

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN**

LICENCIATURA EN SISTEMAS DE INFORMACIÓN

TESIS DE GRADO

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
LICENCIADO EN SISTEMAS DE INFORMACIÓN**

TEMA:

SISTEMAS DE RESERVACIONES DE VUELOS

AUTORES:

María Fernanda Gavilánez Tirado
Sonia Hurtado Casierra

GUAYAQUIL – ECUADOR

AÑO

2002

Agradecimiento

A Dios por su gran bondad y permitirnos llegar a este momento tan importante para nosotras.

A nuestros padres por enseñarnos excelentes valores de la vida que hoy estamos poniendo en alto al culminar con nuestra carrera profesional.

A nuestros familiares por apoyarnos con tesón cuando hemos desfallecido.

A nuestros hijos que en su inocencia nos hacen ver que el superarnos como profesionales nos hace ser mejores madres.

A todos aquellas personas, amigos y compañeros de trabajo, que nos ayudaron a superar distintas fases de nuestras vidas y aprender de ellas en el transcurso de nuestra carrera.

Dedicatoria

A nuestros padres, familiares y amigos por su apoyo incondicional y en especial a nuestros hijos por permitirnos no estar en muchos momentos de su vida a lo largo de la tesis.

A Dios por poner a las personas correctas en nuestro camino.

María Gavilánez

Sonia Hurtado

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

FIRMA DE LOS AUTORES

Sra. María Gaviláñez Tirado



Sra. Sonia Hurtado Casierra

TRIBUNAL DE GRADUACIÓN

Ing. Mónica Villavicencio
Coordinadora

Ing. Carlos Martín
Director Tópico

MIEMBROS PRINCIPALES

Ing. Néstor Arreaga
Miembro Principal

Lic. Jorge Olaya
Miembro Principal

RESUMEN EJECUTIVO

El avance tecnológico y la necesidad de las personas de poder tener un mejor servicio en ciertas compras que realizan, como es la reservación de pasaje de avión, y ya que esto forma parte de un proceso normal dentro las empresas hoy en día; es por esta razón que decidimos crear un sitio web llamado "www.aircondor.com" el mismo que permitirá que las personas puedan realizar sus reservaciones a cualquier lugar del mundo y los asientos que deseen, sin la necesidad de ser atendido por un vendedor.

Para el manejo de este sitio web solo será necesario que las personas tengan un conocimiento básico de computación como por ejemplo conocer las partes principales del computador(CPU, monitor, teclado, mouse), facilitando así la compra de los pasajes de vuelos.

Es por esta razón que nuestro sitio web se ha convertido en una parte indispensable en la vida de los ecuatorianos, ofreciéndole un servicio de primera calidad durante todo el año sin mayores inconvenientes.

Más adelante este documento especifica el análisis y los fundamentos teóricos y técnicos para el desarrollo de nuestro web site www.aircondor.com Posteriormente se muestra el diseño y la presentación de los pasos que se siguieron para la implementación.

INDICE GENERAL

RESUMEN EJECUTIVO

INTRODUCCION

CAPÍTULO 1

1.	ESPECIFICACIONES DEL PROYECTO	1
1.1	Justificación.	1
1.2	Objetivos generales.	4
1.3	Especificaciones funcionales.	4
1.4	Especificaciones no funcionales.	4
1.5	Limitaciones y Restricciones.	5
1.6	Alcances.	6

CAPÍTULO 2 7

2.	ANÁLISIS DE LA INDUSTRIA Y EL PRODUCTO	7
2.1	Definición de la Industria y el Producto.	7
2.2	Clientes.	8
2.2.1	Comportamiento del consumidor ante el mercado convencional.	8
2.2.2	Reacciones del consumidor ante el mercado virtual.	9
2.2.3	Mercado Virtual vs. Mercado Convencional.	10
2.3	Competidores.	12

2.3.1	Funcionalidad de los competidores convencionales existentes.	12
2.3.2	Desafíos encarados por los competidores convencionales.	13
2.4	Productos Sustitutos.	13
CAPÍTULO 3		16
3.	MODELO DE NEGOCIOS	16
3.1	Definición del Modelo de Negocios.	16
3.2	Funcionalidad General.	16
3.2.1	Diagrama de Flujo de Procesos.	18
3.3	Formas de Pagos.	19
3.4	Seguridad.	19
CAPÍTULO 4		21
4.	ARQUITECTURA DEL SISTEMA	21
4.1	Justificación de la selección del modelo.	21
4.1.1	Aplicación StandAlone.	21
4.1.2	Aplicación Web.	23
4.2	Tipos de procesos clientes.	25
4.3	Tipos de procesos servidores.	26
4.3.1	Proceso servidor Web.	26
4.3.2	Proceso servidor de Transacciones.	27
4.3.3	Proceso servidor de Base de datos.	28
4.4	Tipos de Middleware.	28

4.4.1	Hypertext Transfer Protocol (HTTP).	29
4.4.2	COM/DCOM.	30
4.4.3	ADO.	31
4.4.4	OleDb.	32
4.5	Tecnologías y Herramientas Escogidas para la Implantación.	33
4.6	Requerimientos de Hardware y Software para la puesta en producción	34
4.6.1	Requerimientos de Hardware (Servidor).	34
4.6.2	Requerimientos de Hardware (Cliente).	35
4.6.3	Requerimientos de Hardware (Usuario interno).	35
4.6.4	Requerimientos de Software (Servidor).	36
4.6.5	Requerimientos de Software (Cliente).	36
CAPÍTULO 5		38
5. IMPLEMENTACIÓN DE LA CAPA DE DATOS		38
5.1.	Modelo Lógico de Datos.	38
5.2.	Modelo Físico de Datos.	39
5.2.1	Tabla de asientos de vuelos.	40
5.2.2	Tabla de aviones.	41
5.2.3	Tabla de catálogo.	42
5.2.4	Tabla de ciudades.	43
5.2.5	Tabla de clases.	44
5.2.6	Tabla de países.	45
5.2.7	Tabla de clientes.	46

5.2.8	Tabla de provincias.	47
5.2.9	Tabla de reservaciones.	48
5.2.10	Tabla de transacciones.	49
5.2.11	Tabla de tarifas.	50
5.2.12	Tabla de tarjetas.	51
5.2.13	Tabla de temporadas	52
5.2.14	Tabla de vuelos cancelados.	53
5.2.15	Tabla de vuelos.	54
5.3.	Procedimientos Almacenados (Stored Procedures).	55
5.3.1	Procedimiento: spr_CancelaVuelo.	55
5.3.2	Procedimiento: spr_Clientes.	56
5.3.3	Procedimiento: spr_ObtenerDatosReserva.	57
5.3.4	Procedimiento: spr_ObtenerTarifas.	58
5.3.5	Procedimiento: spr_VerificaDisponible.	59
5.3.6	Procedimiento: spr_VerificaReservaciones.	60
5.3.7	Procedimiento: spr_VuelosCancelaALL.	61
5.3.8	Procedimiento: sprActualizaEstado.	62
5.3.9	Procedimiento: sprI_Clientes.	63
5.3.10	Procedimiento: sprE_Reservacion.	64
5.3.11	Procedimiento: sprI_Reservación.	65
5.3.12	Procedimiento: sprI_Reservaciones.	66
5.3.13	Procedimiento: sprI_Transacción.	67
5.3.14	Procedimiento: sprP_VueloEfectuado.	68
5.3.15	Procedimiento: sprR_VuelosCancelWeb.	69

5.3.16 Procedimiento: sprR_VuelosReservaWeb.	70
5.3.17 Procedimiento: sprU_Asientos.	71
5.3.18 Procedimiento: sprU_AsientosC.	72
5.3.18 Procedimiento: sprU_Reservación.	73
5.3.20 Procedimiento: sprU_Clientes.	74
5.3.21 Procedimiento: sprU_Vuelo.	75
5.3.22 Procedimiento: sprVueloCancelado.	76
5.4. Proceso Servidor de Base de Datos.	77

CAPÍTULO 6 **78**

6. IMPLEMENTACION DE LA CAPA DE NEGOCIOS. **78**

6.1 Estructura de los Componentes de Acceso a Datos.	78
6.2 Objetos distribuidos de acceso a datos.	78
6.3 Comunicación con el proceso servidor de acceso a datos.	79
6.4 Estructura de los componentes de negocios.	80
6.5 Objetos distribuidos de negocios.	80
6.6 Comunicación con el proceso servidor de acceso a datos.	85
6.7 Proceso servidor transaccional.	85
6.7.1 Reservación de vuelos.	85
6.7.2 Cancelación de vuelos.	86
6.7.3 Registro de usuario.	87

CAPÍTULO 7	88
7. IMPLEMENTACION DE LA CAPA DE PRESENTACIÓN	88
7.1 Diseño del sitio web-cliente basado en browser.	88
7.2 Diseño del sitio web-cliente basado en browser.	88
7.3 Cliente HTML.	90
7.4 Cliente Script - DHTML.	90
7.5 Server Script - ASP.	91
7.6 CSS.	91
7.7 Comunicación con componentes de negocio.	91
7.8 Proceso Servidor WEB.	92

CONCLUSIONES

RECOMENDACIONES

BIBLIOGRAFÍAS

APÉNDICE A: ARQUITECTURA CLIENTE/SERVIDOR.

APÉNDICE B: PROGRAMACIÓN EN EL WEB.

APÉNDICE C: COM/DCOM.

APÉNDICE D: IIS Y ASP.

OBJETIVOS

Implementar y Proveer al publico en general de un Sitio Web de Calidad para Reservas de Vuelos que cumplan con los siguientes objetivos:

Con respecto a los Clientes, tendrán toda la información del Sitio disponible, permitiendo así que puedan navegar sin problemas (realizar reservas, consultas de vuelos, consultas del Sitio Web, anulación de Vuelos) además nuestro sitio web ofrece seguridad de la Información.

Con respecto a su Funcionamiento, ofrecemos:

- ✈ Información al Día.
- ✈ Fácil acceso.
- ✈ Integridad en la Información.
- ✈ Seguridades por usuarios.
- ✈ Traspaso de pasajeros inmediatamente por vuelos cancelados.
- ✈ Control de vuelos
- ✈ Consultas de vuelos por cliente y sus estados (reservados, ejecutados, y anulados).
- ✈ Imprime comprobante de pago.
- ✈ Imprime las consultas realizadas.

El usuario podrá comunicarse con mayor facilidad a la nuestra Aerolínea.

Mayor cobertura de usuarios

Seguridad en los datos.

DIRIGIDO A

Este proyecto esta dirigido a aquellas personas que desean vivir a la vanguardia de los avances tecnológicos con los que contamos en la actualidad como lo hace la computación, ya que a través de la computación podemos realizar de forma casi inmediata todos los proceso repetitivos que se realizan en las oficinas de una manera mucho más segura y controlada, como por ejemplo las reservaciones de vuelos.

Nuestro sitio web fue diseñado para todo tipo de persona a nivel de conocimiento en computación, es auto educativo, amigable y fácil de usar y permite que las personas realicen sus transacciones sin mayor dificultad.

Además nuestro sitio Web podrá ser visitado por : El publico en general.

INTRODUCCIÓN

Aerolíneas Cónдор, empresa de servicios de vuelos, ha sentido la necesidad de llegar a sus clientes en forma más directa e inmediata, sin importar dónde se encuentre, es por eso que ha escogido a la Internet como el medio que le permitirá obtener estos beneficios.

Por esta razón fue desarrollado www.aircondor.com, un sitio web diseñado con la finalidad de suplir las necesidades actuales de las personas al momento de realizar reservaciones de vuelos, presentando y exponiendo la información necesaria que cada uno de ellos desearía obtener.

Capítulo 1

1. ESPECIFICACIONES DEL PROYECTO

1.1 Justificación.

Proveer al público en general de un sitio web para reservaciones de vuelos ya es una necesidad en nuestro medio, a continuación una breve explicación.

Qué significa reservar vuelos a través del Internet?

Hoy en día realizar una reservación de vuelos, como comúnmente se lo ha realizado en estos últimos años, genera una demanda grande de tiempo consumido por cada una de las personas al momento de realizar esta gestión, como por ejemplo:

- ✈ Informarse sobre las agencias que proveen este servicio.
- ✈ Comunicarse con la agencia.
- ✈ Trasladarse hasta la agencia.
- ✈ Informarse sobre las bondades que ofrece la aerolínea.

- ✈ Trasládarse hasta la agencia para realizar la reservaci3n.
- ✈ Pedir informaci3n sobre sus vuelos.

Los medios de reservaciones que se utilizan hoy en d3a en nuestro medio de vida permiten que los usuarios realicen todas las transacciones con respecto a los vuelos. Logrando as3 por este medio que los usuarios optimicen el tiempo al realizar dichos procesos.

Es por esta raz3n que el comercio electr3nico no es una tecnolog3a 3nica y uniforme, sino que se caracteriza por su diversidad. Puede implicar un amplio rango de operaciones y transacciones comerciales, incluyendo:

- ✈ Establecimiento del contacto inicial, por ejemplo entre un cliente potencial y un proveedor potencial.
- ✈ Intercambio de informaci3n.
- ✈ Soporte pre y posventa (detalles de los productos y servicios disponibles, gu3a t3cnica del uso del producto, respuestas a preguntas de adecuaci3n,...).
- ✈ Ventas.
- ✈ Pago electr3nico (usando transferencia electr3nica de fondos, tarjetas de cr3dito, cheques electr3nicos, caja electr3nica).

- ✈ Distribución, incluyendo tanto gestión de distribución y reparto para productos físicos, como distribución de los productos que puedan ser repartidos electrónicamente).
- ✈ Asociaciones virtuales, grupos de empresas independientes que aúnan sus competencias de manera que puedan ofrecer productos o servicios que van más allá de la capacidad de cada una de ellas individualmente.
- ✈ Procesos empresariales compartidos que son llevados a cabo y de los que son propietario una empresa y sus socios.

Igualmente, el comercio electrónico implica un amplio rango de tecnologías de comunicaciones incluyendo correo electrónico, fax, intercambio electrónico de datos (EDI) y transferencia electrónica de fondos (EFT). La elección de unas u otras depende del contexto.

Además se define una reservación electrónica de un servicio como aquella en la que el usuario realiza su transacción de una forma completa y entera.

