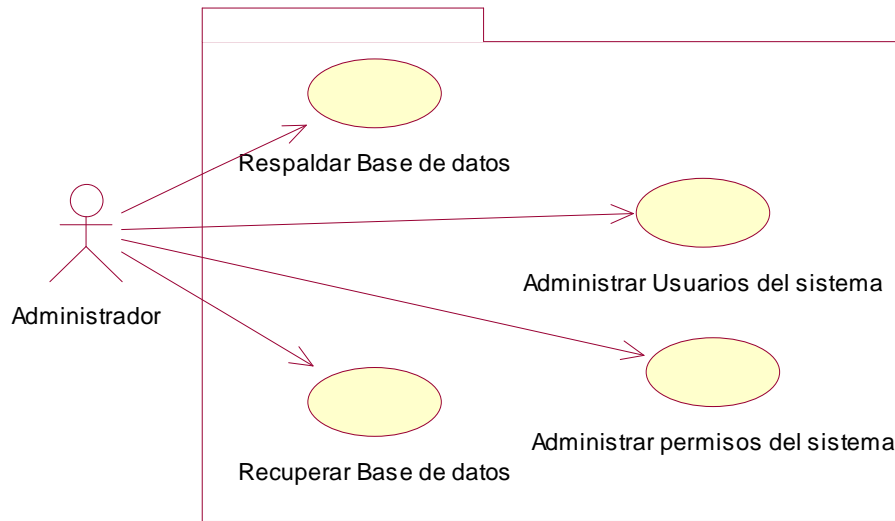
administrador

ADMINISTRADOR.- Administra y da soporte técnico al sistema.

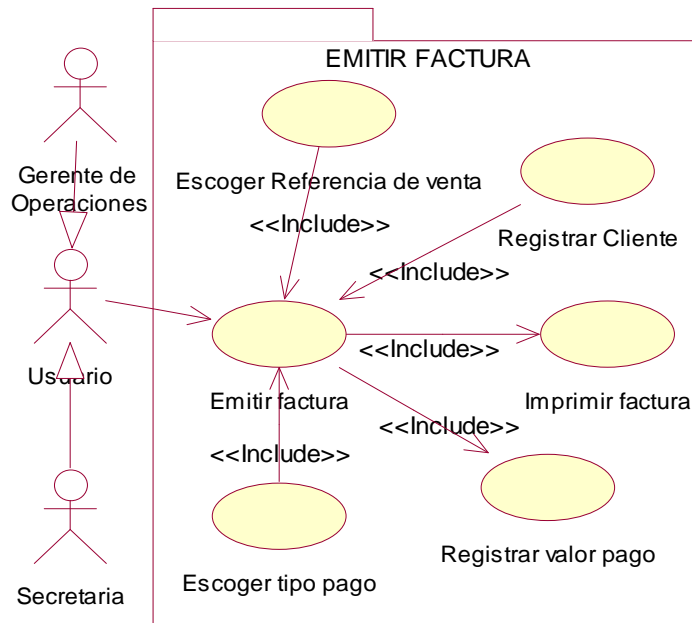
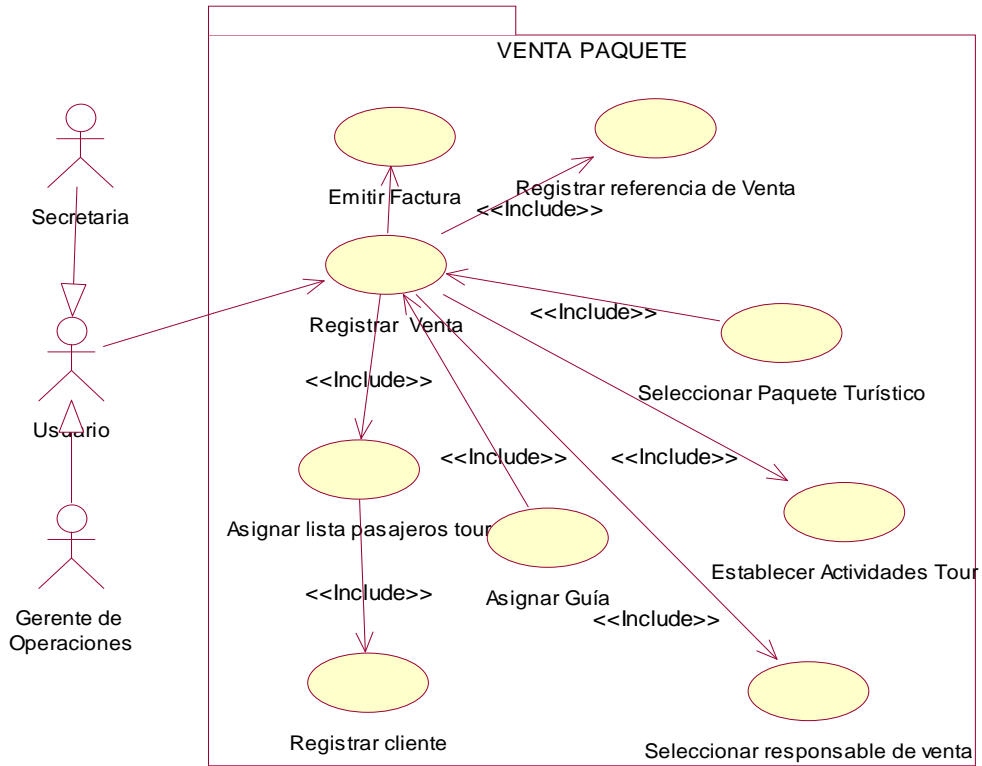
1.1.2 Diagramas de Casos de Uso

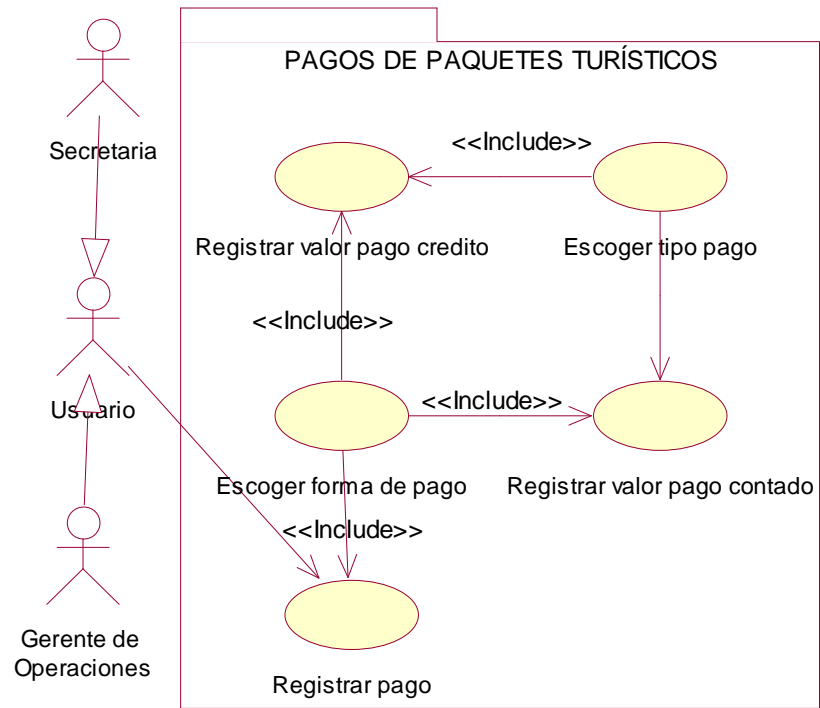
1.1.2.1 Por Actores





1.1.2.2 Por Actividades





1.1.2.3 Diccionario De Casos De Uso

SECRETARIA:	SISTEMA:
1. Registrar venta	Crear código de referencia de venta turístico Registrar número de personas integran tour Establecer fecha tour Seleccionar paquete turístico Seleccionar vendedor Establecer porcentaje de ganancia
2. Emitir Factura	Seleccionar código de referencia de venta Registrar datos del cliente Registrar forma de pago Registrar valor de pago Registrar información de pago Imprimir factura
3. Operar Tour	Buscar el tour a operar Asignar guía a tour Asignar lista de pasajeros de tour Asignar fechas a actividades de tour
3. Registrar pagos de créditos	Buscar No. De factura Seleccionar la cuota a cobrar Registrar tipo pago Registrar valor pago
4. Emitir Reportes	Generar reportes Imprimir reportes
5. Registrar paquetes turísticos	Registrar servicios Registrar precio de servicios por persona Seleccionar ciudad Registrar precio paquete por persona
6. Registrar vendedores	Verificar vendedor Ingresar datos personales Guardar datos de vendedor

Diccionario De Casos De Uso. 1.1.2.3 CONTINUACION

7. Registrar clientes	<p>Verificar cliente</p> <p>Ingresar datos personales</p> <p>Guardar datos de cliente</p>
8. Registrar Guías turísticos	<p>Verificar Guía</p> <p>Ingresar datos personales</p> <p>Guardar datos de Guía</p>
GERENTE OPERACIONES	SISTEMA:
9. Registrar venta	<p>Crear código de referencia de venta turístico</p> <p>Registrar número de personas integran tour</p> <p>Establecer fecha tour</p> <p>Seleccionar paquete turístico</p> <p>Seleccionar vendedor</p> <p>Establecer porcentaje de ganancia</p>
10. Emitir Factura	<p>Seleccionar código de referencia de venta</p> <p>Registrar datos del cliente</p> <p>Registrar forma de pago</p> <p>Registrar valor de pago</p> <p>Registrar información de pago</p> <p>Imprimir factura</p>
11. Operar Tour	<p>Buscar el tour a operar</p> <p>Asignar guía a tour</p> <p>Asignar lista de pasajeros de tour</p> <p>Asignar fechas a actividades de tour</p>
12. Registrar pagos créditos	<p>Buscar No. De factura</p> <p>Seleccionar la cuota a cobrar</p> <p>Registrar tipo pago</p> <p>Registrar valor pago</p>

Diccionario De Casos De Uso. 1.1.2.3 CONTINUACION

13. Emitir Reportes	Generar reportes Imprimir reportes
14. Administrar información adicional	Registrar Bancos Registrar Información de tarjetas de crédito Registrar intereses
VENDEDOR	SISTEMA
15. Registrar asistencia diaria	Ingresar a Control de asistencia Digitar password Guardar asistencia
GUIA TURISTICO	SISTEMA
16. Registrar asistencia	Ingresar a Control de asistencia Digitar password Guardar asistencia

FUENTE: (HERUMA Ltda.)

ELABORACIÓN: Flores C, Quishpe G.

1.1.3 Diagrama de Clases

1.1.3.1 Diccionario de Clases

CLASE	DESCRIPCIÓN
GUÍA	Se registra los datos de guía es decir, tipo de guía e idioma del guía.
IDIOMA	Representa los diferentes tipos de idiomas que puede tener un guía turístico.
CLIENTE	Representa los datos de las personas naturales o jurídicas quienes adquieren los paquetes turísticos.
VENDEDOR	Representa los datos de los responsables de realizar las ventas de los paquetes turísticos.
USUARIOSISTEMA	Se registra los datos de las personas que tienen acceso al sistema con sus diferentes permisos.
PERSONA	Almacena datos personales de los diferentes tipos de personas como son clientes, vendedores, guías y usuarios del sistema.
PAQUETE TURÍSTICO	Datos e información del producto que comercializa la empresa, se detallan todas las especificaciones de servicio que ofrece la empresa.
SERVICIOS	Se asigna los servicios que contienen cada uno de los diferentes paquetes.
VENTAPAQUETE	Representa a los datos de las proformas que se realizan antes de las ventas.
FACTURA	Se registra toda la información requerida al realizar una venta y sus respectivas formas de pago.

Diccionario De Clases. 1.1.3.1 CONTINUACION

PAGO	Se determina la forma de pago de los paquetes turísticos vendidos.
PAGOCHEQUE	Representa el pago realizado con cheque.
PAGOTARJETA	Representa el pago realizado con tarjeta.
PAGOCREDITO	Representa el pago realizado con crédito personas.
INTERESTARJETA	Representa los diferentes porcentajes de interés asignados por los créditos concedidos.
ASISTENCIA	Tiene que ver con el control de la asistencia.
ACTIVIDADESTOUR	Representa información a de las diferentes actividades a realizarse previa una venta.
ESTADOVENTA	Se registra el estado de una venta.
EMPRESA	Representa la información general de la Empresa.

FUENTE: (HERUMA Ltda.)

ELABORACIÓN: Flores C, Quishpe G.

1.1.4 Diagrama de Objetos

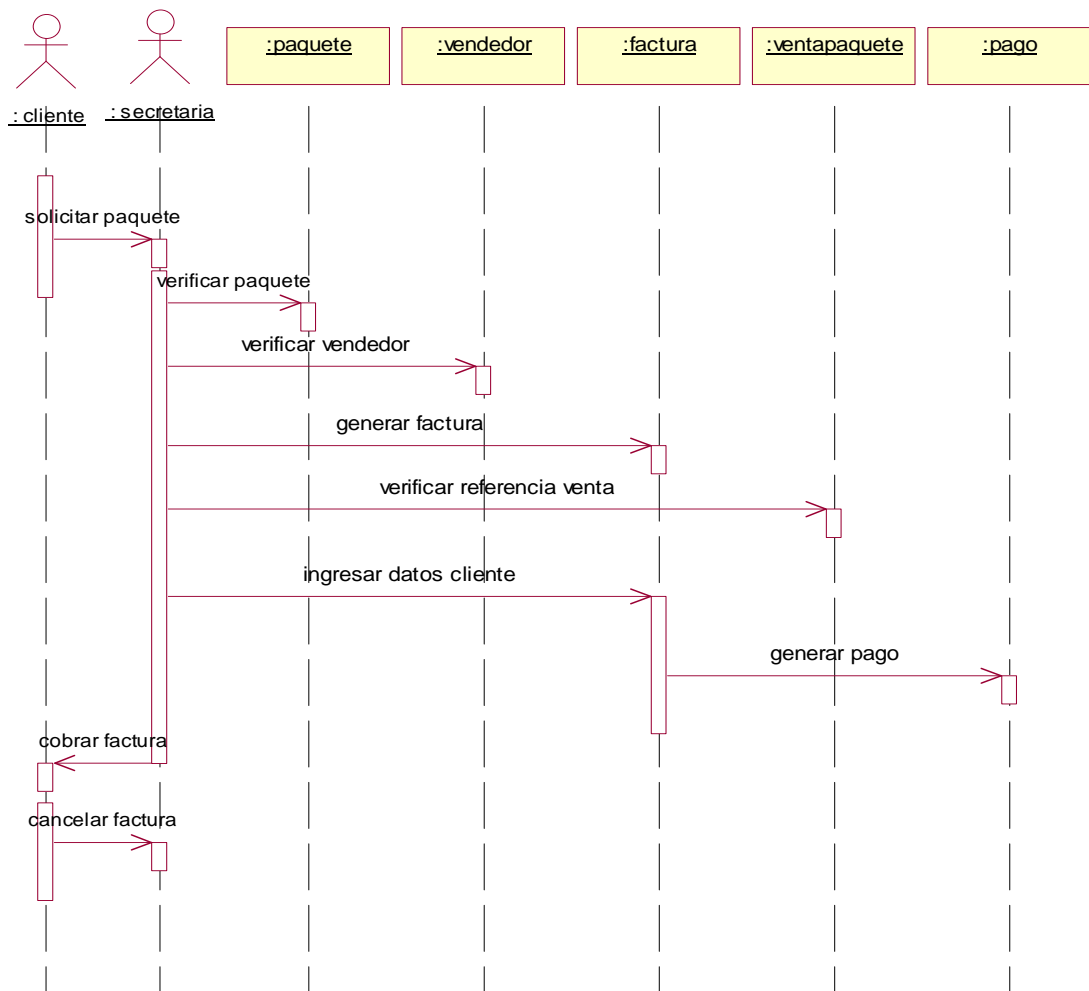
2 DISEÑO

2.1 MODELO DINÁMICO

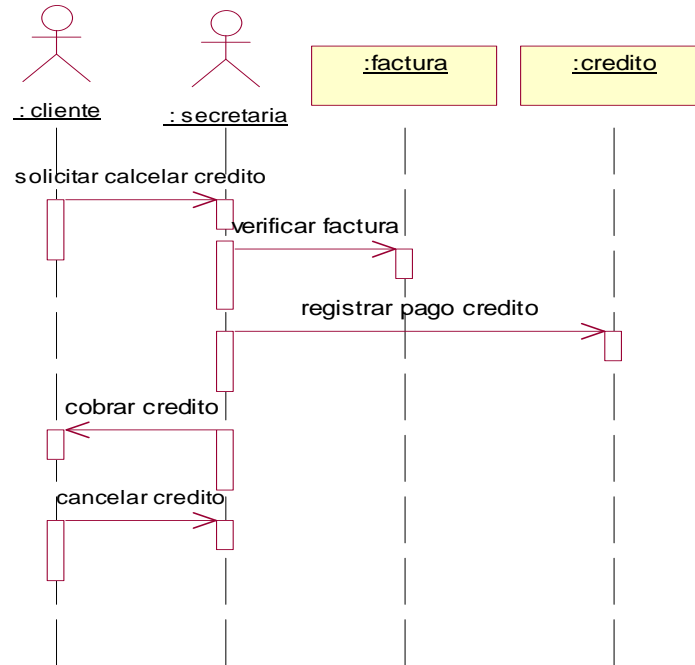
2.1.1 Diagramas de Interacción

2.1.1.1 Diagramas de Secuencia

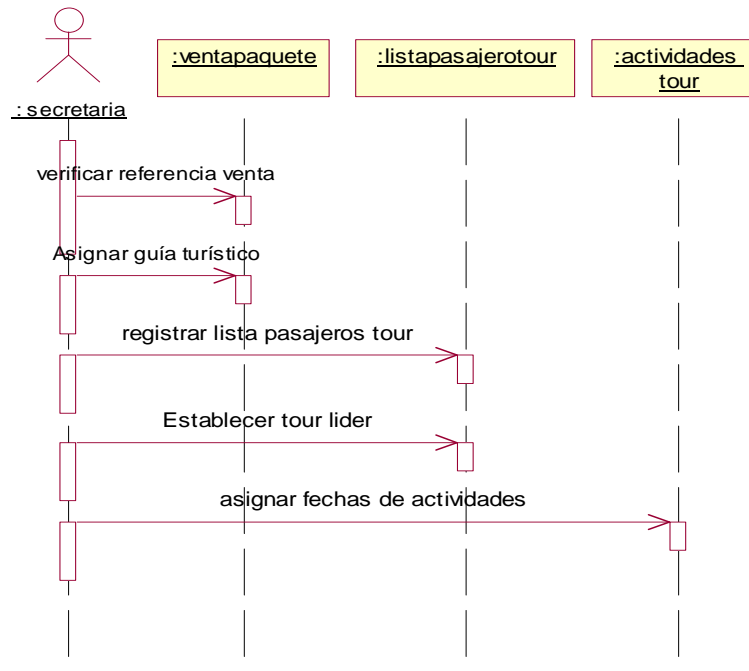
Venta de paquetes turísticos



Pago crédito

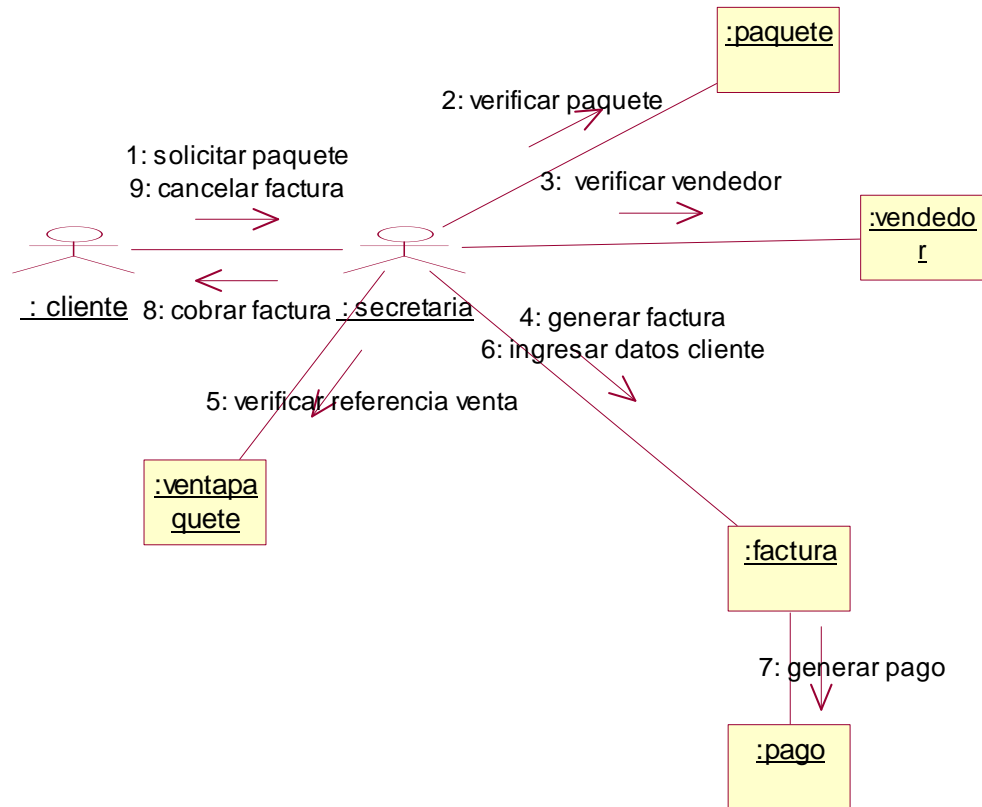


Operar paquete turístico

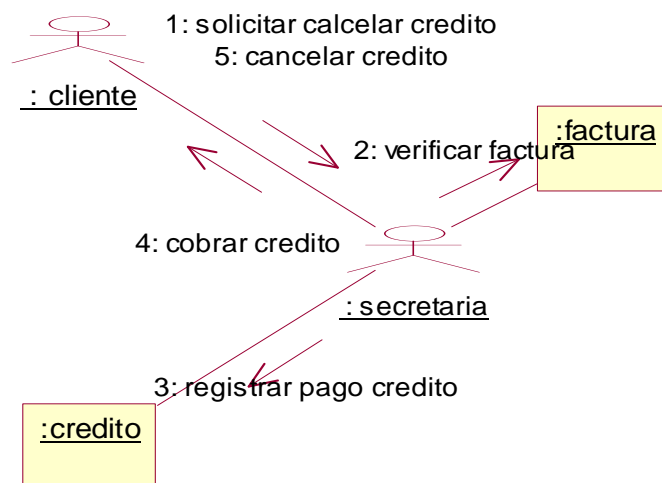


2.1.1.2 Diagramas de Colaboración

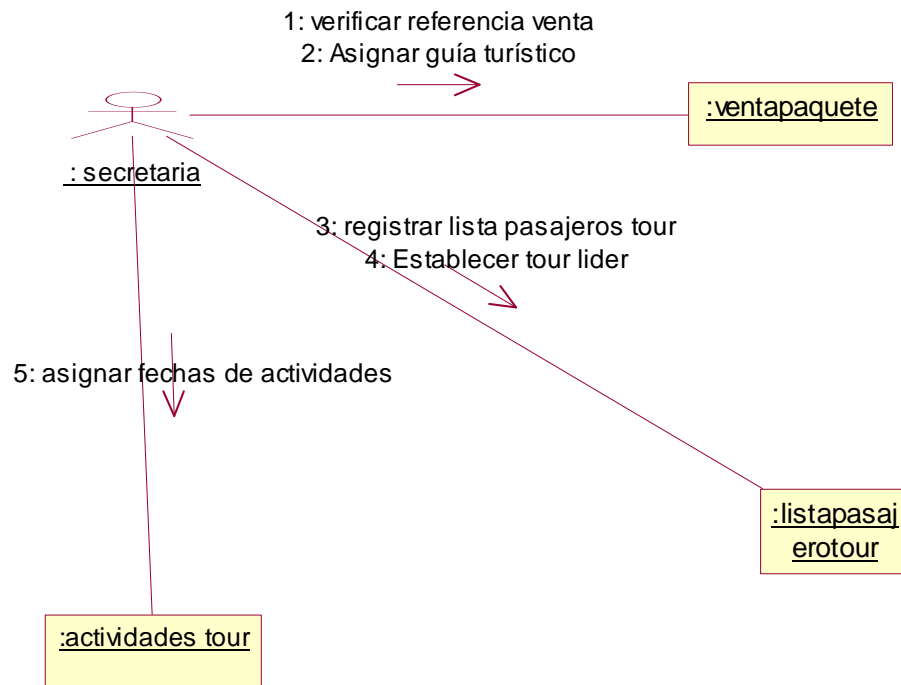
Venta de paquetes turísticos



Pago crédito

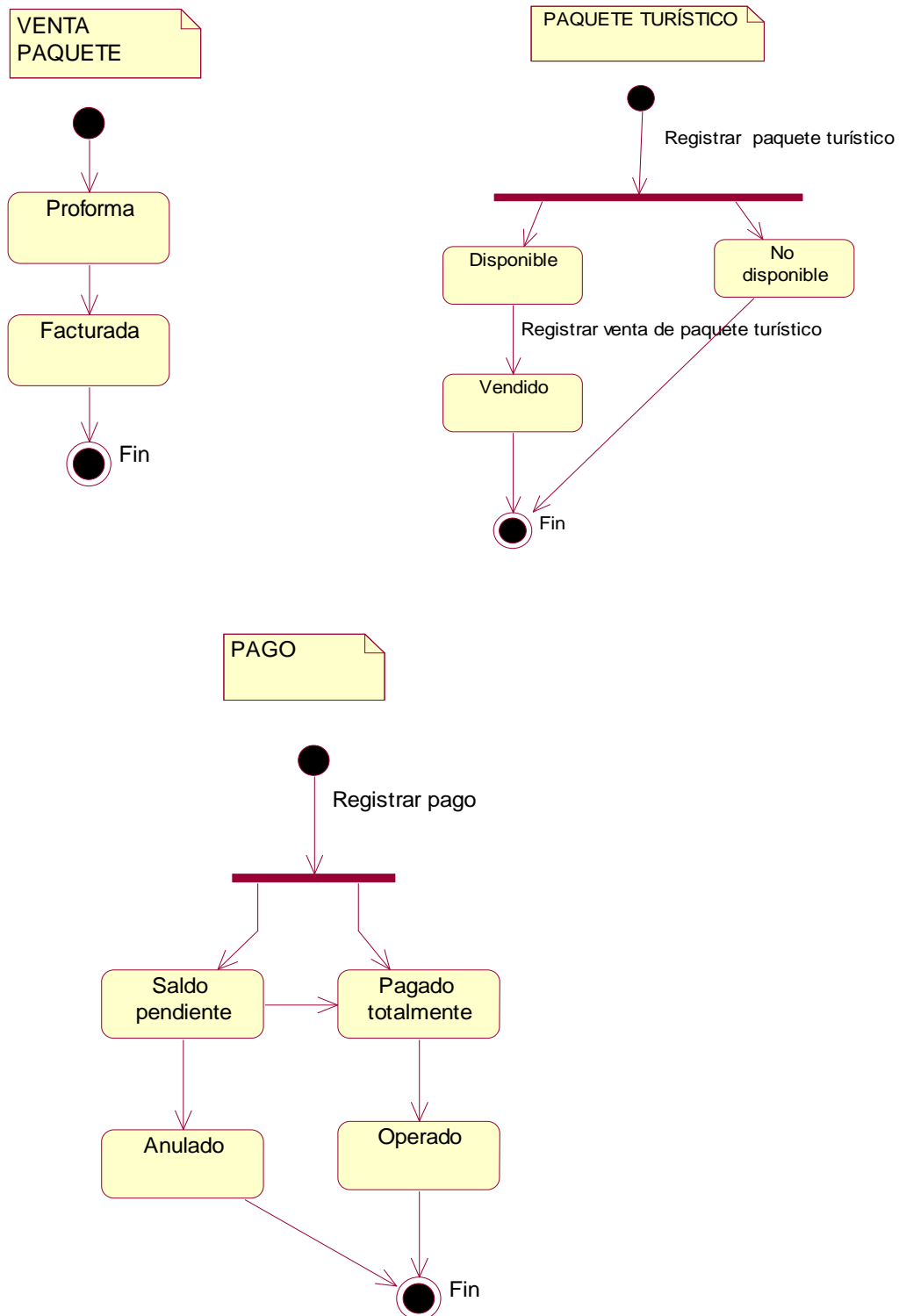


Operar paquete turístico



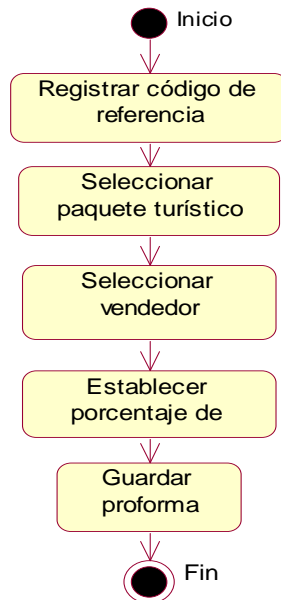
2.2 MODELO FUNCIONAL

2.2.1 Diagrama de Estado

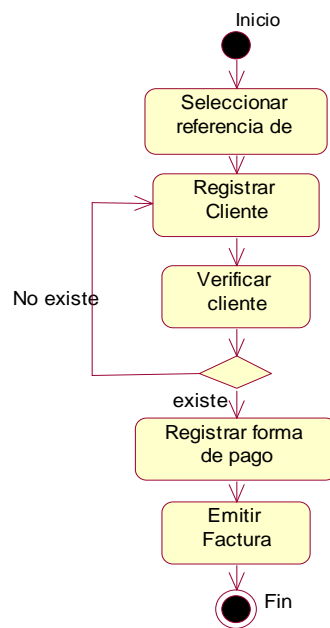


2.2.2 Diagrama de Actividades

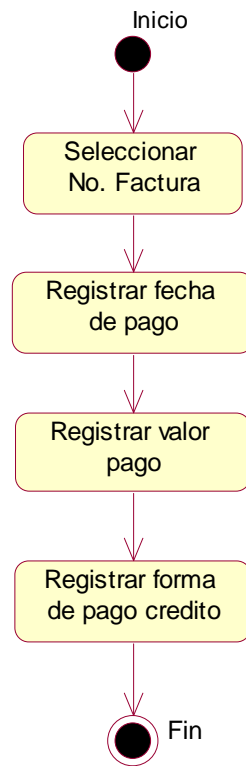
VENTA DE PAQUETES TURÍSTICOS



CREAR FACTURA



REGISTRAR PAGO CRÉDITO

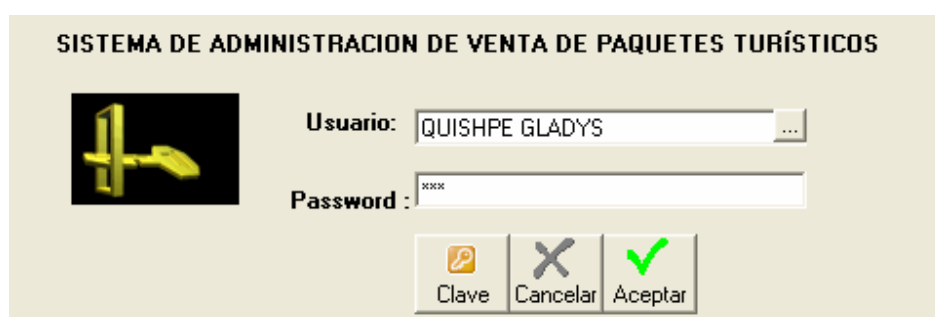