Basándonos en toda esta investigación especificada anteriormente nace la necesidad de crear un proyecto que permita cubrir la necesidad de realizar reservaciones rápidas y efectivas con la seguridad extrema que se necesitan con el manejo de esta información, con una amplia cobertura; como lo es el sitio web de reservaciones de vuelos de Aerolíneas Cónдор.

1.2 Objetivos generales.

Realizar un sitio web el mismo en el que cualquier persona pueda consultar la disponibilidad de vuelos y realizar las respectivas reservaciones, reduciendo a los usuarios el tiempo empleado al realizar éstas gestiones.

Uno de los objetivos más importantes de nuestro sitio web es ser indispensable en la vida de las personas, así ellos podrán realizar con mayor facilidad sus reservaciones de vuelos a cualquier parte del mundo en www.aircondor.com.

1.3 Especificaciones funcionales.

Diseñar un sistema virtual a través de la Internet que permita al público en general realizar sus reservaciones de vuelos en pocos minutos, pero de una manera más confiable. El usuario deberá ingresar a nuestro sitio www.aircondor.com donde quedará registrado como nuestro cliente; y se le permitirá realizar diversos tipos de consultas e imprimirlas, fuera de los procesos que abarca una reservación de vuelos.

1.4 Especificaciones no funcionales.

Este sitio web permite realizar las consultas tanto específicas como generales por medio de un usuario registrado. Dando así a nuestros clientes seguridad en el ingreso de su información.

Esta forma de ingreso es necesaria para que el sitio web realice los reportes de auditorias que serán de utilidad para Aerolíneas Cónдор.

1.5 Limitaciones y Restricciones.

Las limitaciones que nuestro sitio web presenta son la siguientes:

- ✈ El cliente no podrá realizar ningún tipo de transacción ni consultas si no se encuentra registrado como usuario del sitio web.
- ✈ No se registran nombres de las personas para los pasajes extras que compre un cliente.
- ✈ El débito ó crédito de las distintas tarjetas que se manejan no es en línea.
- ✈ No se permite crear vuelos con escala.

Las restricciones que el sitio web presenta son la siguientes:

- ✈ Si un vuelo se cancela por fallas técnicas no existe canje de dinero por pasaje, los clientes son asignados a otro vuelo.
- ✈ No se generan vuelos por solicitudes de clientes.
- ✈ Los vuelos tienen sus horarios de salida y llegada programados.

1.6 Alcances.

El sitio web tiene alcance a nivel nacional, toda persona en este país que tenga acceso a Internet podrá acceder a www.aircondor.com .

El cliente puede realizar transacciones como:

- ✈ Reservaciones de 1 ó más vuelos
- ✈ Reservar pasajes de: ida e ida y vuelta
- ✈ El cliente puede cancelar sus vuelos y el pago por el mismo es devuelto.
- ✈ Permite realizar pagos por medio de diversas tarjetas.

El sitio web ofrece las siguientes opciones:

- ✈ Registro de usuarios.
- ✈ Actualización de datos del usuario.
- ✈ Consulta de vuelos disponibles y vuelos realizados en la aerolínea.
- ✈ Impresión de comprobante de pago (reservación) y comprobantes de reembolso (cancelación).

Capítulo 2

2. ANÁLISIS DE LA INDUSTRIA Y EL PRODUCTO

2.1 Definición de la Industria y el Producto.

Hoy en día, Internet cuenta con millones de usuarios en todo el mundo y el número sigue aumentando a diario, debido a la facilidad que brinda Internet para obtener información desde la comodidad de la casa u oficina. Este número creciente de usuarios pueden ser clientes potenciales para las empresas o negocios. En Internet, cada día también aumenta la cantidad de productos y servicios que se comercializan por este medio, facilitando conseguir mejores precios y nuevos proveedores mejorando así los márgenes de utilidad.

Por estas razones es que Aerolíneas Cónдор, empresa de servicio aéreo, ha considerado la necesidad de tener presencia en Internet y beneficiarse de todos los recursos que éste ofrece. Para ello ha sido indispensable colocar y

estructurar toda la información que desea sea vista en páginas web, obteniendo como resultado www.aircondor.com el cual es a su vez un sitio de la World Wide Web (www) que permite realizar reservaciones de vuelos on-line.

Existen otros sistemas que permiten realizar reservaciones de vuelos pero no cuentan con la funcionalidad que éste sitio web ofrece a sus clientes.

2.2 Clientes.

Este sistema ha sido desarrollado con el fin de obtener los siguientes beneficios:

- ✈ Aumentar las ventas de pasajes facilitando a los clientes el proceso de reservación de vuelos.
- ✈ Incrementar la base de datos de clientes o posibles clientes.
- ✈ Mejorar la imagen de la empresa como una Aerolínea innovadora que ofrece un mejor servicio haciendo uso de la tecnología actual.

2.2.1 Comportamiento del consumidor ante el mercado convencional.

En Aerolíneas Cónдор la reservación de vuelos se podía realizar de dos maneras:

- ✈ Visitando personalmente las oficinas de la Aerolínea.

- ✈ Visitando agencias de viaje que ofrecen en sus paquetes los vuelos de Aerolíneas Cónдор.

Debido al bajo volumen de ventas, Aerolíneas Cónдор realizó una encuesta de satisfacción a los clientes en la cual se detectó los motivos que han ocasionado la pérdida de posibles ventas:

Falta de tiempo de los clientes para ir a los puntos de venta de los vuelos.

Falta de información acerca de la disponibilidad de los vuelos(rutas, horarios, asientos).

Este estudio de mercado fue el que determinó que Aerolíneas Cónдор necesitaba una herramienta de apoyo para ofrecer mejor sus servicios y en forma más directa con el cliente.

Es por eso que para esta empresa el contar con un sitio web no es un gasto es una inversión totalmente justificada. Actualmente no se puede vislumbrar a una gran empresa en cualquier parte del mundo sin un sitio web.

2.2.2 Reacciones del consumidor ante el mercado virtual.

A pesar de que se conoce las ventajas del uso de Internet y la importancia que tiene en nuestras actividades diarias, el consumidor aún siente temor en la utilización de los sistemas virtuales, especialmente en aquellos en los que se realizan pagos, pues se consideran no confiables ya sea porque puede

existir fraude electrónico o por dificultades en su manejo. Sin embargo este temor ha ido desapareciendo poco a poco por la difusión de la cultura sobre el uso de la Internet demostrando al consumidor que un sistema virtual es una herramienta fácil, segura, necesaria e incluso se puede convertir en algo indispensable para su vida cotidiana.

En nuestro país, el comercio electrónico está empezando a alcanzar un gran auge y crecimiento ya que muchas empresas ofrecen sus servicios y productos a través de la Internet.

2.2.3 Mercado Virtual vs. Mercado Convencional.

Mercado Virtual (Ventajas)

Fácil acceso: cualquier usuario que cuenta con un computador con acceso a Internet puede hacer uso del sitio www.aircondor.com

Gestión inmediata: evita tener que realizar actividades personalmente.

Proporcionar información: hacer consultas de transacciones efectuadas (reservaciones y cancelaciones) como también de vuelos, itinerarios y costos.

Reducción de Costos: eliminar costos de publicidad, ventas y papelería ya que se automatizan procesos de negocios.

Recibir retroalimentación: es más probable que un cliente nos envíe retroalimentación, ya que la pueden enviar en forma anónima.

Nuevo Canal de Ventas: se agrega un nuevo canal de ventas a los ya existentes, permitiéndonos llegar a nuevos clientes.

Atención Permanente: permite que los clientes sean atendidos las 24 horas del día.

Mercado Virtual (Desventajas)

Tener conexión a Internet: se requiere contar con conexión a Internet para poder hacer uso del sitio.

Costos elevados por uso telefónico (conexión dial-up): las tarifas telefónicas en nuestro país son altas lo que muchas veces obliga a las personas prescindir de este servicio.

Transacciones de tarjetas de crédito no se reflejan automáticamente: se emite un reporte de reservaciones y/o cancelaciones efectuadas por cada tarjeta de crédito con corte diario.

Marco legal: algunos temas legales y regulaciones no están todavía resueltos para considerar todo lo que puede ocurrir dentro del comercio electrónico.

Mercado Convencional (Ventajas)

Confianza al realizar la gestión personalmente: a través del tiempo las personas se han acostumbrado en la utilización del sistema convencional por lo que existe resistencia al cambio de un sistema virtual.

Servicio personalizado: la atención al cliente es de persona a persona.

Mercado Convencional (Desventajas)

Pérdida de tiempo: el cliente actualmente tiene que desplazarse físicamente al punto de venta de la aerolínea y en algunos casos para no encontrar vuelos disponibles.

2.3 Competidores.

2.3.1 Funcionalidad de los competidores convencionales existentes.

El sistema convencional de reservaciones de vuelos se realiza de la siguiente manera:

El cliente interesado en realizar una reservación se dirige a la oficina de la aerolínea o a una agencia de viajes para averiguar los vuelos disponibles según la ruta, la fecha e itinerario especificado.

Tanto la agencia o aerolínea le muestran las opciones y las tarifas según la clase seleccionada.

Luego que el cliente escoge el vuelo se emite el boleto respectivo y un comprobante de pago.


2.3.2 Desafíos encarados por los competidores convencionales.

Aircondor.com no representa un desafío para las agencias de viajes sino un beneficio porque podrían hacer uso del sitio en el momento de solicitar información sobre los vuelos disponibles y sus tarifas.

2.4 Productos Sustitutos.

Además de las agencias de viajes, en el mercado existen otros sitios web que cumple con éstas funcionalidades.

A continuación se muestran las fortalezas que tiene Aircondor.com y los productos de la competencia.

Característica	 AirCondor	 amADEUS	 Sabre
<i>Explicativo</i>	Provee formulario muy explicativo para ingresar los datos de registro.	D: Requiere que otro le explique que datos debe ingresar en que sitio.	D: El formulario lo encuentra sólo en Inglés.
<i>Compatible</i>	Excelente resolución con Internet Explorer 5.0 y posteriores versiones, se comporta bien con Netscape Navigator.	D: Corre muy bien con Netscape Navigator pero no soporta ninguna versión de Internet Explorer.	D: Tiene buena resolución con Netscape Navigator. No soporta Internet Explorer 5.0
<i>Manejo de gran volumen de transacciones</i>	Diseñado para manejar accesos masivos de consultas y registros por varios usuarios.	Es rápido con gran volumen de información.	D: Presenta un poco de retardo al realizar consultas.

Característica	 AirCondor		
<i>Interfase amigable</i>	Cuenta con diseño gráfico atractivo y amigable al usuario con lo que podrá realizar fácilmente sus transacciones.	D: Se necesita realizar un curso para que el usuario pueda manejar el sistema.	Es amigable para el usuario.
<i>Interactivo</i>	Los requerimientos son hechos en línea, mediante el ingreso de datos por formulario en el sitio.	D: No brinda la oportunidad de ver resultados en línea.	Si posee comunicación en línea.
<i>Multiusuario</i>	Varios usuarios pueden trabajar en la aplicación al mismo tiempo sin problemas	Varios usuarios pueden trabajar en la aplicación al mismo tiempo sin problemas	D: Varios usuarios pueden trabajar en la aplicación con tiempo de retardo mínimo.
<i>Seguridades</i>	Sus datos no corren riesgos en nuestra base e interfase.	D: No muy confiable por usar método get de envío de datos.	D: se puede obtener información de usuarios.

Característica	Oportunidades de www.aircondor.com 
<i>Explicativo</i>	Es el portal más ágil que nuestros competidores del mercado.
<i>Compatible</i>	Si posee sólo un browser no importa, usted puede ver nuestra página en los dos navegadores más conocidos a nivel mundial (Internet Explorer y Netscape)
<i>Manejo de gran volumen de transacciones</i>	Brindamos la mayor cantidad de transacciones por usuarios.
<i>Interfase amigable</i>	Contamos con interfase sencilla que permite llegar a usar la página con agilidad.

Característica	 Oportunidades de www.aircondor.com
<i>Interactivo</i>	Permite mantener la información actualizada en tiempo real.
<i>Multiusuario</i>	Acceso por varios usuarios que trabajaran al mismo tiempo en la aplicación.
<i>Seguridades</i>	Confiabilidad en todo sentido durante y después de su registro de información.

Debilidades

Característica	 AirCondor	 AMADEUS	 Sabre
<i>Falta de reconocimiento</i>	La empresa es conocida en el medio turístico pero en el mundo de la Internet.	Es muy conocido por las aerolíneas, usuarios y agencias de viajes.	Es conocido por las agencias de viajes.

Característica	 Amenazas de www.aircondor.com
<i>Falta de reconocimiento</i>	Esto es superable en el transcurso de un mes mediante el uso de varios sectores de información como los diarios, revistas, correo directo y los mismo sitios de la Internet con los que Aerolíneas Cónдор se dará a conocer mayoritariamente.

Capítulo 3

3. MODELO DE NEGOCIOS

3.1 Definición del Modelo de Negocios.

Nuestro sitio web www.aircondor.com toma sus bases en el modelo de negocio B2C (Business to Consumer), que brinda la oportunidad de realizar reservaciones de vuelos a todo tipo de persona de una manera más óptima a través del Internet.

Este modelo de negocio fue implementado para cubrir una necesidad en Aerolíneas Cónдор: ofrecer un mejor servicio a sus clientes. Además de agilizar el proceso de retroalimentación en la información de los clientes.

3.2 Funcionalidad General.

El propósito de Aerolíneas Cónдор en crear el sitio web www.aircondor.com es dar a sus clientes un mejor servicio al momento de realizar sus

reservaciones de vuelos a cualquier lugar del mundo. A continuación mencionaremos el funcionamiento de este sitio web.

Acceso y Seguridad, El cliente tendrá un usuario registrado en este sitio web con una clave de acceso para una mayor seguridad, el mismo con el que tendrá acceso a todas las opciones con las que cuenta www.aircondor.com.

Manejo de la Información, www.aircondor.com presenta un sitio web totalmente fácil de manejar, por la forma clara y ordenada que se presenta.