MANUAL DE USUARIO

1 DESCRIPCIÓN GENERAL

1.1 ACCESO AL SISTEMA

La interfaz de acceso al sistema nos permite ingresar como usuarios mediante la clave que pertenece a cada usuario.



The screenshot shows a login window titled "SISTEMA DE ADMINISTRACION DE VENTA DE PAQUETES TURÍSTICOS". On the left is a logo of a yellow key. To the right, there are two input fields: "Usuario:" containing the text "QUISHPE GLADYS" and "Password:" containing "xxx". Below the fields are three buttons: "Clave" with a key icon, "Cancelar" with a red X icon, and "Aceptar" with a green checkmark icon.

Figura 1. Interfaz acceso al sistema

1.2 MENU DE ACTIVIDADES

El menú de actividades es el más importante del Sistema, con la ayuda de este menú podemos conocer las diferentes actividades que se puede realizar en la administración de venta de paquetes turísticos.



Figura 2. Menú de actividades

1.1.2 DESCRIPCIÓN DEL MENÚ DE ACTIVIDADES

1.2.1.1 Menú Archivo

Contiene los submenús de Control de asistencia de vendedores y Guías Turísticos y la opción de salir.

Control de Asistencia

Permite registrar la asistencia diaria de vendedores y guías turísticos, escogiendo el usuario y luego digitando su número de cédula como contraseña, además podemos emitir reportes diarios para llevar un control de esta actividad.

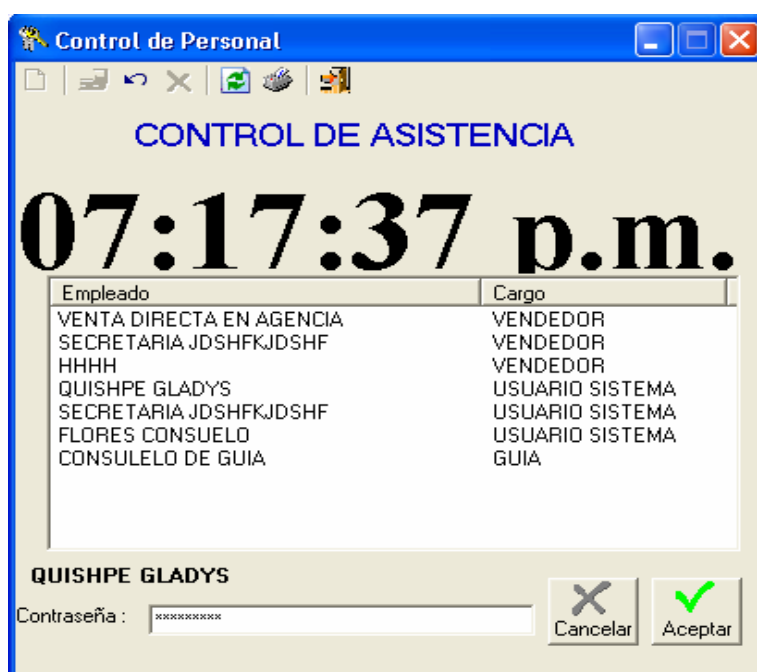


Figura 3. Control de Asistencia

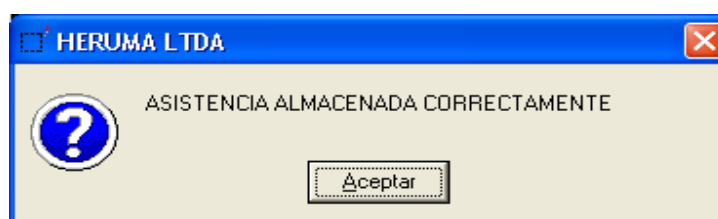


Figura 4. Mensaje de registro correcto

1.2.1.2 Menú Administración

El menú Administración es un elemento importante permite la administración de los diferentes elementos que todo el sistema.