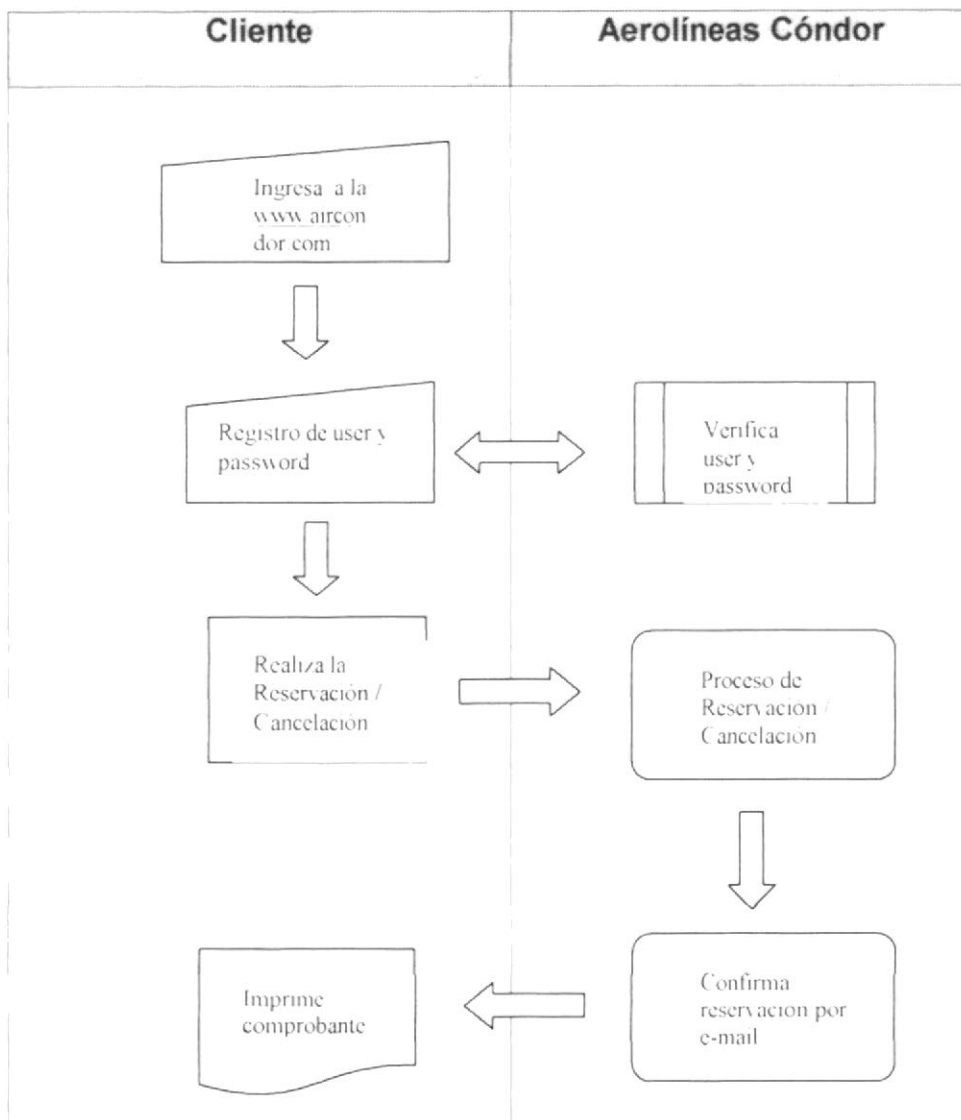
Procesos del sitio web, Al momento que el usuario ingresa a www.aircondor.com deberá ingresar su usuario y clave y elegir el proceso que desea efectuar, en ese momento se le pedirá que ingrese cierta información de importancia para realizar estos procesos. Si todo es correcto el proceso se efectuará, caso contrario presentaremos mensajes de ayuda para el cliente.

Mensajes salientes, www.aircondor.com ofrece a sus clientes confirmación de los procesos realizados, esto es permitiéndoles imprimir un comprobante de débito / crédito dependiendo de la transacción realizada además de la confirmación por e-mail.

Información presentada, www.aircondor.com maneja diversos tipos de consultas y reportes a nivel de clientes como para la empresa en sí. Los

clientes podrán consultar todas sus transacciones a través de su usuario. Mientras que la empresa podrá ver toda la información tanto de configuración como de los transacciones que se realizan en el sitio web.

3.2.1 Diagrama de Flujo de Procesos.



3.3 Formas de Pagos.

El sitio web www.aircondor.com ofrece a sus clientes la oportunidad de cancelar sus reservaciones de vuelos con tarjetas de crédito, acepta todo tipo de tarjeta de crédito aprobada en el Ecuador.

La forma de pago que adoptó Aerolíneas Cónдор es la más segura de manejar a través de la Internet, logrando así que los usuarios puedan utilizar el crédito de sus tarjetas para la compra de sus pasajes de vuelos.

Cuando un cliente cancela una reservación por la Internet, el pago realizado por los boletos se convierte en una nota de débito para los clientes.

El valor de los pasajes de vuelo varían de acuerdo a las rutas de vuelo.

3.4 Seguridad.

Conocer el potencial que tiene la Internet al momento de cambiar la forma de hacer los negocios, a través de un sistema de servicio que soporta los procesos de ventas, marketing, pagos, etc. El proporcionar a la aerolínea de una buena solución es una tarea compleja que requiere de una considerable investigación y planificación así como el conocimiento de las diferentes seguridades y leyes de legislación que este medio nos proporciona para su puesta en marcha.

Por esta razón mencionaremos algunas de las medidas que Aerolíneas Cándor adoptó para brindar un servicio seguro a los clientes que realizan transacciones en el sitio web:

- ✈ El cliente que desee realizar transacciones deberá registrarse como usuario indicando sus datos personales.
- ✈ El boleto reservado por el cliente se entrega físicamente en ventanilla previo a la presentación del comprobante impreso que se muestra al término de la transacción o del e-mail enviado a la dirección del cliente y de documentos personales como la cédula o pasaporte.
- ✈ En el caso de cancelaciones de vuelos la aerolínea confirma vía telefónica si el cliente realizó la cancelación respectiva.
- ✈ Finalizado cualquier proceso de verificación la aerolínea llama a la entidad encargada de la tarjeta de crédito para que realice el débito o el crédito respectivo en la tarjeta del cliente.

Capítulo 4

4. ARQUITECTURA DEL SISTEMA

4.1 Justificación de la selección del modelo.

El sistema desarrollado para Aerolíneas Cónдор ha sido dividido en :

✈ Aplicación StandAlone

✈ Aplicación Web.

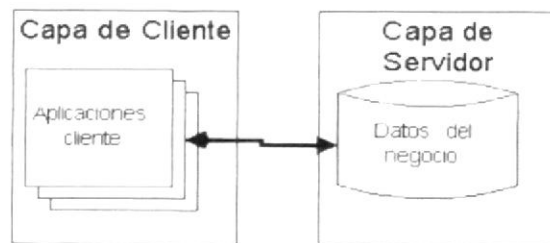
La primera está basada en la arquitectura de dos capas (Two-Tier Fat-Client) y la segunda, arquitectura de tres capas (Three-Tier).

4.1.1 Aplicación StandAlone.

La aplicación standalone fue desarrollada como complemento a la aplicación web.

Debido a la cantidad de usuarios que harán uso de ésta aplicación (la empresa indicó que serían 2 usuarios) se seleccionó el Modelo de dos capas, cuyas características mencionamos a continuación:

Modelo Dos Capas.



El modelo de dos capas o aplicaciones cliente-servidor estándar agrupan presentación y componentes lógicos de aplicación en la máquina del cliente y el acceso a la fuente de datos compartidos usando la conexión de red.

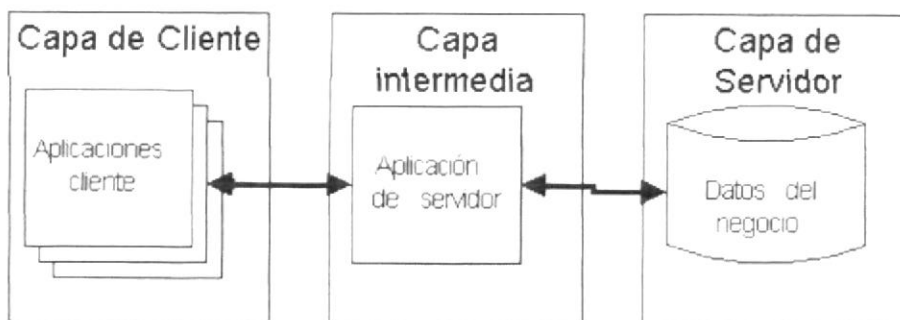
El proceso cliente se conecta al proceso servidor (servidor de base de datos) para el envío de solicitudes.

Las aplicaciones de dos capas trabajan bien en ambientes controlados con un número de usuarios por debajo de los 100, una base de datos única y segura y una red rápida.

4.1.2 Aplicación Web.

La aplicación web debido al volumen de transacciones que van a ser manejadas se seleccionó el modelo de tres capas, cuyas características mencionamos a continuación:

Modelo Tres Capas.



En esta arquitectura la capa de presentación, lógica y datos están conceptualmente separadas. Los componentes de presentación administran la interacción con el usuario, y los servicios de requerimientos de la aplicación por la llamada de componentes de la capa intermedia. Los componentes de la aplicación ejecutan lógica del negocio y hacen requerimientos a la base de datos.

La arquitectura de tres capas es a menudo llamada servidor-central (server-centric) debido a que ella únicamente habilita a los componentes de la aplicación a correr en los servidores de la capa intermedia, independiente tanto de la interfase de presentación y la implementación de la base de datos.

La independencia de la lógica de la aplicación de la presentación y los datos ofrece los siguientes beneficios:

Soporte multi-lenguaje: los componentes pueden ser desarrollados usando lenguajes generales de programación (visual fox, visual basic, visual c)

Componentes centralizados: facilidad en el desarrollo, mantenimiento e implantación.

Balaceo de carga: componentes de la aplicación pueden ser ubicados en múltiples servidores concediendo mejor escalabilidad (facilidad con la cual una aplicación o conjunto de servicios pueden ser movidos a un ambiente más demandante, con un pequeño impacto en el rendimiento del sistema).

Acceso a datos más eficiente: problemas de limitación de conexión de base de datos son minimizados debido a que la base de datos es vista sólo por el componente de la aplicación y no por todos sus clientes.

Seguridad mejorada: se puede permitir o denegar el acceso componente por componente, simplificando la administración.

4.2 Tipos de procesos clientes.

Como se mencionó anteriormente Aircondor.com está compuesto por dos tipos de clientes, los cuales son: dependiente e independiente.

Cliente Dependiente.- Es el sitio virtual a ser utilizado por los usuarios a quienes está dirigida la funcionalidad de Aircondor.com. Los lenguajes de programación utilizados fueron: JAVASCRIPT (ASP del cliente) y VBSCRIPT (ASP del servidor), utilizando como herramienta de desarrollo Visual InterDev 6.0. Puesto que las páginas del cliente fueron implementadas con JAVASCRIPT la aplicación no tiene dependencia de browser.

Este portal o sitio web estará disponible al público las 24 horas del día y los 365 días del año, en donde los clientes(usuarios) de Aerolíneas Cónдор podrán efectuar cualquier tipo de reservaciones que deseen.

Cliente Independiente.- Es el sistema aplicativo en donde se realizará:

- ✈ Mantenimiento de la información que se muestra en el sitio virtual (países, provincias, ciudades, tarjetas de crédito, aviones, vuelos, temporadas, tarifas por temporadas, asientos disponibles para cada vuelo, etc).

- ✈ Transferencias de personas de un vuelo a otro cuando existe la cancelación total de un vuelo.
- ✈ Reportes sobre clientes y sus movimientos.
- ✈ Reportes de movimientos de tarjetas de crédito.

Esta aplicación fue desarrollada en Microsoft Visual Basic 6.0 y está dirigida al usuario administrativo(interno) de Aerolíneas Cónдор.

4.3 Tipos de procesos servidores.

Para el desarrollo de Aircondor.com se utilizó tres tipos de procesos servidores :

- ✈ Servidor Web
- ✈ Servidor de Transacciones
- ✈ Servidor de Bases de datos

4.3.1 Proceso servidor Web.

El servidor web que se utilizó para la publicación y administración de Aircondor.com fue Microsoft Internet Information Server (IIS) que es un producto del Option Pack de Windows NT 4.0

El cliente solicita al web browser (Internet Explorer o Netscape Navigator) requerimientos HTTP al proceso servidor que es el Web Server (IIS). La solicitud que este recibe es por una página estática HTML o por una programa que genere dinámicamente el código HTML que posteriormente será enviado al browser.

En conclusión el objetivo principal de este proceso servidor es:

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

Para más información véase Apéndice.

4.3.2 Proceso servidor de Transacciones.

El servidor de transacciones utilizado en Aircondor.com es Microsoft Transaction Server, MTS. Es un programa que corre en un servidor de Internet o de red y administra las solicitudes de transacción de aplicaciones y solicitudes en representación del usuario de una estación cliente. El servidor de transacciones verifica el equipo del usuario y cliente para formular a los

servidores de bases de datos (SQL Server 7.0). También administra la seguridad, la conexión a otros servidores y la integridad de las transacciones.

MTS corresponde al Option Pack de Windows NT 4.0.

Para más información véase Apéndice.

4.3.3 Proceso servidor de Base de datos.

El servidor de base de datos utilizado para Aircondor.com es Microsoft SQL Server 7.0, el cual tiene como finalidad:

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

Para más información véase Apéndice.

4.4 Tipos de Middleware.

Middleware es el software distribuido necesario para el soporte de interacciones entre clientes y servidores, es decir, el software que ocupa la

parte intermedia del sistema cliente/servidor. En Aircondor.com se hace uso de los siguientes tipos de middleware:

- ✈ Middleware para internet: HTTP
- ✈ Middleware para objetos: COM/DCOM
- ✈ Middleware para base de datos: ADO/OLEDB.

4.4.1 Hypertext Transfer Protocol (HTTP).

Es un protocolo de comunicación entre Web Browser y Web Server usado para acceder y obtener recursos. Es un protocolo sin estado (stateless).

Por cada requerimiento realiza lo siguiente:

- ✈ El cliente establece una conexión con el servidor.
- ✈ El cliente envía el requerimiento.
- ✈ El servidor entonces procesa el requerimiento, retorna una respuesta y cierra la conexión.

Es forma detallada diríamos que:

El cliente hace un requerimiento por una página de hipertexto (HTML) y a medida que "lee" el documento identifica recursos adicionales (imágenes, videos, etc.), luego envía una secuencia de adicionales requerimientos para

obtener todos los recursos referenciados en el documento y una vez que el cliente ha obtenido el documento principal (HTML) y todos los recursos adicionales, le indica al usuario. Finalmente el usuario examinará la página web y típicamente hará un 'click' sobre un enlace y se moverá a otro documento.

Este protocolo tiene como característica hacer una nueva conexión por cada requerimiento, el cliente debe esperar por una respuesta antes de enviar un nuevo requerimiento.

Http es un protocolo simple porque hace "una cosa a la vez", lo que hace fácil su implementación.

4.4.2 COM/DCOM.

Es una tecnología de apoyo para el desarrollo de aplicaciones distribuidas que resuelve algunos de los problemas que la industria del software ha experimentado al intentar mantener y extender grandes sistemas de información. Está basada en la programación orientada a objetos (OOP).

Cuenta con una infraestructura de software sofisticada que permite que objetos se comuniquen a través de los límites de procesos, computadores y redes (transparencia de ubicación)

Es un estilo de programación y un conjunto de disciplinas requeridas para trabajar en un ambiente distribuido que provee un estándar para la escritura de componentes porque indica cómo construir su estructura física, cómo crear relaciones entre ellos y cómo deben comunicarse e intercambiar datos. Además tiene un conjunto de estándares para asegurar interoperabilidad entre componentes.

Este modelo está basado en el reuso binario cuyo propósito es permitir que:

- ✈ Los desarrolladores pueden compartir su trabajo sin revelar su propiedad intelectual.
- ✈ Eliminar el problema del tiempo de compilación
- ✈ Independencia del lenguaje de programación

4.4.3 ADO.

Ado es una interfase a nivel de aplicación que hace más fácil el acceso a cualquier OLEDB data provider:

- ✈ Base de datos relacionales.
- ✈ Base de datos no relacionales.
- ✈ Bodegas de correos.
- ✈ Sistemas de archivos.

- ✈ Texto, gráficos.
- ✈ ODBC data sources.

Ado es un API fácil de usar e independiente del lenguaje de programación. Mejora los tiempos de respuesta y provee un acceso a los datos con alto rendimiento:

- ✈ Minimiza el tráfico en la red.
- ✈ Minimiza el número de capas entre la aplicación cliente y la fuente de los datos.

4.4.4 OLEDB.

Es una tecnología de Microsoft para crear documentos compuestos. La idea de OLEDB es embeber o enlazar documentos producidos en un tipo de aplicación, en un documento de otra aplicación.

Un documento compuesto puede incluir objetos de 2 formas:

- ✈ Vinculados (Linking): El documento compuesto contiene una referencia al objeto, pero no el objeto en sí
- ✈ Incrustados (Embedding): El objeto en sí se convierte en parte del documento compuesto

Ventajas de linking.

- ✈ Si el objeto cambia, todos los documentos compuestos lo reflejarán automáticamente.
- ✈ Lo del cambio es relativo.
- ✈ Tamaño del documento compuesto es menor de lo que sería si el objeto estuviera incrustado.

Ventajas de embedding.

- ✈ El objeto está fijo en el documento compuesto, por lo que es portable junto con el documento.
- ✈ Tamaño del documento compuesto es mayor, ya que contendrá realmente el objeto.

4.5 Tecnologías y Herramientas Escogidas para la Implantación.

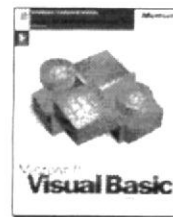
A continuación se presentan las tecnologías que se utilizaron para el desarrollo del sitio web Aircondor.com :

- ✈ Tecnología COM.
- ✈ Active Server Page.

- ✈ Protocolo HTTP.
- ✈ Tecnología OLEDB.
- ✈ Tecnología ADO.

Las herramientas utilizadas son las siguientes:

- ✈ Microsoft Visual Interdev 6.0.
- ✈ Microsoft Visual Basic 6.0 .
- ✈ Microsoft Sql Server 7.0.
- ✈ Crystal Reports Professional versión 7.0.



4.6 Requerimientos de Hardware y Software para la puesta en producción

4.6.1 Requerimientos de Hardware (Servidor).

Un computador con las siguientes características :

- ✈ Capacidad en DD 40GB o superior.
- ✈ 256 MB en Memoria Ram o superior.
- ✈ Procesador Pentium III o superior.
- ✈ Con conexión a Internet dedicada.

4.6.2 Requerimientos de Hardware (Cliente).

Un computador con las siguientes características :

- ✈ Capacidad en DD 30GB o superior.
- ✈ 128 MB en Memoria Ram o superior.
- ✈ Procesador Pentium III o superior.

4.6.3 Requerimientos de Hardware (Usuario interno).

Un computador con las siguientes características :

- ✈ Capacidad en DD 10GB o superior
- ✈ 64 MB en Memoria Ram o superior
- ✈ Procesador Pentium III o superior

4.6.4 Requerimientos de Software (Servidor).

El software que se requiere es el siguiente:

- ✈ Windows NT 4.0 y el Option Pack que incluye Transaction Server e Internet Information Server (IIS).
- ✈ Microsoft Sql Server 7.0 (SP3).
- ✈ Microsoft Internet Explorer 4.0 o superior.
- ✈ ADO 2.5 es utilizado para los accesos a las bases de datos SQL.

4.6.5 Requerimientos de Software (Cliente).

El software que se requiere es el siguiente:

- ✈ Windows 98 o superior.
- ✈ Microsoft Internet Explorer 4.0 o superior.

4.6.6 Requerimientos de Software (Usuario interno)

El software que se requiere es el siguiente:

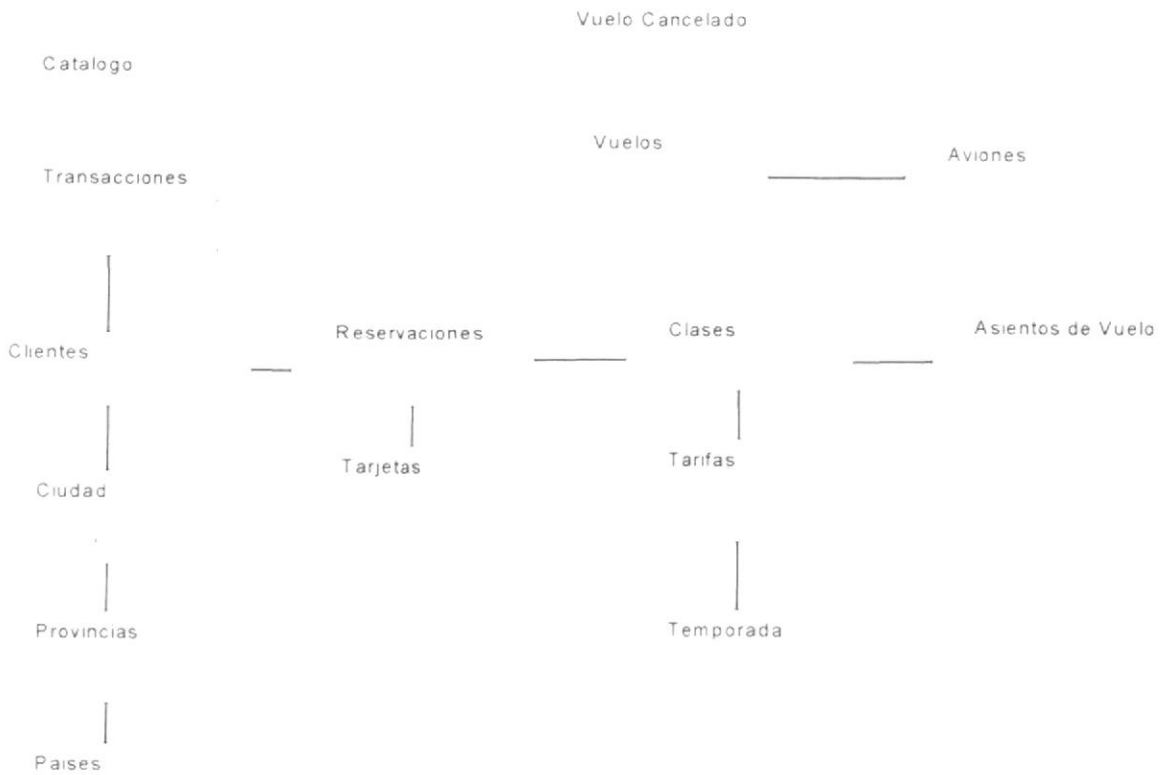
- ✈ Windows 98 o superior.

- ✈ ADO 2.5 es utilizado para los accesos a las bases de datos SQL.
- ✈ Crystal Reports 7.0 (instalador generado para estaciones clientes).

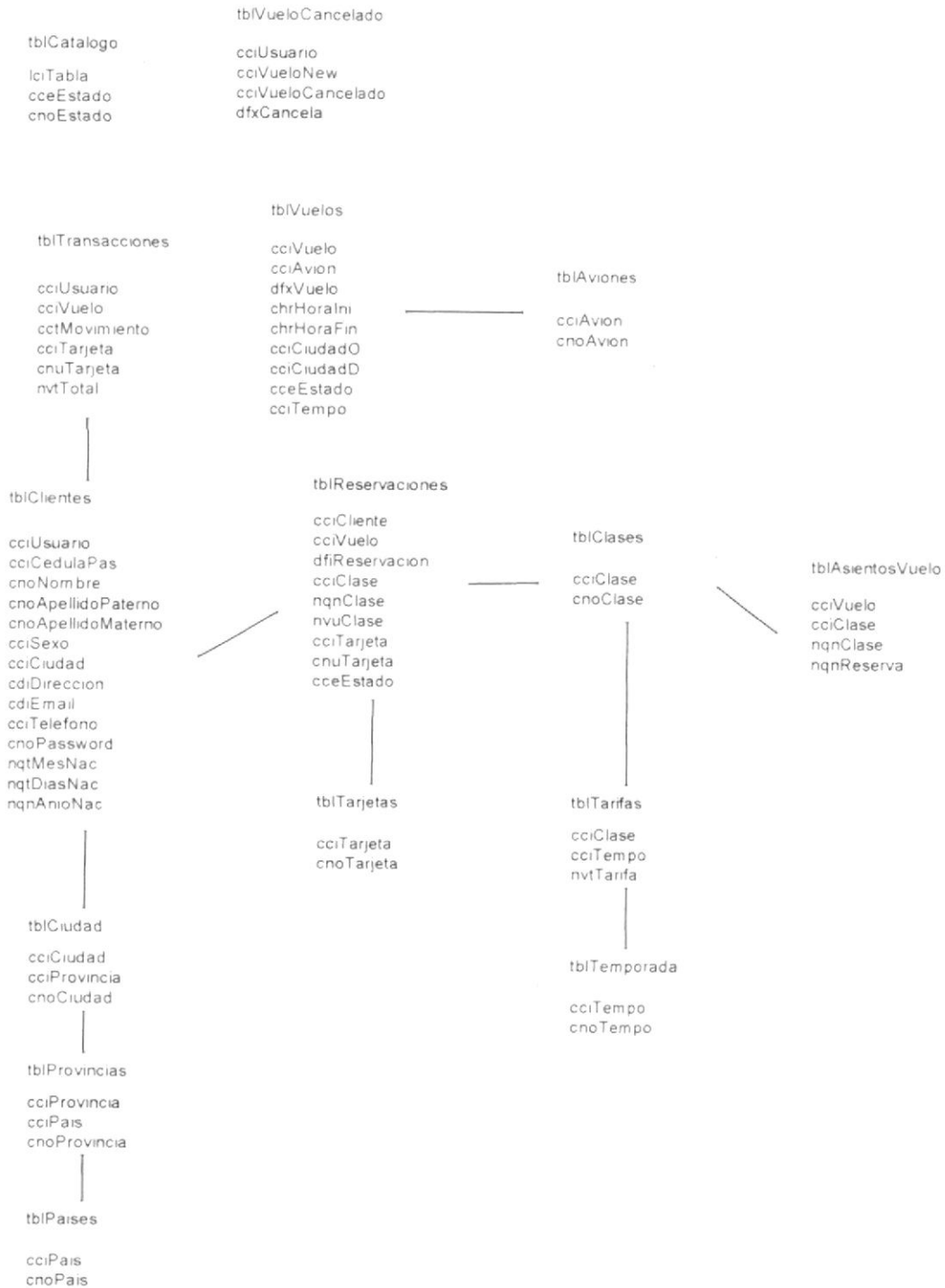
Capítulo 5

5. IMPLEMENTACIÓN DE LA CAPA DE DATOS


5.1. Modelo Lógico de Datos.




5.2. Modelo Físico de Datos.




5.2.1 Tabla de asientos de vuelos.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gaviláñez Sonia Hurtado	1/15	
Nombre Lógico: Asientos de Vuelo		Nombre Físico: tblAsientosVuelo		
Clave Primaria: PK_tblAsientosVuelo (cciVuelo, cciClase)				
Clave Foránea:				
FK_tblAsientosVuelo_tblClases → tblClases (cciClase)				
FK_tblAsientosVuelo_tblVuelos → tblVuelos (cciVuelo)				
Descripción				
Mantiene información de los asientos disponibles para reservar un vuelo.				
Campo	Tipo	Long.	Nulo	Descripción
cciVuelo	nvarchar	8	No	Código del Vuelo
cciClase	nvarchar	4	No	Código de la Clase
nqnClase	smallint	5	No	Cantidad de Asientos de la Clase
nqnReserva	smallint	5	No	Cantidad de Asientos reservados de la Clase.