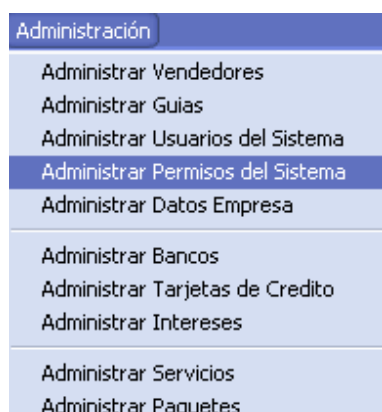


Figura 5. Menú Administración.

BOTONES UTILIZADOS EN EL INGRESO DE INFORMACIÓN

Los siguientes botones que se encuentran en todos los formularios de ingreso de información realizan la misma operación:



Permite el ingreso de un nuevo registro.



Permite guardar la información ingresada en el formulario.

Es necesario llenar todos los campos.



Permite cancelar un ingreso.



Permite eliminar un registro.



Refresca el formulario.



Permite la impresión los registros activos del formulario.



Permite salir del formulario.

VALIDACIÓN DEL CAMPO CÉDULA

La cédula de identidad es uno de los parámetros más importantes en el registro de la información a través de la cédula podremos acceder a toda la información de los diferentes personas que interactúan con la agencia como son clientes, vendedores, guías turísticos y usuarios del sistema.

En este campo se debe ingresar solo números, no debe quedar en blanco, permite ingresar números de cédula erróneos pero al momento de guardar la información nos da la opción de poder cancelar esto es súper importante porque la agencia trabaja con clientes que podrían ser niños y no tener numero de cédula.

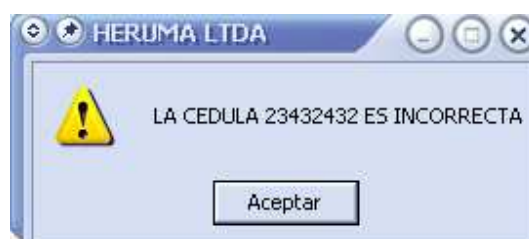


Figura 6. Mensaje de validación del número de cédula.

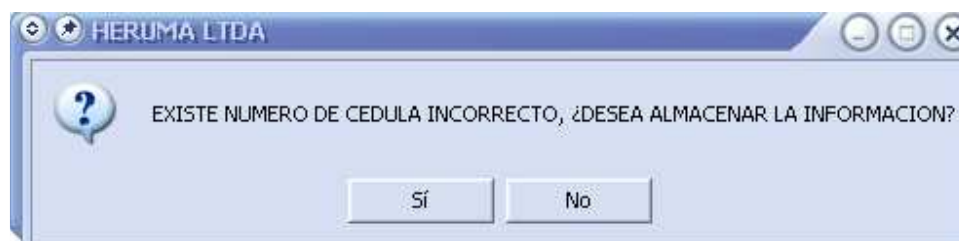


Figura 7. Mensaje al querer guardar la información con número de cédula erróneo.

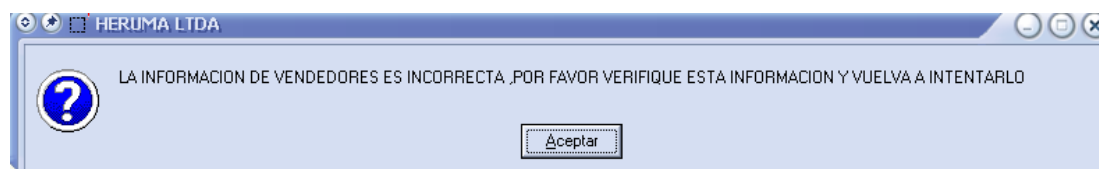
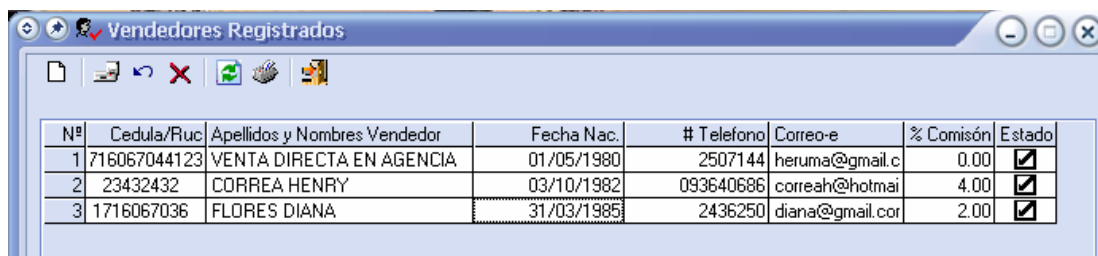


Figura 8. Mensaje de campo en blanco

Administrar vendedores.

Al ingresar a este submenú observamos el siguiente formulario:



Nº	Cedula/Ruc	Apellidos y Nombres Vendedor	Fecha Nac.	# Telefono	Correo-e	% Comisión	Estado
1	716067044123	VENTA DIRECTA EN AGENCIA	01/05/1980	2507144	heruma@gmail.c	0.00	<input checked="" type="checkbox"/>
2	23432432	CORREA HENRY	03/10/1982	093640686	correah@hotmail	4.00	<input checked="" type="checkbox"/>
3	1716067036	FLORES DIANA	31/03/1985	2436250	diana@gmail.com	2.00	<input checked="" type="checkbox"/>

Figura 9. Formulario de ingreso de información de Vendedores

Esta opción nos permite codificar la información más importante y necesaria de vendedores que trabajan en la Agencia.

En el ingreso de información de vendedores se deben tomar en cuenta los siguientes puntos.

Cedula / Ruc.- En este campo se debe ingresar el número de cédula del vendedor.

Apellidos y Nombres Vendedor.- Este campo permite ingresar solo letras mayúsculas.

Fecha Nac.- Se ingresa la fecha de nacimiento del vendedor, este campo permite validar que los vendedores sean mayores de 18 años de edad.



Figura 10. Mensaje de validación que los vendedores sean mayores de edad.

Teléfono.- Permite ingresar solo números.

% Comisión.- Se ingresa el porcentaje de comisión que el vendedor va a ganar por cada una de las ventas que realiza.

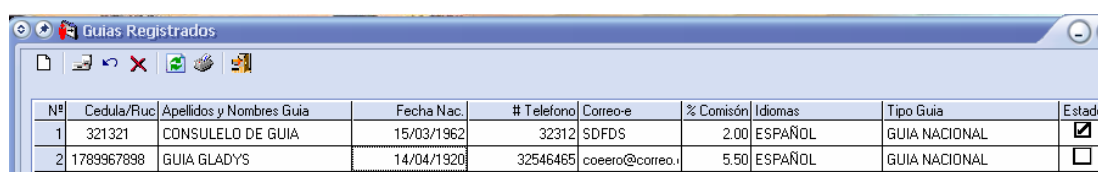
Estado.- Indica el estado del Vendedor.

Activo.- Trabaja actualmente en la Agencia.

Pasivo.- Ya no trabaja en la Agencia.

Administrar Guías.

Esta opción muestra el siguiente formulario:



N°	Cedula/Ruc	Apellidos y Nombres Guia	Fecha Nac.	# Telefono	Correo-e	% Comisión	Idiomas	Tipo Guia	Estado
1	321321	CONSULELO DE GUIA	15/03/1962	32312	SDFDS	2.00	ESPAÑOL	GUIA NACIONAL	<input checked="" type="checkbox"/>
2	1789967898	GUIA GLADYS	14/04/1920	32546465	coeero@correo.	5.50	ESPAÑOL	GUIA NACIONAL	<input type="checkbox"/>

Figura 11. Formulario de ingreso de información de Guías Turísticas.

Cedula / Ruc.- En este campo se debe ingresar el número de cédula del guía Turístico.

Apellidos y Nombres Vendedor.- Este campo permite ingresar solo letras mayúsculas.

Fecha Nac.- Se ingresa la fecha de nacimiento del guía turístico, este campo permite validar que los Guías sean mayores de 18 años de edad.



Figura 12. Mensaje de validación que los Guías sean mayores de edad.

Teléfono.- Permite ingresar solo números.

% Comisión.- Se ingresa el porcentaje de comisión que el guía va a ganar por cada uno de los tours realizados.

Idiomas.- Permite ingresar el tipo de idioma que tienen los Guías turísticos.

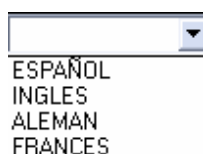


Figura 13. Elegir idioma de Guías Turísticas.

Tipo Guía.- Permite ingresar ámbito en el que se desenvuelven los Guías Turísticos.

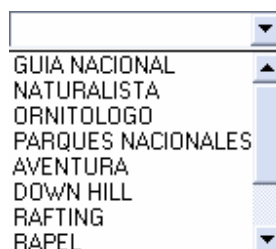


Figura 14. Elegir tipo de Guías Turísticas.

Estado.- Indica el estado de Guías turísticas.



Activo.- Trabaja actualmente en la Agencia.



Pasivo.- Ya no trabaja en la Agencia.

Administrar Usuarios del Sistema.

Este formulario permite ingresar los usuarios que tendrán acceso al sistema con sus diferentes permisos.

Al ingresar a esta opción aparecerá el siguiente formulario:

Nº	Cedula/Ruc	Apellidos y Nombres Usuario del Sistema	Fecha Nac.	# Telefono	Correo-e	Clave	Grupo	Estado
1	1716067044	QUISHPE GLADYS	15/08/1924	3555555	correo@correo.com	147149172	ADMINISTRADOR	<input checked="" type="checkbox"/>
2	1716067044	FLORES CONSUELO	29/08/1949	666666666	cooreo@correo.com	147149172	ADMINISTRADOR	<input checked="" type="checkbox"/>
3	1716067036	PRUEBAS	23/09/1981	99998222	sineamil@email.com	195197220	ADMINISTRADOR	<input type="checkbox"/>

Figura 15. Formulario Usuarios del sistema.

Cedula / Ruc.- En este campo se debe ingresar el número de cédula del usuario del sistema.

Apellidos y Nombres Usuario del sistema.- Este campo permite ingresar solo letras mayúsculas.

Fecha Nac.- Se ingresa la fecha de nacimiento del usuario del sistema, este campo permite validar que los usuarios sean mayores de 18 años de edad.

Teléfono.- Permite ingresar solo números.

Clave.- Permite ingresar la contraseña con la que el usuario va poder ingresar al sistema.

Grupo.- Seleccionar al grupo al que va pertenecer el usuario

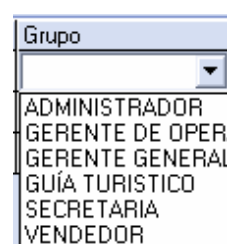


Figura 16. Lista de grupo de acceso de usuarios al sistema.

Estado.- Indica el estado de Usuarios del sistema.



Activo.- El usuario aun tiene acceso al sistema.



Pasivo.- El usuario no tiene acceso al sistema.

Administrar Permisos del Sistema.

Este submenú es muy importante en este asignamos las opciones del sistema a las que van a tener acceso los usuarios, dependiendo del grupo de acceso al que pertenecen los usuarios. Al escoger esta opción aparecerá al siguiente formulario:

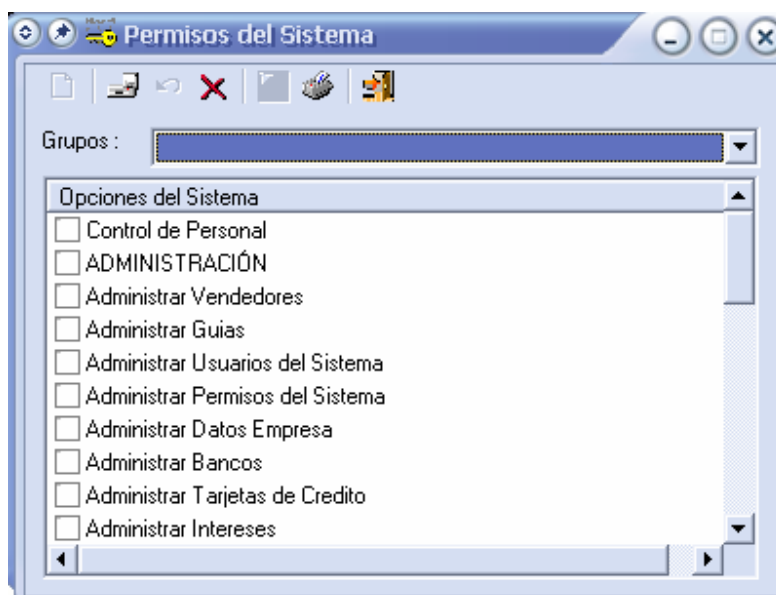


Figura 17. Formulario administrar permisos del sistema.

Grupos.- Permite seleccionar el grupo de acceso.

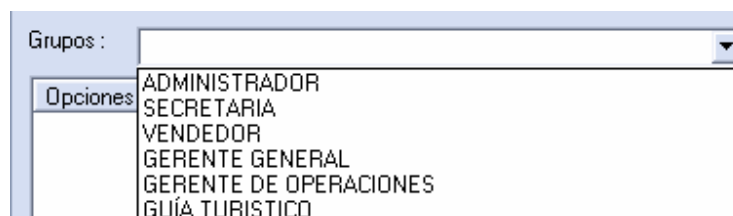


Figura 18. Despliegue de los grupos de acceso al sistema.

Opciones del sistema.- Se elige las opciones a las que cada grupo de acceso va tener control.

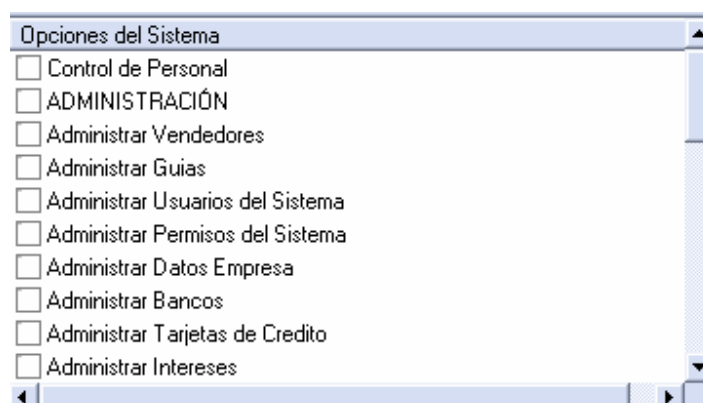


Figura 19. Opciones de Acceso.

Administrar Datos de la Empresa.

Esta opción permite ingresar la información de la Empresa, al elegir esta opción aparecerá el siguiente formulario:

 A screenshot of a software window titled "Datos de Empresa". The form contains the following fields:

- Nombre: HERUMA LTDA.
- Dirección: CALLE COLÓN Y DIEGO DE ALMAGRO, CENTRO COMERCIAL ALCAZ
- Telefono: 2436250
- email: heruma@gmail
- I.V.A.: 12.00
- Tasa de Servicios: 10.00

Figura 20. Formulario de ingreso de datos de la Empresa.

IVA.- Permite ingresar la tasa de porcentaje del IVA vigente, con el que se trabajara en la venta de paquetes turísticos.(12%).

Tasa de servicios.- Permite ingresar la tasa de servicio con la que trabajará la agencia.

Administrar Bancos.

Permite ingresar los datos de los bancos con los que trabajará la Agencia.

Al seleccionar está opción aparecerá este formulario.



Nº	Nombre del Banco	Estado
1	BANCO DEL PACIFICO	<input type="checkbox"/>
2	BANCO DEL PICHINCHA	<input checked="" type="checkbox"/>

Figura 21. Formulario de administrar bancos.

Nombre del Banco.- Permite ingresar solo letras y mayúsculas.

Estado.- Indica el estado de los bancos.



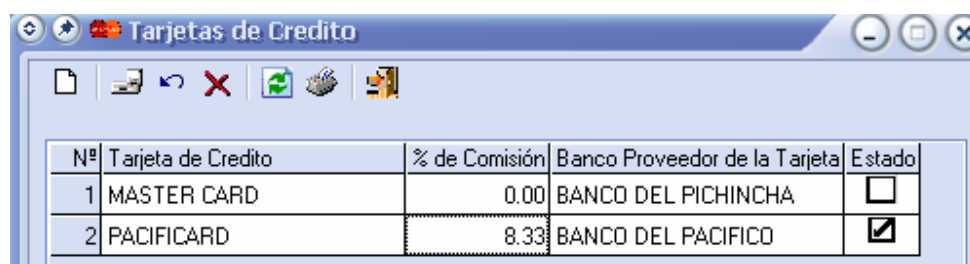
Activo.- La agencia trabaja con esos bancos.



Pasivo.- La agencia trabaja ya no trabaja con esos bancos.

Administrar Tarjetas de crédito.

Esta opción permite ingresar datos referentes a la tarjeta de crédito que el cliente puede presentar en el pago de una venta de paquete turístico. Al elegir esta opción se observará el siguiente formulario.



Nº	Tarjeta de Credito	% de Comisión	Banco Proveedor de la Tarjeta	Estado
1	MASTER CARD	0.00	BANCO DEL PICHINCHA	<input type="checkbox"/>
2	PACIFICARD	8.33	BANCO DEL PACIFICO	<input checked="" type="checkbox"/>

Figura 22. Formulario de Administrar tarjetas de crédito.

Tarjeta de crédito.- Permite registrar el nombre de la tarjeta, este campo solo acepta letras mayúsculas.

% de Comisión.- Permite ingresar el porcentaje de comisión por la tarjeta, esto depende del banco a quién se le va cargar.

Banco proveedor de la tarjeta.- Permite seleccionar el nombre del banco al que pertenece la tarjeta únicamente despliega los bancos que estén activos.

Banco Proveedor de la Tarjeta

BANCO DEL PICHINCHA

Figura 23. Bancos activos proveedores de tarjetas de crédito.

Estado.- Indica el estado de los bancos.



Activo.- La agencia trabaja con ese tipo de tarjeta.



Pasivo.- La agencia trabaja no trabaja con ese tipo de tarjeta.

Administrar Intereses de tarjetas de crédito.

En este submenú se registra los diferentes porcentajes de interés con las que trabaja las diferentes tarjetas de crédito, estos porcentajes depende de cada banco proveedor de las tarjetas.

Al seleccionar esta opción tendremos el siguiente formulario.

Intereses de Tarjetas de Credito

PACIFICARD

N°	# Meses	% Interés	Estado
1		0.00	<input type="checkbox"/>

Figura 24. Formulario de Intereses de tarjetas de crédito.

Meses.- Se ingresa el número de meses que se requiere el crédito.

% Interés.- Permite registrar el porcentaje de interés que se le cobrará de acuerdo al tiempo del crédito.

Estado.- Indica el estado de los las tasas de interés de las tarjetas de crédito.



Activo.- La agencia trabaja con ese porcentaje de interés en el tiempo indicado.



Pasivo.- La agencia no trabaja con ese porcentaje de interés en el tiempo.

Administrar Servicios

Esta opción permite ingresar los diferentes tipos de servicios y servicios que ofrece la Agencia en cada uno de los paquetes turísticos, los precios que se ingresan son individuales por persona. Al elegir esta opción tendremos el siguiente formulario:

N° Servicio	Ciudad	1 -> 2 Personas	3 -> 8 Personas	9 -> 18 Personas	19 -> 30 Personas	Estado
1		0.00	0.00	0.00	0.00	<input type="checkbox"/>

Figura 25. Formulario de Administración de Servicios.

.- Se ingresa el nombre del tipo de servicio que queremos ingresar, si intentamos guardar con esté campo en blanco tendremos el siguiente mensaje.

Figura 26. Mensaje si el campo esta en blanco.

COMIDA
GUIA
HOTEL
SITIOS TURISTICOS
TICKET AEREOS
TRANSPORTE

.- Se visualiza el nombre de los servicios existentes.

Al seleccionar un servicio observaremos de manera detallada que incluye cada tipo de servicio.

Nº	Servicio	Ciudad	1 -> 2 Personas	3 -> 8 Personas	9 -> 18 Personas	19 -> 30 Personas	Estado
1	Comida Hotel 2 Estrellas	Quito	4.00	2.50	2.00	1.50	<input checked="" type="checkbox"/>
2	Comida en Marisqueria	Manta	3.00	2.80	2.50	2.00	<input checked="" type="checkbox"/>
3	Comida por un Dia	Manta	9.00	8.00	7.00	6.00	<input checked="" type="checkbox"/>
4	nueva comida	Guayaquil	5.00	4.00	3.00	2.00	<input checked="" type="checkbox"/>

Figura 27. Despliegue de lo que ofrece el servicio COMIDA.

Si deseamos borrar o editar un servicio existente damos un clic derecho y elegimos la opción deseada.

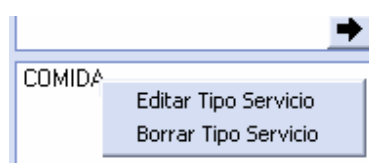



Figura 28. Opciones editar y borrar tipo de servicio.

La información requerida para el ingreso de un nuevo servicio es la siguiente:

Servicio.- Se ingresa el nombre del servicio que ofrece el tipo de servicio.

Ciudad.- Seleccionar la ciudad en la que se va dar ese servicio.

En los demás casilleros se registra el valor del servicio de acuerdo al número de personas que pueden solicitar al servicio, el valor ingresado será individual por persona.

Estado.- Indica el estado de los servicios componen los paquetes turísticos. 

Activo.- La agencia ofrece el servicio.



Pasivo.- El servicio no esta disponible.

Administrar Paquetes Turísticos

En esta opción permite asignar los servicios que componen cada paquete turístico, indicando el valor del paquete por número de personas y el valor por persona.