5.2.2 Tabla de aviones.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	2/15	
Nombre Lógico: Aviones			Nombre Físico: tblAviones	
Clave Primaria: PK_tblAviones_1(cciAvion)				
Clave Foránea:				
Descripción				
Mantiene información de los aviones con que cuenta la Empresa.				
Campo	Tipo	Long.	Nulo	Descripción
cciAvion	nvarchar	8	No	Código del Avión
cnoAvion	char	20	No	Nombre del Avión
cdsAvion	char	100	Si	Descripción del Avión
cceEstado	char	1	No	Estado del Avión. (A) Activo (I) Inactivo (M) Mantenimiento


5.2.3 Tabla de catálogo.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservaciones de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	3/15	
Nombre Lógico: Catálogo		Nombre Físico: tblCatalogo		
Clave Primaria:				
Clave Foránea:				
Descripción				
Mantiene información de los estados del sistema (Aviones, vuelos).				
Campo	Tipo	Long.	Nulo	Descripción
lciTabla	char	1	No	Código identificador del catálogo
cceEstado	char	1	No	Nombre del Estado
cnoEstado	char	20	No	Descripción del Estado


5.2.4 Tabla de ciudades.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	4/15	
Nombre Lógico: Ciudad			Nombre Físico: tblCiudad	
Clave Primaria: PK_tblCiudad (cciCiudad)				
Clave Foránea: FK_tblCiudad_tblProvincias → tblProvincias (cciProvincia)				
Descripción Mantiene información de las Ciudades.				
Campo	Tipo	Long.	Nulo	Descripción
cciCiudad	nvarchar	4	No	Código de la Ciudad.
cciProvincia	nvarchar	4	No	Código de la Provincia
cnoCiudad	nvarchar	20	No	Nombre de la Ciudad


5.2.5 Tabla de clases.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	5/15	
Nombre Lógico: Clases		Nombre Físico: tblClases		
Clave Primaria: PK_tblClases (cciClase)				
Clave Foránea:				
Descripción				
Mantiene información de las Clases				
Campo	Tipo	Long.	Nulo	Descripción
cciClase	nvarchar	4	No	Código de la Clase.
cnoClase	varchar	20	No	Nombre de la Clase.


5.2.6 Tabla de países.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	6/15	
Nombre Lógico: Países		Nombre Físico: tblPaíses		
Clave Primaria: PK_tblPaíses (cciPaís)				
Clave Foránea:				
Descripción				
Mantiene información de los Países.				
Campo	Tipo	Long.	Nulo	Descripción
cciPaís	nvarchar	4	No	Código del País.
cnoPaís	nvarchar	20	No	Nombre del País.

5.2.7 Tabla de clientes.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	7/15	
Nombre Lógico: Clientes		Nombre Físico: tblClientes		
Clave Primaria: PK_tblClientes (cciUsuario)				
Clave Foránea: FK_tblClientes_tblCiudad → tblCiudad (cciCiudad)				
Descripción Mantiene información de las Clientes.				
Campo	Tipo	Long.	Nulo	Descripción
cciUsuario	nvarchar	10	No	Código del Usuario/Cliente
cciCedulaPas	nvarchar	20	No	Cédula del Cliente
cnoNombre	varchar	50	No	Nombre del Cliente
cnoApellidoPaterno	varchar	50	No	Apellido Paterno del Cliente
cnoApellidoMaterno	varchar	50	No	Apellido Materno del Cliente
cciSexo	char	1	No	Sexo (M) Masculino (F) Femenino
cciCiudad	nvarchar	4	No	Código de la Ciudad
cdiDireccion	nvarchar	50	No	Dirección del Cliente
cdiEmail	varchar	30	No	Dirección E-mail
cciTelefono	nvarchar	15	No	Teléfono del Cliente
cnoPassword	varchar	10	No	Password del Cliente
nqtMesNac	varchar	2	No	Mes de Nacimiento
nqtDiaNac	varchar	2	No	Día de Nacimiento
nqnAnioNac	varchar	4	No	Año de Nacimiento


5.2.8 Tabla de provincias.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor		Página
Sistema de Reservaciones de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado		8/15
Nombre Lógico: Provincias			Nombre Físico: tblProvincias	
Clave Primaria: PK_tblProvincias (cciProvincia)				
Clave Foránea:				
FK_tblProvincias_tblPaises → tblPaises (cciPais)				
Descripción				
Mantiene información de las Provincias.				
Campo	Tipo	Long.	Nulo	Descripción
cciProvincia	nvarchar	4	No	Código de la Provincia.
cciPais	nvarchar	4	No	Código del País
cnoProvincia	nvarchar	20	No	Nombre de la Provincia


5.2.9 Tabla de reservaciones.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservaciones de Vuelo (AIRCONDOR)	FLIGHTS	Maria Gavilánez Sonia Hurtado	9/15	
Nombre Lógico: Reservaciones		Nombre Físico: tblReservaciones		
Clave Primaria: PK_tblReservaciones (cciCliente, cciVuelo)				
Clave Foránea:				
FK_tblReservaciones_tblClases → tblClases (cciClase)				
FK_tblReservaciones_tblClientes → tblClientes (cciCliente)				
FK_tblReservaciones_tblTarjetas → tblTarjetas (cciTarjeta)				
Descripción				
Mantiene información de las reservaciones efectuadas por el Cliente.				
Campo	Tipo	Long.	Nulo	Descripción
cciCliente	nvarchar	20	No	Código del Cliente
cciVuelo	nvarchar	16	No	Código del Vuelo
dfiReservacion	datetime	8	No	Fecha de la Reservación.
cciClase	nvarchar	8	No	Código de la Clase
nqnClase	smallint	2	No	Número de Asientos
nvuClase	decimal	9	No	Tarifa de la Clase
cciTarjeta	nvarchar	16	No	Código de la Tarjeta
cnuTarjeta	nvarchar	40	No	Número de Tarjeta
cceEstado	char	1	No	Estado (A) Activo (I) Inactivo (C) Cancelado (E) Efectuado


5.2.10 Tabla de transacciones.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	10/15	
Nombre Lógico: Transacciones		Nombre Físico: tblTransacciones		
Clave Primaria: PK_tblTransacciones (cciUsuario, cciVuelo, cctMovimiento)				
Clave Foránea: FK_tblTransacciones_tblClientes → tblClientes (cciUsuario)				
Descripción: Mantiene información de las transacciones efectuadas por el Cliente.				
Campo	Tipo	Long.	Nulo	Descripción
cciUsuario	nvarchar	10	No	Usuario/Cliente de la Transacción.
cciVuelo	nvarchar	8	No	Código de Vuelo
cctMovimiento	char	1	No	Tipo de Movimiento (D) Débito (C) Crédito.
cciTarjeta	nvarchar	8	No	Código de la Tarjeta
cnuTarjeta	nvarchar	20	No	Número de Tarjeta de Crédito.
nvtTotal	decimal	17(2)	No	Valor Total de la Transacción.


5.2.11 Tabla de tarifas.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	11/15	
Nombre Lógico: Tarifas			Nombre Físico: tblTarifas	
Clave Primaria: PK_tblTarifas (cciClase, cciTempo)				
Clave Foránea:				
FK_tblTarifas_tblClases → tblClases (cciClase)				
FK_tblTarifas_tblTemporada → tblTemporada (cciTempo)				
Descripción				
Mantiene información de las Tarifas.				
Campo	Tipo	Long.	Nulo	Descripción
cciClase	nvarchar	4	No	Código de la Clase .
cciTempo	nvarchar	4	No	Código de la Temporada.
nvtTarifa	decimal	17(2)	No	Nombre de la Provincia


5.2.12 Tabla de tarjetas.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor		Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado		12/15
Nombre Lógico: Tarjetas			Nombre Físico: tblTarjetas	
Clave Primaria: PK_tblTarjetas (cciTarjeta)				
Clave Foránea:				
Descripción				
Mantiene información de las Tarjetas de Crédito del Sistema.				
Campo	Tipo	Long.	Nulo	Descripción
cciTarjeta	nvarchar	8	No	Código de la Tarjeta.
cnoTarjeta	nvarchar	20	No	Nombre de la Tarjeta.


5.2.13 Tabla de temporadas

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	13/15	
Nombre Lógico: Temporadas		Nombre Físico: tblTemporada		
Clave Primaria: PK_tblTemporada (cciTempo)				
Clave Foránea:				
Descripción				
Mantiene información de las temporadas turísticas.				
Campo	Tipo	Long.	Nulo	Descripción
cciTempo	nvarchar	4	No	Código de la Temporada.
cnoTempo	varchar	10	No	Nombre de la Temporada.

5.2.14 Tabla de vuelos cancelados.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservasiones de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	14/15	
Nombre Lógico: Vuelos Cancelados		Nombre Físico: tblVueloCancelado		
Clave Primaria: PK_tblVueloCancelado (cciUsuario, cciVueloNew, ccivueloCancelado)				
Clave Foránea:				
FK_tblVueloCancelado_tblVuelos → tblVuelos (cciVueloNew)				
FK_tblVueloCancelado_tblVuelos1 → tblVuelos (ccivueloCancelado)				
Descripción				
Mantiene información de los vuelos cancelados.				
Campo	Tipo	Long.	Nulo	Descripción
cciUsuario	nvarchar	10	No	Código del Usuario.
cciVueloNew	nvarchar	8	No	Código del Nuevo Vuelo.
ccivueloCancelado	nvarchar	8	No	Código del Vuelo Cancelado.
dfxCancela	datetime	8	No	Fecha de cancelación.

5.2.15 Tabla de vuelos.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información				
Proyecto	Base de Datos	Autor	Página	
Sistema de Reservasiones de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	15/15	
Nombre Lógico: Vuelos		Nombre Físico: tblVuelos		
Clave Primaria: PK_tblVuelos (cciVuelo)				
Clave Foránea: FK_tblVuelos_tblAviones → tblAviones (cciAvion)				
Descripción Mantiene información de los vuelos registrados.				
Campo	Tipo	Long.	Nulo	Descripción
cciVuelo	nvarchar	8	No	Código del Vuelo
cciAvion	nvarchar	8	No	Código del Avión
dfxVuelo	datetime	8	No	Fecha de vuelo.
chrHoraIni	nvarchar	16	No	Hora de salida.
chrHoraFin	nvarchar	16	No	Hora de llegada.
cciCiudadO	nvarchar	8	No	Ciudad Origen
cciCiudadD	nvarchar	8	No	Ciudad Destino
cceEstado	char	1	No	Estado (A) Activo (C) Cancelado (E) Efectuado.
cciTempo	nvarchar	4	No	Temporada del vuelo.

5.3. Procedimientos Almacenados (Stored Procedures).


5.3.1 Procedimiento: spr_CancelaVuelo.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gaviláñez Sonia Hurtado	1/22
Nombre Físico: spr_CancelaVuelo			
Descripción			
Realiza el proceso de cancelación de un vuelo reservado por el cliente.			
N°	Parámetro	Tipo	Longitud
1	@cciCliente	nvarchar	10
2	@cciVuelo	nvarchar	8


5.3.2 Procedimiento: spr_Clientes.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	2/22
Nombre Físico: spr_Clientes			
Descripción			
Consulta información para el reporte de clientes.			
N°	Parámetro	Tipo	Longitud
1	@ComputerName	nvarchar	30


5.3.3 Procedimiento: spr_ObtenerDatosReserva.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	Maria Gavilánez Sonia Hurtado	3/22
Nombre Físico: spr_ObtenerDatosReserva			
Descripción			
Consulta información de la reservación efectuada por un cliente en un vuelo específico.			
N°	Parámetro	Tipo	Longitud
1	@cciUsuario	nvarchar	10
2	@cciVuelo	nvarchar	8
3	@cciClase (OUT)	nvarchar	4
4	@nvtPago (OUT)	decimal	17(2)
5	@nqnAsientos (OUT)	smallint	5
6	@cciTarjeta (OUT)	nvarchar	8
7	@cnuTarjeta (OUT)	nvarchar	20

5.3.4 Procedimiento: spr_ObtenerTarifas.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	4/22
Nombre Físico: spr_ObtenerTarifas			
Descripción			
Obtiene información de la tarifa para un vuelo de acuerdo a una clase especial.			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8
2	@cciClase	nvarchar	4
3	@nvtTarifa (OUT)	decimal	17(2)


5.3.5 Procedimiento: spr_VerificaDisponible.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	5/22
Nombre Físico: spr_VerificaDisponible			
Descripción			
Verifica la disponibilidad de asientos en un vuelo de acuerdo a una clase específica.			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8
2	@cciClase	nvarchar	4
3	@nqnReserva	smallint	5
4	@nqnClase (OUT)	smallint	5


5.3.6 Procedimiento: spr_VerificaReservaciones.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservaciones de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	6/22
Nombre Físico: spr_VerificaReservaciones			
Descripción			
Verifica si existen reservaciones activas en un vuelo.			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8
2	@cceEstado (OUT)	char	1


5.3.7 Procedimiento: spr_VuelosCancelaALL.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	7/22
Nombre Físico: spr_VuelosCancelaALL			
Descripción			
Consulta información para el reporte de vuelos cancelados.			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8
2	@ComputerName	nvarchar	30


5.3.8 Procedimiento: sprActualizaEstado.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	8/22
Nombre Físico: sprActualizaEstado			
Descripción			
Actualiza el estado de los vuelos en el proceso de cancelación.			
N°	Parámetro	Tipo	Longitud
1	@sActualiza	char	1
2	@Vuelo	nvarchar	40
3	@VueloCancel	nvarchar	40


5.3.9 Procedimiento: sprl_Clientes.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	9/22
Nombre Físico: sprl_Clientes			
Descripción Realiza el proceso de inserción de la información del cliente.			
N°	Parámetro	Tipo	Longitud
1	@cciUsuario	nvarchar	10
2	@cciCedulaPas	nvarchar	20
3	@cnoNombre	varchar	50
4	@cnoApellidoPaterno	varchar	50
5	@cnoApellidoMaterno	varchar	50
6	@cciSexo	char	1
7	@cciCiudad	nvarchar	4
8	@cdiDireccion	char	50
9	@cdiEmail	varchar	30
10	@cciTelefono	nvarchar	15
11	@cnoPassword	varchar	10
12	@nqtMesNac	varchar	2
13	@nqtDiaNac	varchar	2
14	@nqnAnioNac	varchar	4


5.3.10 Procedimiento: sprE_Reservacion.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservaciones de Vuelo (AIRCONDOR)	FLIGHTS	María Gaviláñez Sonia Hurtado	10/22
Nombre Físico: sprE_Reservacion			
Descripción			
Actualiza las reservaciones de un vuelo específico a un estado inactivo.			
N°	Parámetro	Tipo	Longitud
1	@cciUsuario	nvarchar	10
2	@cciVuelo	nvarchar	8