Al elegir esta opción tendremos el siguiente formulario:

Nº	Paquete	# Días	1 -> 2 Personas	3 -> 8 Personas	9 -> 18 Personas	19 -> 30 Personas	Estado
1	TOUR AVENTURA	1	112.00	110.80	109.50	108.00	<input type="checkbox"/>
2	TOUR VISITAS	2	24.00	24.00	24.00	24.00	<input checked="" type="checkbox"/>
3	VISTA DE PARQUES	3	128.00	124.50	123.00	119.50	<input checked="" type="checkbox"/>

Nº	Tipo Servicio	Servicio	Ciudad	1 -> 2 Personas	3 -> 8 Personas	9 -> 18 Personas	19 -> 30 Personas	Estado
1	TRANSPORTE	Furgoneta por 1 Dia	Cuenca	30.00	25.00	22.00	12.00	<input type="checkbox"/>
2	COMIDA	Comida Hotel 2 Estrellas	Quito	4.00	2.50	2.00	1.50	<input type="checkbox"/>
3		Comida en Marisquería	Manta	3.00	2.80	2.50	2.00	<input type="checkbox"/>
4		Comida por un Dia	Manta	9.00	8.00	7.00	6.00	<input type="checkbox"/>
5		nueva comida	Guayaquil	5.00	4.00	3.00	2.00	<input type="checkbox"/>
6	TICKET AEREOS	Quito-Cuenca	Quito	100.00	100.00	100.00	100.00	<input type="checkbox"/>
7	YA NO ES PRUET	Servicio de Pruebas	Cuenca	12.00	10.00	9.00	6.00	<input type="checkbox"/>
8	SITIOS TURISTIC	PARQUE NACIONAL YASUNI	Quito	12.00	12.00	12.00	12.00	<input checked="" type="checkbox"/>
9		PARQUE NACIONAL COTOPAXI	Quito	12.00	12.00	12.00	12.00	<input checked="" type="checkbox"/>

Figura 29. Formulario de Administración de paquetes turísticos.

Paquete.- Permite ingresar el nombre del paquete turístico, únicamente letras mayúsculas.

Días.- Permite ingresar el número de días que durará ese tour.

El resto de campos se ingresan automáticamente dependiendo del servicio que ofrece el paquete turístico.

Estado.- Indica el estado de los Paquetes turísticos que ofrece la Agencia.



Activo.- El paquete esta disponible.



Pasivo.- El paquete no esta disponibles.

Servicios paquetes.- Despliega todos los servicios que ofrece la agencia y que conforman los diferentes paquetes turísticos.

Servicios Paquetes								
Nº	Tipo Servicio	Servicio	Ciudad	1 -> 2 Personas	3 -> 8 Personas	9 -> 18 Personas	19 -> 30 Personas	Estado
1	TRANSPORTE	Furgoneta por 1 Dia	Cuenca	30.00	25.00	22.00	12.00	<input type="checkbox"/>
2	COMIDA	Comida Hotel 2 Estrellas	Quito	4.00	2.50	2.00	1.50	<input type="checkbox"/>
3		Comida en Marisqueria	Manta	3.00	2.80	2.50	2.00	<input type="checkbox"/>
4		Comida por un Dia	Manta	9.00	8.00	7.00	6.00	<input type="checkbox"/>
5		nueva comida	Guayaquil	5.00	4.00	3.00	2.00	<input type="checkbox"/>
6	TICKET AEREOS	Quito-Cuenca	Quito	100.00	100.00	100.00	100.00	<input type="checkbox"/>
7	YA NO ES PRUEB	Servicio de Pruebas	Cuenca	12.00	10.00	9.00	6.00	<input type="checkbox"/>
8	SITIOS TURISTIC	PARQUE NACIONAL YASUNI	Quito	12.00	12.00	12.00	12.00	<input checked="" type="checkbox"/>
9		PARQUE NACIONAL COTOPAXI	Quito	12.00	12.00	12.00	12.00	<input checked="" type="checkbox"/>

Figura 30. Servicios que conforman los diferentes paquetes turísticos.

Estado.- Indica si el servicio esta asignado al paquete



Activo.- El servicio esta asignado al paquete



Pasivo.- El servicio no esta asignado al paquete turístico.

Para asignar un servicio se debe escoger o ingresar el paquete turísticos al que va ha pertenecer el servicio de lo contrario aparecerá el siguiente mensaje.



Figura 31. Mensaje al no escoger paquete turístico para asignar servicio.

1.2.1.2 Menú Paquetes turísticos

En esta parte del menú se encuentran los siguientes submenús:

Venta de paquetes turísticos

Se crea la proforma de venta de paquete turístico, se registra el código de referencia de venta, la misma que está formada por el primer apellido del tour líder, el carácter x y el número de personas que van a conformar el tour, luego se registra el número de personas que integrarán el tour, la fecha que se va ha realizar el tour, se procede a escoger el paquete turístico que esté creado de acuerdo a los requerimientos del cliente, escogemos el vendedor, establecemos el porcentaje de ganancia de la empresa, la fecha límite se crea automáticamente en caso de tratarse de una venta a crédito esta debe ser cancelada 15 días antes de la fecha que se estableció para

realizarse el tour, en observaciones se registran observaciones que se requieran de acuerdo a la venta, a continuación se despliegan los valores detallados correspondientes a la venta que se a creado, precio por persona se refiere al valor que esta registrado de acuerdo al paquete que se elige para la venta, P.V.P se refiere al valor del precio por persona multiplicado por el número de personas que se registran al tour, tasa de Servicios es el valor ya calculado del P.V.P. de acuerdo a la tasa de servicios con que opere la empresa, el valor ganancia se calcula de acuerdo al porcentaje de ganancia que se estableció anteriormente, el sub-total corresponde a la suma de los valores calculados anteriormente, se calcula de este valor el 12% de IVA, se calcula el precio total que el cliente debe paga y finalmente se generan 2 reportes, la que corresponde a la empresa y la que se debe entregar al cliente.

Servicio	Ciudad	Tipo Servicio
Comida Hotel 2 Estrellas	Quito	COMIDA
Quito-Cuenca	Quito	TICKET AEREDS
Servicio de Pruebas	Cuenca	YA NO ES PRUEBAS
PARQUE NACIONAL YASUNI	Quito	SITIOS TURISTICOS ...

Precio por Persona :	124.50
P.V.P :	373.50
Tasa Servicios :	37.35
Valor Ganancia :	11.20
Sub-Total :	422.05
I.V.A :	50.65
Precio Total :	472.70

Figura 32. Generación de proforma antes de realizar la venta

Emitir Factura

Si el cliente decide adquirir el paquete turístico se procede a emitir la factura luego de que este cancele el valor correspondiente, en este proceso se debe escoger la proforma que ya existe.

Buscar Proformas

Fecha de Venta Desde: 08/08/2006 Hasta: 07/09/2006
 Fecha del Tour

Buscar

Cod. Referencia Paquete	# Personas	Fecha Tour	Precio Total
PX4	4	02/09/2006	96
PRUEBAX2	2	06/09/2006	48
PPX2	2	06/09/2006	48
FLORESX2	2	06/09/2006	256
ESPINX3	3	07/10/2006	373.5

✓

Figura 33. Buscar proforma.

Luego de escoger la proforma se procede a crear la factura escogiendo el cliente si existe caso contrario registrando el cliente al que se le va a dirigir la factura, se registra el valor escogiendo el tipo de pago, si el pago es en efectivo solo se registra el valor, si el pago es con cheque se registra el valor luego la información del cheque, banco, titular, No. Cheque y fecha.

Emitir Factura

Factura :

Cod. Referencia Venta: ESPINX3 Dirección Cliente: CALDERON
 Ruc/CC Cliente: 1713734489 Telefono: 098235648
 Nombre Cliente: RUBEN ESPIN Valor Total: **472.70**

Efectivo: 0
 Cheque: 472.7

Información de Pago con Cheque
 Bancos: Banco de Pruebas # Cheque: 45454
 Titular Cheque: MARCOS ESPIN Fecha: 07/09/2006

Información de Pago con Credito Directo
 Meses: Total a Pagar:

Información de Pago con Tarjeta de Credito
 Tarjetas: Meses:
 # de Tarjeta: T. Pagar:
 Titular Tarjeta: # Voucher:
 # Auto:

Credito Directo: 0
 Tarjeta de Crédito: 0

Figura 34. Interfaz para generar factura.

Luego tendremos como resultado el reporte para la empresa y la factura que se le entregará al cliente detallando las actividades que corresponden al pago:

Jueves, 07 de Septiembre de 2006

Pago de la Factura # : 7

Pago en Cheque : **472.70**

Banco :
Banco de Pruebas

Titular Cheque :
MARCOS ESPIN

Numero Cheque :
45454

Saldo del Credito :

Figura 35. Documento para la empresa.

Jueves, 07 de Septiembre de 2006

7

Ruc/CC : 1713734489 Dirección : CALDERON

Nombre : RUBEN ESPIN Fecha Factura : 07/09/2006

3 Servicios de Tour :	140.68	422.05
ESPINX3		
PAQUETE 1		
	Sub-Total :	422.05
	Iva	50.65
	Total Factura	472.70

Figura 36. Factura para el cliente.

Proforma para pago a crédito personal

Para el caso de pago a crédito personal se llena de la misma manera que la proforma anterior pero con seguridad variará la fecha de Tour así como la fecha límite que es de 15 días antes del tour.

Servicio	Ciudad	Tipo Servicio
Comida Hotel 2 Estrellas	Quito	COMIDA
Quito-Cuenca	Quito	TICKET AEREOS
Servicio de Pruebas	Cuenca	YA NO ES PRUEBAS
PARQUE NACIONAL YASUNI	Quito	SITIOS TURISTICOS ...

Precio por Persona : 124.50
P.V.P : 622.50
Tasa Servicios : 62.25
Valor Ganancia : 31.12
Sub-Total : 715.87
I.V.A : 85.90
Precio Total : 801.78

Figura 37. Proforma pago crédito directo.

Cod. Venta :	12	Ref :	CORREAX5	Uso Interno Agencia	
# Personas :	5				
Paquete :	PAQUETE 1	Numero de Dias :	3		
Vendedor :	VENTA DIRECTA EN AGENCIA	% Comisión Vendedor :	0.00	Valor Comisión :	0.00
Precio Paquete :	622.50	Obs :	La fecha del tour se mantendrá sin alterar		
Valor Iva :	85.90				
Tasa Servicios :	62.25				
Valor Ganancia :	31.12				
Fecha Venta :	07/09/2006				
Fecha Limite Pago :	08/12/2006			Precio Paquete :	801.77

Figura 38. Documento para la empresa.

Cod. Venta :	12	Ref :	CORREAX5	Cliente Final
# Personas :	5	Fecha Venta :	23/12/2006	
Paquete :	PAQUETE 1	Numero de Dias :	3	
Vendedor :	VENTA DIRECTA EN AGENCIA			
Fecha Venta :	07/09/2006			
Fecha Limite Pago :	08/12/2006	Precio Paquete :	801.77	
Actividades Paquete				
COMIDA	Comida Hotel 2 Estrellas			
SITIOS TURISTICO	PARQUE NACIONAL YASUNI			
TICKET AEREOS	Quito-Cuenca			
YA NO ES PRUEBA	Servicio de Pruebas			

Figura 39. Proforma para el cliente.

Factura para pago a crédito personal

El procedimiento es igual a la factura anterior con la diferencia de que en este caso se debe escoger para cuantos meses se concede el crédito.