5.3.11 Procedimiento: sprl_Reservación.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservaciones de Vuelo (AIRCONDOR)	FLIGHTS	María Gaviláñez Sonia Hurtado	11/22
Nombre Físico: sprl_Reservacion			
Descripción			
Inserta un registro en la tabla de reservaciones para un cliente específico.			
N°	Parámetro	Tipo	Longitud
1	@cciCliente	nvarchar	10
2	@cciVuelo	nvarchar	8
3	@cciClase	nvarchar	4
4	@cciTarjeta	nvarchar	8
5	@cnuTarjeta	nvarchar	20
6	@nqnReserva	smallint	5
7	@nvtTarifa	decimal	17 (2)


5.3.12 Procedimiento: sprl_Reservaciones.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservaciones de Vuelo (AIRCONDOR)	FLIGHTS	Maria Gavilánez Sonia Hurtado	12/22
Nombre Físico: sprl_Reservaciones			
Descripción			
Realiza el proceso de reservación de un vuelo para un cliente, incluye actualizaciones de asientos, inserción de reservación e inserción de la transacción (tarjeta de crédito).			
N°	Parámetro	Tipo	Longitud
1	@cciCliente	nvarchar	10
2	@cciVuelo	nvarchar	8
3	@cciClase	nvarchar	4
4	@cciTarjeta	nvarchar	8
5	@cnuTarjeta	nvarchar	20
6	@nqnReserva	smallint	5


5.3.13 Procedimiento: sprl_Transacción.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	13/22
Nombre Físico: sprl_Transaccion			
Descripción			
Inserta un registro en la tabla de transacciones un histórico de los pagos / reembolsos del cliente.			
N°	Parámetro	Tipo	Longitud
1	@cciUsuario	nvarchar	10
2	@cciVuelo	nvarchar	8
3	@cctMovimiento	char	1
4	@cciTarjeta	nvarchar	8
5	@cnuTarjeta	nvarchar	20
6	@nvtTotal	decimal	17(2)


5.3.14 Procedimiento: sprP_VueloEfectuado.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	14/22
Nombre Físico: sprP_VueloEfectuado			
Descripción			
Actualiza el estado de la reservación y del vuelo a eliminado.			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8


5.3.15 Procedimiento: sprR_VuelosCancelWeb.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gaviláñez Sonia Hurtado	15/22
Nombre Físico: sprR_VuelosCancelWeb			
Descripción			
Consulta información de las reservas canceladas en el WEB de acuerdo a un vuelo específico.			
N°	Parámetro	Tipo	Longitud
1	@cciUsuario	nvarchar	10
2	@cciVuelo	nvarchar	8


5.3.16 Procedimiento: sprR_VuelosReservaWeb.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	16/22
Nombre Físico: sprR_VuelosReservaWeb			
Descripción			
Consulta información las reservas efectuadas en el WEB de acuerdo a un vuelo específico.			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8
2	@ComputerName	nvarchar	30

5.3.17 Procedimiento: sprU_Asientos.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gaviláñez Sonia Hurtado	17/22
Nombre Físico: sprU_Asientos			
Descripción			
Actualiza la cantidad de asientos disponibles para un vuelo y una clase específica (reservación de vuelos).			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8
2	@cciClase	nvarchar	4
3	@nqnReserva	smallint	5


5.3.18 Procedimiento: sprU_AsientosC.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	18/22
Nombre Físico: sprU_AsientosC			
Descripción			
Actualiza la cantidad de asientos disponibles para un vuelo y una clase específica. (cancelación de vuelos).			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8
2	@cciClase	nvarchar	4
3	@nqnReserva	smallint	5


5.3.18 Procedimiento: sprU_Reservación.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	18/22
Nombre Físico: sprU_Reservacion			
Descripción			
Establece todas las reservas activas de un vuelo a estado ELIMINADO.			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8


5.3.20 Procedimiento: sprU_Clientes.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	Maria Gavilánez Sonia Hurtado	20/22
Nombre Físico: sprU_Clientes			
Descripción			
Actualiza la información del cliente.			
N°	Parámetro	Tipo	Longitud
1	@cciUsuario	nvarchar	10
2	@cciCedulaPas	nvarchar	20
3	@cnoNombre	varchar	50
4	@cnoApellidoPaterno	varchar	50
5	@cnoApellidoMaterno	varchar	50
6	@cciSexo	char	1
7	@cciCiudad	nvarchar	4
8	@cdiDireccion	char	50
9	@cdiEmail	varchar	30
10	@cciTelefono	nvarchar	15
11	@cnoPassword	varchar	10
12	@nqtMesNac	varchar	2
13	@nqtDiaNac	varchar	2
14	@nqnAnioNac	varchar	4

5.3.21 Procedimiento: sprU_Vuelo.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gavilánez Sonia Hurtado	21/22
Nombre Físico: sprU_Vuelo			
Descripción			
Establece a un vuelo específico el estado ELIMINADO.			
N°	Parámetro	Tipo	Longitud
1	@cciVuelo	nvarchar	8

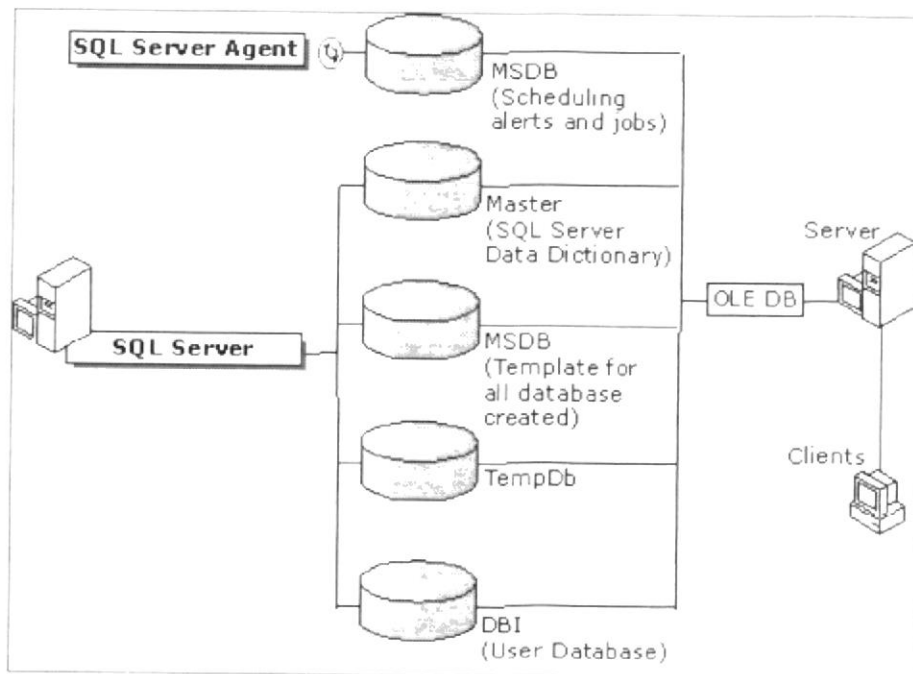
5.3.22 Procedimiento: sprVueloCancelado.

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Base de Datos	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)	FLIGHTS	María Gaviláñez Sonia Hurtado	22/22
Nombre Físico: sprVueloCancelado			
Descripción			
Realiza el proceso de cambio de un vuelo a otro para un cliente específico. (cuando se cancela un vuelo y se transfiere a los pasajeros de un vuelo a otro.)			
N°	Parámetro	Tipo	Longitud
1	@Cliente	nvarchar	20
2	@Vuelo	nvarchar	40
3	@VueloCancel	nvarchar	40
4	@Reservacion	datetime	8
5	@Clase	nvarchar	8
6	@CantClase	smallint	5
7	@vClase	decimal	17(2)
8	@CodTarjeta	nvarchar	16
9	@numTarjeta	nvarchar	40
10	@cceEstado	char	1

5.4. Proceso Servidor de Base de Datos.

En el Sistema de Reservas de Vuelos (AIRCONDOR) la administración de la Base de Datos y del acceso a los componentes de datos es Microsoft SQL SERVER 7.0.

Este DataBase Server permite la ejecución de procedimientos almacenados (stored procedures), los mismos que fueron escritos en Lenguaje SQL y se encuentran precompilados y almacenados en la Base de Datos.



Esquema de acceso a SQL SERVER.

Capítulo 6

6. IMPLEMENTACION DE LA CAPA DE NEGOCIOS.

6.1 Estructura de los Componentes de Acceso a Datos.

En el sistema de reservaciones Aircondor.com utilizamos un componente central de acceso a datos, llamado "Servidor.dll", el cual es el contenedor de todas las clases que se encargan de acceder directamente a la base de datos.

6.2 Objetos distribuidos de acceso a datos.

Las clases por las que está compuesto el componente "Servidor.dll" son:

- ✈ Clientes
- ✈ Consulta

✈ Reservaciones

Los métodos que contienen cada una de éstas clases son:

<i>Cientes</i>	InsertarCliente
	ActualizarCliente
<i>Consulta</i>	EjecutarConsulta
<i>Reservaciones</i>	IngresarReservacion
	CancelarReservacion

6.3 Comunicación con el proceso servidor de acceso a datos.

El sistema de Aircondor.com utiliza como middleware para la comunicación con el proceso servidor de datos, COM. (para más información véase Apéndice).


Otro tipo de middleware empleado por el sistema es Microsoft ActiveX Data Objects (ADO) como la interfaz para el acceso a los datos permitiendo la llamada a los procedimientos almacenados creados en Microsoft SQL Server.


6.4 Estructura de los componentes de negocios.


En Aircondor.com se realizó la agrupación de las clases de negocios en un solo componente llamado "Servidor".

6.5 Objetos distribuidos de negocios.

El componente "Servidor" esta conformado por las siguientes clases:

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Descripción de componentes de negocio.	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)		María Gaviláñez Sonia Hurtado	1/3
Clase: Clientes			
Descripción			
Se encarga del mantenimiento de la información de los clientes o usuarios.			
Método	Objeto de datos asociado	Descripción	
InsertarCliente	sprI_Clientes	Permite el ingreso de información de los usuarios del sitio web que posteriormente se convertirán en clientes	
ActualizarCliente	sprU_Clientes	Permite la actualización de la información del usuario.	

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Descripción de componentes de negocio.	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)		María Gavilánez Sonia Hurtado	2/3
Clase: Consulta			
Descripción			
Ejecuta cualquier consulta enviada como parámetro (cadena) en sentencia SQL.			
Método	Descripción		
EjecutarConsulta	Obtiene información de la base de datos a partir de cualquier sentencia SQL. (no referencia a ningún objeto de datos en especial).		

 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Licenciatura en Sistemas de Información			
Proyecto	Descripción de componentes de negocio.	Autor	Página
Sistema de Reservas de Vuelo (AIRCONDOR)		María Gavilánez Sonia Hurtado	3/3
Clase: Reservas			
Descripción			
Se encarga del manejo de transacciones de reservación.			
Método	Objeto de datos asociado	Descripción	
IngresarReservacion.	sprl_Reservaciones	Registra la reservación solicitada por el usuario en el sitio web.	
CancelarReservacion	spr_CancelaVuelo	Registra la cancelación de un vuelo para un usuario específico.	

IDL Generado

```
[ uuid(7CBFF9CB-6276-11D6-B4CA-00D009F3EF58),  
  version(1.0)  
]  
library Servidor  
{ importlib("msado15.dll");  
  importlib("StdOle2.Tlb");  
  interface Clientes;  
  interface Consulta;  
  interface Reservasiones:  
  [  
    odl,  
    uuid(2CF1C95A-68A5-11D6-B4D3-00D009F3EF58),  
    version(1.0),  
    dual,  
    nonextensible,  
    oleautomation  
  ]  
}
```

```

interface Clientes : IDispatch {
    [id(0)]
    HRESULT ActualizarCliente(
        [in] BSTR DSN,
        [in] BSTR cciUsuario,
        [in] BSTR cciCedulaPas,
        [in] BSTR cnoNombre,
        [in] BSTR cnoApellidoPaterno,
        [in] BSTR cnoApellidoMaterno,
        [in] BSTR cciSexo,
        [in] BSTR cciCiudad,
        [in] BSTR cdiDireccion,
        [in] BSTR cdiEmail,
        [in] BSTR cciTelefono,
        [in] BSTR cnoPassword,
        [in] BSTR nqtMesNac,
        [in] BSTR nqtDiaNac,
        [in] BSTR nqnAnioNac,
        [out, retval] long* );

    [id(1)]
    HRESULT InsertarCliente(
        [in] BSTR DSN,
        [in] BSTR cciUsuario,
        [in] BSTR cciCedulaPas,
        [in] BSTR cnoNombre,
        [in] BSTR cnoApellidoPaterno,
        [in] BSTR cnoApellidoMaterno,
        [in] BSTR cciSexo,
        [in] BSTR cciCiudad,
        [in] BSTR cdiDireccion,
        [in] BSTR cdiEmail,
        [in] BSTR cciTelefono,
        [in] BSTR cnoPassword,
        [in] BSTR nqtMesNac,
        [in] BSTR nqtDiaNac,
        [in] BSTR nqnAnioNac,
        [out, retval] long* );
};

```

```
[ odl,
  uuid(C53AF239-64E0-11D6-B4CF-00D009F3EF58),
  version(1.0),
  dual,
  nonextensible,
  oleautomation
]
```

```
interface Consulta : IDispatch {
  [id(0)]
  HRESULT EjecutarConsulta(
    [in] BSTR DSN,
    [in] BSTR SentenciaSQL,
    [out, retval] _Recordset** );
}
```

```
[ odl,
  uuid(22B4990A-84A6-11D6-B50B-00D009F3EF58),
  version(1.1),
  dual,
  nonextensible,
  oleautomation
]
```

```
interface _Reservaciones : IDispatch {
  [id(0)]
  HRESULT IngresarReservacion(
    [in] BSTR DSN,
    [in] BSTR cciCliente,
    [in] BSTR cciVuelo,
    [in] BSTR cciClase,
    [in] BSTR cciTarjeta,
    [in] BSTR cnuTarjeta,
    [in] short nqnReserva,
    [out, retval] long* );

  [id(1)]
  HRESULT CancelarReservacion(
    [in] BSTR DSN,
    [in] BSTR cciCliente,
    [in] BSTR cciVuelo,
    [out, retval] long* ); };
```

6.6 Comunicación con el proceso servidor de acceso a datos.

En el sistema de reservaciones Aircondor.com se utiliza como middleware para la comunicación con los componentes de acceso a datos, COM. (para más información véase Apéndice).