Emitir Factura Factura :

Cod. Referencia Venta : CORREAX5 Dirección Cliente : LA TOLA
Ruc/CC Cliente : 1703531754 Telefono : 2568945
Nombre Cliente : ERNESTO CORREA Valor Total : **801.77**

Efectivo : 100
Cheque : 0

Información de Pago con Cheque:
Bancos : [] # Cheque : []
Titular Cheque : [] Fecha : 07/09/2006

Credito Directo : 701.77
Información de Pago con Credito Directo:
Meses : **3 Meses - 3%** Total a Pagar : **722.82**

Tarjeta de Crédito : []
Información de Pago con Tarjeta de Credito:
Tarjetas : [] Meses : []
de Tarjeta : [] T. Pagar : []
Titular Tarjeta : [] # Voucher : []
Auto. : []

Figura 40. Factura para pago a crédito personal.

En el reporte que se crea se detalla las fechas en que debe cancelarse y el valor de las cuotas mensuales, así como el valor pagado.

Jueves, 07 de Septiembre de 2006

Cod. Referencia : CORREAX5
 No Factura : **8**

No	Valor Cuota	Fecha a Pagar Cuota	Fecha Pago Cuota	Estado
1	240.94	07/10/2006		PENDIENTE
2	240.94	06/11/2006		PENDIENTE
3	240.94	06/12/2006		PENDIENTE

Valor Pagado :
 Saldo : **722.82**

Figura 41. Detalle de pagos pendientes.

Jueves, 07 de Septiembre de 2006

Pago de la Factura # : **8**

Pago en Efectivo **100.00**

Saldo de la Factura :

Figura 42. Detalle del pago

Jueves, 07 de Septiembre de 2006

8

Ruc/CC : 1703531754 Dirección : LA TOLA
 Nombre : ERNESTO CORREA Fecha Factura : 07/09/2006

5 Servicios de Tour : CORREAX5 PAQUETE 1	143.17	715.87
	Sub-Total :	715.87
	Iva	85.90
	Total Factura	801.77

Figura 43. Factura para cliente.

Proforma pago con tarjeta

Se crea una proforma igual a las anteriores.

Venta de Paquetes Turísticos [Generación de Proforma antes de Realizar la Venta]

Cod. Referencia Venta : CHICAIZAX3

Numero de Personas : 3 Fecha Tour : 30/11/2006

Paquete : PAQUETE 1

Vendedor : VENTA DIRECTA EN AGENCIA

% Ganancia : 6 Fecha Limite : 15/11/2006

Obs : La fecha límite del pago se mantiene como se a determinad

Servicio	Ciudad	Tipo Servicio
Comida Hotel 2 Estrellas	Quito	COMIDA
Quito-Cuenca	Quito	TICKET AEREOS
Servicio de Pruebas	Cuenca	YA NO ES PRUEBAS
PARQUE NACIONAL YASUNI	Quito	SITIOS TURISTICOS ...

Precio por Persona : 124.50
P.V.P : 373.50
Tasa Servicios : 37.35
Valor Ganancia : 22.41

Sub-Total : 433.26
I.V.A : 51.99

Precio Total : 485.25

Figura 44. Proforma pago con tarjeta.

Factura para pago con tarjeta

Se especifica el detalle de la primera cuota y para el saldo se debe seleccionar la tarjeta con la que paga el plazo, No. Tarjeta, Titular de la Tarjeta, No. Voucher y No. Autorización.

Emitir Factura

Factura :

Cod. Referencia Venta : CHICAIZAX3 Dirección Cliente : SANGOLQUI

Ruc/CC Cliente : 1713734459 Telefono : 2897456

Nombre Cliente : MAGALI CHICAIZA Valor Total : 485.25

Efectivo : 150

Cheque : 0

Credito Directo : 0

Tarjeta de Crédito : 335.25

Información de Pago con Cheque

Bancos : # Cheque :

Titular Cheque : Fecha : 07/09/2006

Información de Pago con Credito Directo

Meses : Total a Pagar :

Información de Pago con Tarjeta de Credito

Tarjetas : Tarjeta de Pruebas Meses : 3 Meses - 6%

de Tarjeta : 154545 T. Pagar : 355.36

Titular Tarjeta : MAGALI CHICAIZA # Voucher : 1254

Auto. : 225566

Figura 45. Factura para pago con tarjeta.

A continuación tendremos como resultado el detalle del valor pagado con tarjeta y los datos de la misma.

Jueves, 07 de Septiembre de 2006	
Pago de la Factura # :	9
Pago en Tarjeta :	335.25
Numero Tarjeta :	154545
Titular Tarjeta :	MAGALI CHICAIZA
Numero Meses :	3
% Interes :	6.00
Numero de Voucher :	1254
Saldo de la Factura	

Figura 46. Detalle de pago con tarjeta.

Se genera también un reporte del valor que se pagó como cuota inicial.

Jueves, 07 de Septiembre de 2006	
Pago de la Factura # :	9
Pago en Efectivo	150.00
Saldo de la Factura :	

Figura 47. Detalle de pago

Finalmente tenemos la factura del que se le entrega al cliente.

		Jueves, 07 de Septiembre de 2006	
		9	
Ruc/CC :	1713734459	Dirección :	SANGOLQUI
Nombre :	MAGALI CHICAIZA	Fecha Factura :	07/09/2006
3 Servicios de Tour :		144.42	433.26
CHICAIZAX3			
PAQUETE 1			
	Sub-Total :		433.26
	Iva		51.99
	Total Factura		485.25

Figura 48. Factura para cliente.

Registro de pagos cuotas crédito

Primeramente se procede a buscar la factura a la que pertenece el pago que se va a registrar, luego se debe escoger la cuota que va a ser cancelada dando doble clic en la cuota correspondiente, registrar la fecha en la que se está recibiendo el pago y poner en estado activo, seguidamente se debe registrar el valor de acuerdo a la forma de pago sea en efectivo, cheque o tarjeta de crédito.

Pago Cuotas Credito

Nº Factura : 8

Nº	Valor Cuota	Fecha a Pagar	Fecha Pagada Cuota	Estado
1	240.94	07/10/2006	07/09/2006	<input checked="" type="checkbox"/>
2	240.94	06/11/2006		<input type="checkbox"/>
3	240.94	06/12/2006		<input type="checkbox"/>

Valor Total Credito : 722.82
 Valor Pagado : 240.94
 Saldo : 481.88

Cod Referencia : CORREAX5
 Paquete : PAQUETE 1
 Fecha Max. Pago : 08/12/2006
 Vendedor : VENTA DIRECTA EN AGENCIA
 Fecha de Venta : 07/09/2006
 Fecha Tour : 23/12/2006

Efectivo : 240.94
 Cheque :
 Tarjeta de Crédito :

Información de Pago con Cheque:
 Bancos :
 Titular Cheque :
 # Cheque :
 Fecha : 08/09/2006

Información de Pago con Tarjeta de Credito:
 Tarjetas :
 # de Tarjeta :
 Titular Tarjeta :
 Meses :
 T. Pagar :
 # Voucher :
 # Auto.:

Figura 49. Registro de pagos.

Se genera el reporte de forma de pago con su respectivo valor.

Jueves, 07 de Septiembre de 2006

Pago de la Factura # : 8

Pago en Efectivo 240.94

Saldo de la Factura : 481.88

Figura 50. Detalle de pago.

Operar Paquete Turístico

Aquí se procede a registrar a los clientes que integrarán el tour, para realizar esta actividad se debe registrar al primero sin necesidad de poner nuevo porque encontraremos el espacio disponible para hacerlo directamente, para ingresar otros se debe poner la opción de nuevo registro, **es muy importante llenar todos los campos requeridos**, en caso de que no se cuente con la información se debe poner algún comentario de manera que los espacios no queden en blanco a excepción de Observaciones en este campo se puede dejar sin llenar y finalmente se debe poner el estado que nos indique si pertenece o no al tour.

The screenshot shows the 'Operar Paquete' application window. At the top, there are fields for 'Cod. Referencia Venta: CHICAIZAX3', 'Fecha de Salida: 30/11/2006', 'No Paxs: 3', 'Guia del Tour: GUIA GLADYS', 'Fecha de Retorno: 03/12/2006', and 'Paquete: PAQUETE 1'. Below these fields are two tabs: 'Lista de Pasajeros' and 'Lista de Actividades'. The 'Lista de Pasajeros' tab is active, displaying a table with the following data:

Nº	Fecha Nac.	# Telefono	Correo-e	Dirección	Alergias	Telefono Emergencia Medica	Obs	Estado
1	05/09/1978	2564894	julialopez@hotmail.com	LA CORUÑA Y DIEGO DE VAZCO	NINGUNA	23689366	NINGUNA	<input checked="" type="checkbox"/>
2	12/08/1968	098451245	xavibast@yahoo.com	COLON Y DIEGO DE ALMAGRO	NINGUNA	095784512	NINGUNA	<input checked="" type="checkbox"/>
3	05/09/1976	098457845	angelica@yahoo.com	CARAPUNGO	NINGUNA	2247887		<input checked="" type="checkbox"/>

At the bottom of the window, there is a button labeled 'Boton Derecho para Asignar Tour Leader'.

Figura 51. Operar Paquete.

Designar tour líder.

Luego de haber registrado a todos los integrantes del tour se procede a escoger a la persona que será tour líder para esto hay que escoger al cliente de la lista que se presenta en la grilla dando doble clic luego se debe dar clic derecho y queda escogido como tour líder marcado de color azul.

The screenshot shows the 'Operar Paquete' application window, similar to Figure 51. The 'Lista de Pasajeros' tab is active, and the first row of the table is highlighted in blue, indicating it has been selected as the tour leader. The table data is as follows:

Nº	Cedula/Ruc	Apellidos Y Nombres Cliente	Fecha Nac.	# Telefono	Correo-e	Dirección	Alergias	Telefono Emergencia Med
1	454545454	LOPEZ CHARRO JULIA	05/09/1978	2564894	julialopez@hotmail.com	LA CORUÑA Y DIEGO DE VAZCO	NINGUNA	23689366
2	44887887	BASTIDAS PROAÑO XAVIER	12/08/1968	098451245	xavibast@yahoo.com	COLON Y DIEGO DE ALMAGRO	NINGUNA	095784512
3	445454455	SALAS CALDERON ANGELICA	05/09/1976	098457845	angelica@yahoo.com	CARAPUNGO	NINGUNA	2247887

At the bottom of the window, there is a button labeled 'Boton Derecho para Asignar Tour Leader'.

Figura 52. Designar tour líder.

Asignar fechas al listado de actividades

En este listado encontraremos las actividades que se desarrollarán en ese tour y que pertenecen al paquete que se detalla. En el campo de fecha de actividad se debe asignar la fecha que corresponde a cada actividad.

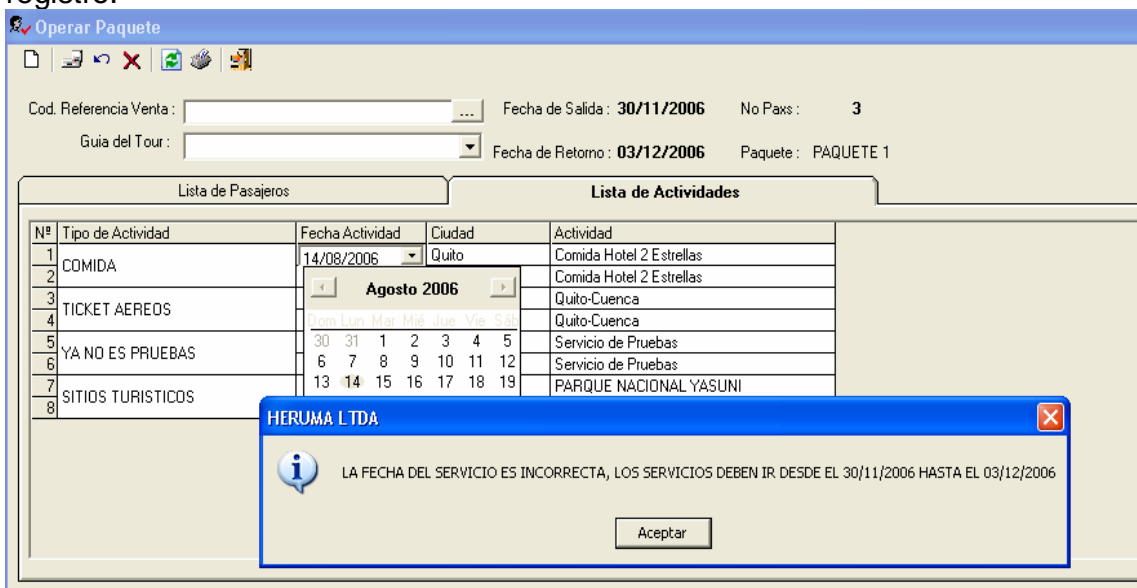


N°	Tipo de Actividad	Fecha Actividad	Ciudad	Actividad
1	COMIDA	30/11/2006	Quito	Comida Hotel 2 Estrellas
2	TICKET AEREOS	01/12/2006	Quito	Quito-Cuenca
3	YA NO ES PRUEBAS	02/12/2006	Cuenca	Servicio de Pruebas
4	SITIOS TURISTICOS	03/12/2006	Quito	PARQUE NACIONAL YASUNI

Figura 53. Asignar fechas al listado de actividades

Mensaje de alerta de fecha incorrecta.

Si se registra fechas que estén fuera del intervalo de tiempo establecido para las actividades del tour se presentara este mensaje para evitar equivocaciones en el registro.



N°	Tipo de Actividad	Fecha Actividad	Ciudad	Actividad
1	COMIDA	14/08/2006	Quito	Comida Hotel 2 Estrellas
2				Comida Hotel 2 Estrellas
3	TICKET AEREOS			Quito-Cuenca
4				Quito-Cuenca
5	YA NO ES PRUEBAS			Servicio de Pruebas
6				Servicio de Pruebas
7	SITIOS TURISTICOS			PARQUE NACIONAL YASUNI
8				

HERUMA LTDA

LA FECHA DEL SERVICIO ES INCORRECTA, LOS SERVICIOS DEBEN IR DESDE EL 30/11/2006 HASTA EL 03/12/2006

Aceptar

Figura 54. Mensaje de alerta de fecha incorrecta.

Estados de pago del Tour

Este reporte da información de los paquetes que han sido vendidos, de acuerdo al color se puede identificar su estado de la siguiente manera:

- El color rojo indica que la fecha límite de cobro del paquete ha excedido.
- El color verde indica que el tour ha sido pagado en su totalidad.
- El color
- El color blanco nos indica tour operado.



N°	Cod. Referencia Tour	Paquete	Fecha Limite	Fecha Tour	Estado Tour	Operado
1	PRUEBAS X 6	JJJJJJJJJJJJ	21/08/2006	21/08/2006	SALDO PENDIENTE	<input type="checkbox"/>
2	Catolicax5	PAQUETE 1	29/08/2006	29/08/2006	PAGADO TOTALMENTE	<input checked="" type="checkbox"/>
3	ESPINX3	PAQUETE 1	22/09/2006	07/10/2006	PAGADO TOTALMENTE	<input type="checkbox"/>

Tour Operado ■ Excedido Fecha Limite ■ Tour Pagado ■ Tour No Pagado dentro de la Fecha Limite

Figura 55. Reporte de estados de pago del Tour.

1.2.1.4 Reportes

Se puede generar reportes de acuerdo a fecha desde y fecha hasta, dentro del menú reportes podemos encontrar los submenús:

- Caja diario
- Pago Créditos
- Cheques recibidos
- Vouchers recibidos
- Reporte de proformas emitidas
- Comisiones
- Asistencia
- Paquetes más vendidos
- Vendedor _ ventas

Todos estos reportes se manejan de acuerdo a la interfaz de la figura 28

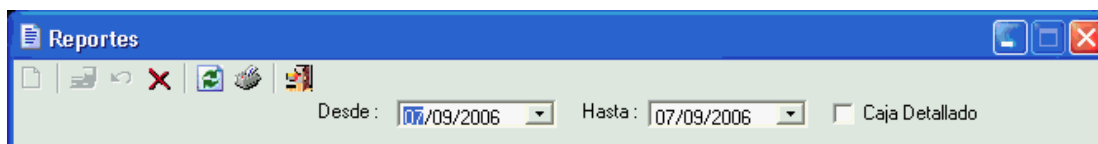


Figura 56. Interfaz para generar reportes.

1.2.1.5 Avanzadas

Datos de conexión a la base de datos

Esta actividad que esta relacionada con las actividades que realiza el administrador consta en registrar el nombre del servidores donde se encuentra la base de datos, usuario SA (Usuario de Autenticación de Windows), el nombre de la base de datos, clave de servidor.



Figura 57. Interfaz para la conexión a la base de datos.

Respaldo de la Base de Datos

Se debe crear una carpeta con el nombre de Respaldos en la unidad donde se encuentre instalada la aplicación, luego se procede a ejecutar el respaldo.

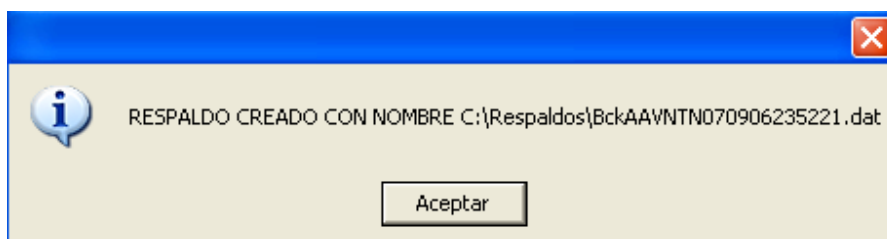


Figura 58. Interfaz para crear respaldo de base de datos

Restaurar Base de Datos

Para restaurar la base se debe realizar los siguientes pasos:

- 1.- Parar la base de datos
- 2.- En el menú Avanzadas, submenú Recuperar BDD

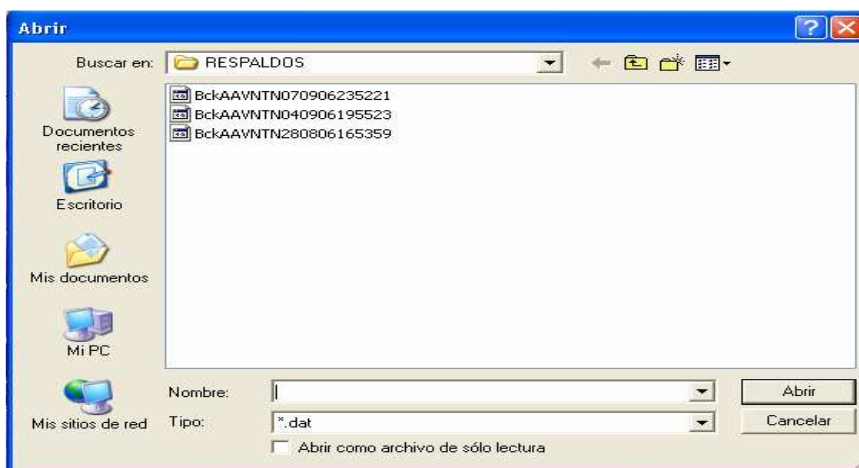


Figura 59. Interfaz que permite localizar la base de datos a restaurar.

- 3.- Se escoge la base que se desee restaurar y Aceptar.
- 4.- Se inicia el Servidor.

CONTENIDO

CAPITULO I	1
1.1 AMBITO	2
1.2 PROBLEMA	2
1.3 OBJETIVOS	3
1.3.1 OBJETIVO GENERAL	3
1.3.2 OBJETIVOS ESPECÍFICOS	3
1.4 ALCANCE Y LIMITACIONES	4
1.4.1 ALCANCE	4
1.4.2 LIMITACIONES	4
CAPITULO II	5
2.1 INGENIERÍA DE SOFTWARE	5
2.1.1 CALIDAD DEL SOFTWARE	5
2.1.2 SOFTWARE DE CALIDAD	6
2.1.3 CONTROL DE LA CALIDAD DEL SOFTWARE	6
2.2 ARQUITECTURA DE DESARROLLO CLIENTE/SERVIDOR	7
2.2.1 CLIENTE	7
2.2.2 SERVIDOR	8
2.2.3 CARACTERÍSTICAS DE LA ARQUITECTURA CLIENTE-SERVIDOR	9
2.2.4 VENTAJAS DE LA ARQUITECTURA CLIENTE/SERVIDOR	9
2.2.5 DESVENTAJAS DE LA ARQUITECTURA CLIENTE/SERVIDOR	11
2.3 REQUERIMIENTOS DE INFRAESTRUCTURA	12
2.3.1 HARDWARE	12
2.3.2 SOFTWARE	13
2.3.2.1 Back-end	13
2.3.2.2 Front-end	14
2.3.2.3 Reportes	14
2.3.2.4 Herramientas CASE de soporte al desarrollo	15

CAPITULO III	17
3.1 CUADRO DE RESUMEN.....	17
3.2 PARADIGMA DE DESARROLLO	18
3.2.1 <i>PARADIGMA ESPIRAL INCREMENTAL ORIENTADO A OBJETOS</i>	18
3.2.1.1 <i>ACTIVIDADES DEL MODELO ESPIRAL INCREMENTAL</i>	19
3.2.1.2 <i>VENTAJAS</i>	20
3.3.1.1 <i>Modelo Estático</i>	21
3.3.1.2 <i>Modelo Dinámico</i>	21
3.3.1.3 <i>Modelo de Procesos</i>	22
3.3.2 <i>FASES DEL CICLO DE VIDA DEL SOFTWARE</i>	22
3.4 UNIFIED MODELING LANGUAGE (UML)	25
3.4.1 <i>OBJETIVOS DEL UML</i>	25
3.4.2 <i>ÁREAS CONCEPTUALES DE UML</i>	26
3.4.3 <i>MECANISMOS DE EXTENSIÓN</i>	27
3.4.4 <i>DESCRIPCIÓN DE LOS DIAGRAMAS UML</i>	27
3.4.4.1 <i>Modelo Estático</i>	27
3.4.4.1.1 <i>Diagramas de caso de uso</i>	27
3.4.4.1.2 <i>Diagrama de Clases</i>	29
3.4.4.1.3 <i>Diagrama de objetos</i>	33
3.4.4.2 <i>Modelo Dinámico</i>	34
3.4.4.2.1 <i>Diagrama de colaboración</i>	34
3.4.4.2.2 <i>Diagrama de secuencia</i>	34
3.4.4.3.1 <i>Diagrama de estados</i>	35
3.4.4.3.2 <i>Diagrama de actividades</i>	35
3.4.4.4 <i>Pruebas</i>	36
3.4.4.4.1 <i>Pruebas de unidad</i>	36
3.4.4.4.2 <i>Pruebas de integración</i>	37
3.4.4.4.3 <i>Pruebas de validación y verificación</i>	37
3.4.4.4.4 <i>Pruebas del sistema</i>	38
3.4.4.4.5 <i>Pruebas de recuperación</i>	38

3.4.4.4.6 Pruebas de seguridad	38
3.4.4.4.7 Prueba funcional	39
3.4.4.4.8 Prueba de desempeño	39
CAPITULO IV.....	40
4.1 CONCLUSIONES.....	40
4.2 RECOMENDACIONES	41
4.3 BIBLIOGRAFÍA	42
ANEXOS	43
MANUAL TÉCNICO	64
MANUAL DE USUARIO	64