Además otro tipo de middleware empleado es ActiveX Data Objects (ADO) como la interfaz para la invocación de los métodos de consulta y luego recibir como dato de retorno los Recordset devueltos por los componentes de acceso a datos con los registros recuperados.

6.7 Proceso servidor transaccional.

En Aircondor.com se manejan las siguientes transacciones:

6.7.1 Reservación de vuelos.

La transacción inicia consultando si el cliente tiene un usuario y contraseña registrado en el sistema, de no ser así deberá registrarse.

Una vez verificado el acceso se le permite al usuario seleccionar las características del vuelo a realizar como son: si desea realizar un viaje de ida y vuelta, el lugar origen y el destino y la fecha de salida y de retorno (si es un viaje de regreso).

Si existe la disponibilidad en el vuelo el usuario especifica la cantidad de asientos que desea con respecto a una clase y la tarjeta de crédito a utilizar. Establecida la información necesaria se registra la reservación, se disminuye la cantidad de asientos disponibles y se registra una transacción con débito a la tarjeta escogida para el pago y el valor (obtenido a partir de la temporada del vuelo y la clase). Finalmente se emite el comprobante de pago (ticket virtual).

El objeto involucrado en la transacción es ***Servidor.Reservaciones.IngresarReservacion.***

6.7.2 Cancelación de vuelos.

Al igual que la reservación de vuelos, la transacción inicia consultando si el cliente tiene un usuario y contraseña registrado en el sistema, de no ser así deberá registrarse.

Una vez que se verifica el acceso el usuario se consultan los vuelos activos que aún pueden ser cancelados. El usuario selecciona el vuelo a cancelar.

Establecido todos los parámetros se procede a la cancelación que consiste en cambiar el estado de la reservación, se actualiza la disponibilidad de asientos en el vuelo según la clase y se genera una transacción con crédito

a la tarjeta que haya utilizado en la reservación. Finalmente se emite una nota de crédito

Los objetos involucrados en la transacción son los siguientes:

El objeto involucrado en la transacción es ***Servidor.Reservaciones.CancelarReservacion.***

6.7.3 Registro de usuario.

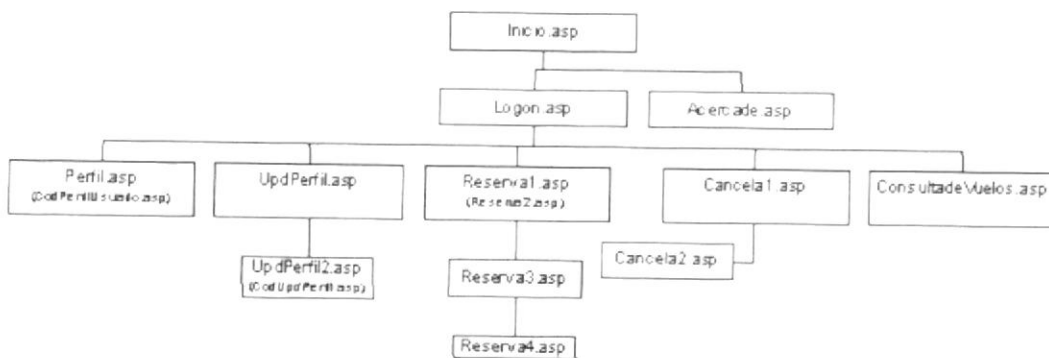
La transacción permite el registro del perfil de usuario en Aircondor.com una vez ingresada la información necesaria (obligatoria).

El objeto involucrado en la transacción es ***Servidor.Clientes.InsertarCliente*** y ***Servidor.Clientes.ActualizarCliente.***

Capítulo 7

7. IMPLEMENTACION DE LA CAPA DE PRESENTACION

7.1 Diseño del sitio web-cliente basado en browser.



7.2 Diseño del sitio web-cliente basado en browser.

Aircondor.com ha implementado un sistema stand-alone utilizando como herramienta de desarrollo a Microsoft Visual Basic 6.0, dentro del cual se

maneja la parte administrativa del sitio, como el mantenimiento de catálogos, transacciones, procesos y reportes.

Dentro de la sección de catálogos existen las siguientes opciones:

- ✈ Países
- ✈ Provincias
- ✈ Ciudades
- ✈ Clases
- ✈ Temporadas
- ✈ Tarjetas de Crédito
- ✈ Aviones

Las transacciones son las siguientes:

- ✈ Tarifas
- ✈ Vuelos
- ✈ Asientos de Vuelos

Los procesos son los siguientes:

- ✈ Cancelación de vuelos
- ✈ Procesar Vuelos efectuados

Los reportes son los siguientes:

- ✈ Vuelos Reservados (WEB)
- ✈ Vuelos Cancelados (WEB)
- ✈ Vuelos Cancelados
- ✈ Clientes
- ✈ Transacciones de tarjetas.

7.3 Cliente HTML.

En Aircondor.com, el código HTML utilizado se encuentra embebido dentro del ASP ya que en el sitio solo se implementaron éste tipo de páginas.

7.4 Cliente Script - DHTML.

En Aircondor.com se implementó Javascript en las validaciones realizadas en el lado del cliente ya que permite flexibilidad en el uso del browser (Internet Explorer o Netscape Navigator).

7.5 Server Script - ASP.

En Aircondor.com todas las páginas fueron desarrolladas en ASP (Active Server Pages), ya que es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente

Las páginas ASP (Active Server Pages) permiten combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web.

7.6 CSS.

En Aircondor.com se utilizó hojas de estilo para dar formato de presentación en todas las páginas (cabeceras, textos, cajas de texto, botones, tablas, etc.)

De tal manera que al cambiar el formato de las páginas solo se modificará el archivo de hojas de estilos.

7.7 Comunicación con componentes de negocio.

A continuación se detalla la comunicación que existe entre las páginas ASP y los componentes de negocio:

Relación Páginas ASP con los componentes de negocio.	
Cancela1.asp	Servidor.reservaciones Servidor.consulta
Cancela2.asp	Servidor.consulta
CodPerfilUsuario.asp	Servidor.clientes Servidor.consulta
CodUpdPerfil.asp	Servidor.clientes Servidor.consulta
ConsultadeVuelos.asp	Servidor.consulta
Reserva1.asp	Servidor.consulta
Reserva2.asp	Servidor.consulta
Reserva3.asp	Servidor.reservaciones Servidor.consulta
Reserva4.asp	Servidor.consulta
UpdPerfil2.asp	Servidor.consulta

7.8 Proceso Servidor WEB.

El proceso servidor web de Aircondor.com es el Internet Information Server. Este permite el desarrollo de aplicaciones de negocio basadas en Web que se pueden extender a Internet o distribuir en intranets corporativas. Internet Information Server, introduce un nuevo concepto en Internet; las

aplicaciones transaccionales. Las transacciones son los conductos que hacen posible ejecutar aplicaciones reales de negocio con desarrollos rápidos, escalables y fiables.

Para Aircondor.com se creó un sitio virtual con nombre de CONDOR, en donde el documento principal a ejecutarse es Inicio.asp.

CONCLUSIONES

Las Empresas de Servicios frente a los avances tecnológicos como es el "Internet", ve la oportunidad de cubrir la necesidad de tener una comunicación más rápida y directa con los clientes, ofreciéndoles un proceso de reservación de vuelos más rápido y eficiente. El presente trabajo ha sido desarrollado con el propósito de suplir estas necesidades, tanto a las empresas de servicios como a sus clientes.

En el desarrollo del sitio Aircondor.com, nuestro propósito ha sido cumplir con todas las expectativas planteadas por Aerolíneas Cónдор a lo largo de este documento.

RECOMENDACIONES

Entre las recomendaciones más importantes podemos enumerar las siguientes:

- ✈ Tener una persona encargada para la administración del Sitio Web.
- ✈ Respaldo Semanalmente los Datos, así como el respaldo de los fuentes.
- ✈ Mantener el estándar en las nuevas páginas que se adhieran al sitio web como al sistema administrativo.
- ✈ Mantener actualizada la información de Aircondor.com mediante el Sistema Administrativo.

BIBLIOGRAFÍA

[http://www.terra.com.ve/informatica/ que-es](http://www.terra.com.ve/informatica/que-es)

<http://www.microsoft.com>

<http://www.galileo.com>

<http://www.sabre.com>

Apéndice A

ARQUITECTURA CLIENTE / SERVIDOR.

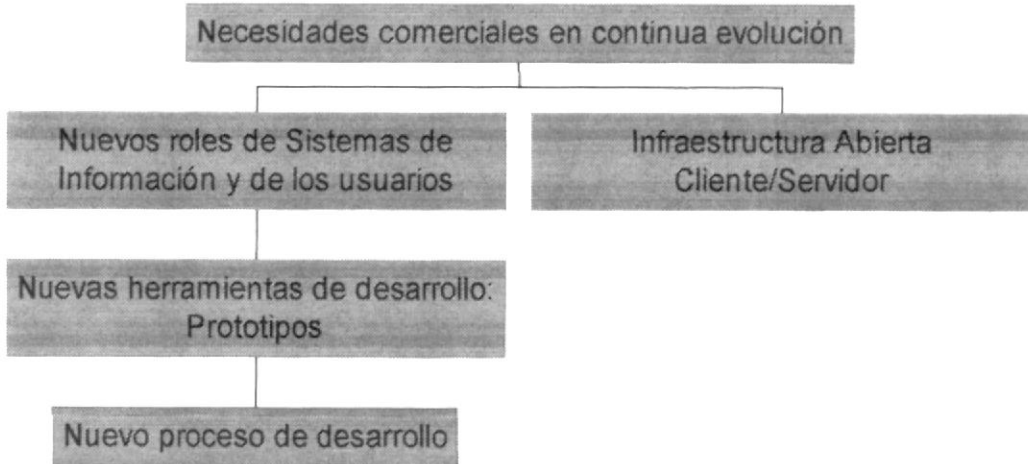
INTRODUCCIÓN

Toda aplicación trata de reflejar parte del funcionamiento del mundo real, para automatizar tareas que de otro modo serían llevadas a cabo de modo más ineficiente, o bien no podrían realizarse. Para ello, es necesario que cada aplicación refleje las restricciones que existen en el negocio dado, de modo que nunca sea posible llevar a cabo acciones no válidas. A las reglas que debe seguir la aplicación para garantizar esto se las llama *reglas de negocio*, o *business rules*. Ejemplos de tales reglas son: no permitir crear facturas pertenecientes a clientes inexistentes, controlar que el saldo negativo de un cliente nunca sobrepase cierta cantidad, etc.

En realidad, la información puede ser manipulada por muchos programas distintos: así, una empresa puede tener un departamento de contabilidad que controle todo lo relacionado con compras, cobros, etc., y otro departamento técnico, que esté interesado en relacionar diversos parámetros de producción con los costes. La visión que ambos departamentos tendrán de la información y sus necesidades serán distintas, pero en cualquier caso siempre se deberán respetar las reglas de negocio. El hecho de que la información sea manipulada por diversos programas hace más difícil garantizar que todos respetan las reglas, especialmente si

las aplicaciones corren en diversas máquinas, bajo distintos sistemas operativos, y están desarrolladas con distintos lenguajes y herramientas.

Implicaciones del modelo Cliente/Servidor



Capas en un sistema Cliente/Servidor

En un esquema Cliente/Servidor clásico (**Figura A**) existen dos capas, el cliente y el servidor: éste está ubicado normalmente en otra máquina, y suele ser un gestor de base de datos, como DB2, SQL Server, Oracle, aunque también puede ser una base de datos más pequeña, como Paradox, dBase, etc., que accedemos directamente desde nuestra aplicación cliente.

Los mejores gestores de base de datos relacionales proporcionan soporte para implementar en ellos bastantes reglas de negocio, mediante el uso de

claves primarias, integridad referencial, triggers, etc., mientras que sistemas como dBase y otros apenas proporcionan soporte para reglas de negocio.

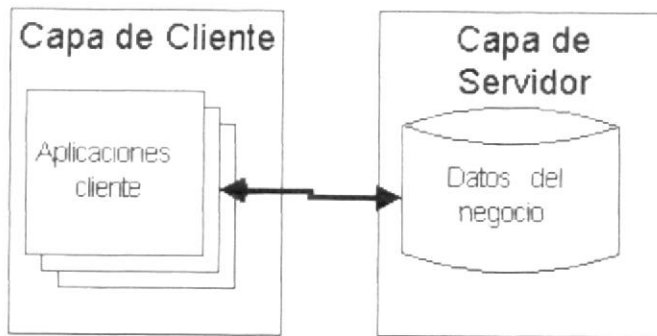


Figura A: Esquema de arquitectura Cliente/Servidor clásica.

Suponiendo que tengamos la información en un gestor de bases de datos potente, podremos despreocuparnos de llevar a cabo la codificación de numerosas validaciones en nuestras aplicaciones: así, si en la base de datos creamos una regla de integridad referencial que indica que todo pedido pertenece a un cliente, el gestor de base de datos rechazará cualquier intento de almacenar un pedido en el que se nos haya olvidado indicar el mismo. Cualquier aplicación que acceda a esta base de datos se beneficiará de esta y otras validaciones automáticamente, sin tener que añadir ni una línea de código.

Si estamos utilizando una base de datos menos potente, como dBase, no estamos de suerte. Casi todas las reglas de negocio deberán implementarse dentro de los programas que accedan a la base de datos. Si los programas que acceden a la base de datos son varios, garantizar que en todos ellos se

respetan todas las reglas puede llegar a ser muy difícil y engorroso, especialmente si se desarrollan con distintas herramientas.

La necesidad de implementar reglas de negocio dentro de las aplicaciones cliente puede surgir también utilizando gestores de bases de datos más potentes. En primer lugar, las bases de datos relacionales son cada vez más potentes, pero no todas las reglas de negocio pueden reflejarse en ellas: por ejemplo, las reglas de flujo son bastante difíciles de implementar dentro de la base de datos, y suelen ser las aplicaciones cliente las que controlan que la información sigue una ruta válida a través del sistema. Sin embargo, muchas de las reglas de negocio pueden reflejarse adecuadamente a nivel de la base de datos con estos gestores.

El problema se agrava cuando la información del negocio se encuentra en distintas bases de datos, gestionadas por distintos gestores, digamos DB2 y Oracle: si, por ejemplo, en DB2 almacenamos la información sobre facturas, etc., y en la base de datos Oracle almacenamos información técnica, como reparaciones llevadas a cabo en la maquinaria, ¿cómo reflejamos que ciertas facturas corresponden a una reparación, y garantizamos que no podamos relacionar una reparación con una factura inexistente? Evidentemente, aquí no hay manera de establecer una regla de integridad referencial entre tablas almacenadas en dos bases de datos distintas y correspondientes a distintos gestores de base de datos. De nuevo, la solución al problema es implementar el chequeo en cada aplicación cliente,

comprobando que exista la factura en la tabla DB2 antes de referenciar la en la tabla de reparaciones Oracle.

Ya que parece que de cualquier modo seremos nosotros mismos los encargados de obligar a que se cumplan algunas reglas de negocio, puede ser conveniente encontrar la manera de centralizar la gestión de estas reglas en un único lugar, de modo que todo el código necesario no se haya de duplicar en cada una de las aplicaciones. La solución puede ser crear una aplicación que se encargue de llevar a cabo estas tareas, de modo que todos los clientes pidan o envíen información a la misma, no al gestor de base de datos en el servidor: a éste solo accederá la nueva aplicación, que conforma una nueva capa dentro de un sistema Cliente/Servidor, la capa intermedia o *middle-tier* (**Figura B**), con lo que nuestro sistema ha pasado de ser un sistema Cliente/Servidor convencional a ser un sistema con tres capas (*three-tiered*). Conviene apuntar que pueden haber varias de estas aplicaciones, que llamaremos *servidores de aplicación*, lo que permite distribuir la carga de trabajo.

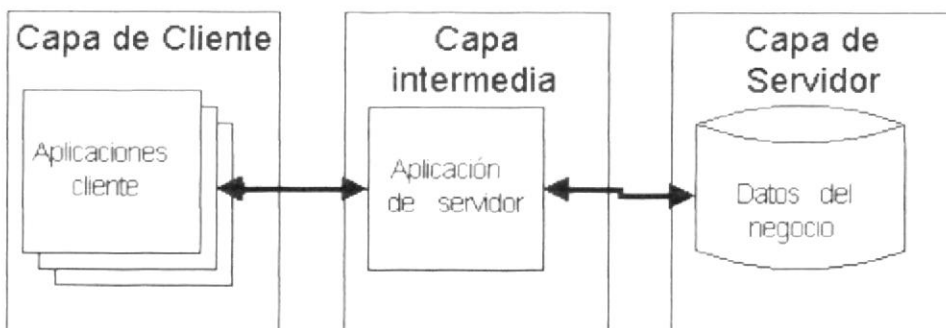


Figura B: Arquitectura Cliente/Servidor en tres capas (three-tier)

Ubicación de las reglas de negocio

La decisión de dónde ubicar una determinada regla de negocio dentro de una arquitectura C/S de tres capas puede simplificarse mucho si se atiende al tipo de regla de que se trata, utilizando la clasificación introducida más arriba.

Las *reglas del modelo de datos* especifican los valores válidos de cada atributo de las diversas entidades que se almacenan, lo que simplificando es lo mismo que decir los valores válidos para cada campo de cada tabla. Estas reglas deben, a ser posible, reforzarse en el servidor: esto se hace en primer lugar escogiendo correctamente el tipo de los campos de cada tabla (no almacenar una fecha en un campo de tipo cadena si nuestra base de datos dispone de campos del tipo fecha), y donde el servidor lo soporte, mediante restricciones (por ejemplo, en Interbase es posible especificar mediante CHECK diversas condiciones que debe verificar un dato para poder almacenarse en un campo), etc. El hacer esto así proporcionará mayor robustez a la base de datos.

La competencia en el mercado de cliente/servidor.

Cliente/servidor, el gran igualador del ramo de la computación, alienta la apertura y representa un campo de juego nivelado en el que puede participar una amplia variedad de plataformas de cliente y servidor. El entorno de cliente/servidor abierto actúa como catalizador para la "conversión en mercancías" del hardware y el software del sistema. La PC es un buen ejemplo de mercancía de cómputo: puede obtenerse con múltiples proveedores y se le vende situaciones de mercado y de precios sumamente competitivos.

Características que poseen todos los sistemas de Cliente/Servidor

Servicio: Cliente/Servidor es fundamentalmente una relación entre procesos ejecutados en aparatos distintos. El proceso del servidor hace de este un proveedor de servicios. El cliente es un consumidor de servicios.

Recursos compartidos: Un servidor puede atender a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.

Protocolos asimétricos: Entre cliente y servidor se establece una relación de "muchos a uno". Son siempre los clientes los que inician el diálogo al

solicitar un servicio. Los servidores aguardan pasivamente las solicitudes de los clientes.

Transparencia de ubicación: El servidor es un proceso que puede residir en el mismo aparato que el cliente o en un aparato distinto a lo largo de una red.

Mezcla e igualdad: El software ideal de cliente/servidor es independiente del hardware o de las plataformas de software del sistema operativo.

Intercambios basados en mensajes: Clientes y servidores son sistemas holgadamente acoplados que interactúan a través de un mecanismo de transmisión de mensajes. El mensaje es el mecanismo de entrega para las solicitudes y respuestas de servicio.

Encapsulamiento de servicios: EL servidor es un "especialista". Un mensaje le indica a un servidor qué servicio se solicita: éste se le envía luego al servidor para determinar el cumplimiento de la tarea. Los servidores pueden ser estudiados sin afectar a los clientes, siempre y cuando la interfaz para la publicación del mensaje no cambie.

Facilidad de escalabilidad: Los sistemas de cliente/servidor pueden escalarse horizontal o verticalmente. La escalabilidad horizontal significa la adición o eliminación de estaciones de trabajo del cliente con apenas un

ligero impacto en el desempeño. La escalabilidad vertical significa migrar a un aparato servidor más grande y más veloz o a servidores múltiples.

Integridad: El código del servidor y los datos del servidor se conservan centralmente, lo que resulta en un mantenimiento de menor costo y en la protección de la integridad de los datos compartidos. Al mismo tiempo, los clientes mantienen su individualidad e independencia.

Existe un verdadero Cliente Servidor?

A muchos sistemas con diferentes arquitecturas se les ha llamado "Cliente/servidor". Los proveedores de sistemas suelen emplear el término cliente/servidor como si se aplicara únicamente a sus paquetes específicos.

La idea de dividir una aplicación a lo largo de las líneas de cliente/servidor se ha utilizado en los últimos diez años para crear diversas modalidades de soluciones de software para red de área local (LAN). Por lo general estas soluciones se venden como paquetes comerciales de software, y en muchos casos las distribuyen varios proveedores. Sin embargo, cada una de estas soluciones se distingue por la naturaleza del servicio que ofrece a sus clientes, como se mostrará en las siguientes secciones.

Servidores de Archivos, el cliente (usualmente una PC) envía solicitudes de registros de archivos al servidor de archivos a través de una red.

Servidores de Base de datos, el cliente envía solicitudes de SQL en calidad de mensaje al servidor de bases de datos. Los resultados de cada orden de SQL son devueltos por medio de la red. EL código que procesa la solicitud de SQL y los datos residen en la misma máquina.

Servidores de Transacciones, el cliente invoca procedimientos remotos que residen en el servidor con un mecanismo de bases de datos de SQL. Estos procedimientos remotos en el servidor ejecutan un grupo de instrucciones de SQL. El intercambio por la red consiste en un solo mensaje de solicitud/respuesta (a diferencia de lo que ocurre con el servidor de bases de datos, en cuyo caso es necesario un mensaje de solicitud/respuesta para cada instrucción de SQL en una transacción).

Servidor de groupware, dirige a la administración de información semiestructurada como texto, imagen, correo, tableros de avisos y flujo de trabajo. Estos sistemas de cliente/servidor establecen un contacto directo entre personas.

Servidores de Objeto, la aplicación cliente/servidor se genera como un conjunto de objetos de comunicación. Los objetos de cliente se comunican con los objetos del servidor mediante un corredor de solicitud de objetos (ORB: object request broker).

Servidores Web, el World Wide Web es la primera aplicación cliente/servidor verdaderamente intergaláctica. Este nuevo modelo de cliente/servidor está integrado por clientes "universales" compactos, portátiles, en comunicación con servidores superamplios. En su modalidad más simple, un servidor web envía documentos cuando los clientes los solicitan por su nombre.

Conclusiones

No hay una única posibilidad a la hora de distribuir las reglas de negocio dentro de un esquema Cliente/Servidor. Sin embargo, sí hay ciertas pautas que se pueden tener en cuenta a la hora de tomar una decisión, basadas en una clasificación de las reglas de negocio aquí expuestas: en general, lo más recomendable suele ser implementar todas las reglas de negocio relativas al modelo de datos y las relaciones en el servidor, dado que los modernos servidores suelen llevar a cabo estas tareas muy eficazmente.

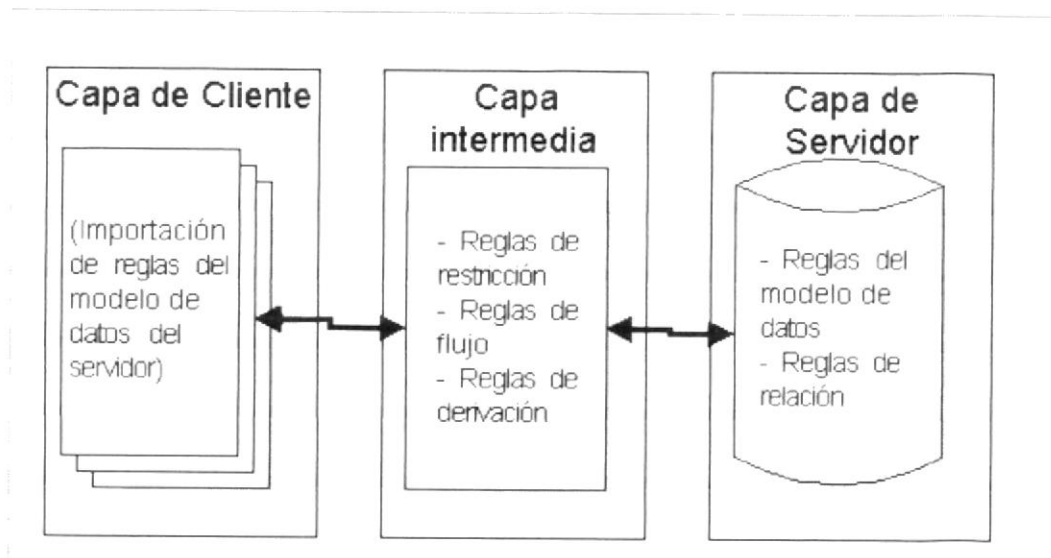


Figura C: Ubicación de las reglas de negocio dentro del esquema de tres capas.

Por lo que respecta al resto de las reglas, la mejor solución suele ser implementarlas en la capa intermedia: si tuviésemos que acceder a varias bases de datos, habríamos de migrar aquí algunas de las reglas que de otro modo irían al servidor, especialmente las de relación. Si bien el tráfico en la red se incrementará al utilizar una capa intermedia, este puede quedar

aliviado haciendo que ésta resida en la misma máquina que el servidor de datos, o al menos dentro de la misma red local. La **Figura C** muestra gráficamente la ubicación recomendable de las reglas de negocio dependiendo de su tipo (para acceso a una única base de datos).

Por último, vale la pena resaltar la conveniencia de implementar las reglas del modelo de datos también en el cliente, para hacer fluida la interacción con el usuario, siendo lo ideal importarlas dinámicamente del servidor de base de datos. Por lo demás, la implementación de reglas en las aplicaciones cliente puede dar lugar a muchos problemas, tanto de velocidad como de portabilidad y fiabilidad, al tener que reflejar una y otra vez las mismas reglas en distintas aplicaciones, quizá desarrolladas con distintos lenguajes y ejecutándose bajo distintos sistemas operativos.

Apéndice B

PROGRAMACIÓN EN EL WEB.

Html.

HTML es un lenguaje sencillo pensado para presentar información en la WWW. HTML (HyperText Markup Lenguaje), como su nombre indica es un lenguaje de marcas para la creación de hipertextos. Por hipertexto entenderemos texto con una presentación agradable, con inclusión de elementos multimedia (gráficos, video, audio) y con la presencia de hiperenlaces que permiten relacionar otras fuentes de información en documentos hipertextos. Es el lenguaje utilizado para representar documentos en la WWW (World Wide Web).

Como se ha dicho es un lenguaje de marcas ya que en el las instrucciones son trozos de texto resaltados convenientemente que definirán la estructura lógica del documento. Por tanto un documento HTML constará de texto que será el contenido y la información del documento y de instrucciones HTML que resaltarán este contenido y le darán un formato fácil y agradable de leer y con la posibilidad de relacionar documentos y fuentes de información mediante hiperenlaces.

Las marcas del lenguaje HTML especifican:

✓ La estructura lógica del documento:

- Cabeceras, tipos y tamaños de las fuentes.
- Párrafos de texto.
- Centrado.
- Enumeraciones o listas.
- Formularios.
- Tablas.

✓ Distintos estilos que definen el texto:

- Negrita.
- Cursiva.
- Diferentes efectos: (direcciones de correo, citas textuales, etc...).

✓ Inclusión de hipertextos para acceder a otros documentos relacionados

✓ Inclusión de imágenes y ficheros multimedia.

El lenguaje HTML actualmente está en su versión 2.0, aunque se encuentra en desarrollo la 3.0. En este estándar se definen las etiquetas básicas y las estructuras de datos que forman el lenguaje.

Dhtml.

El HTML Dinámico (DHTML) no es más que, en pocas palabras, una forma que tienen las páginas de aportar interactividad a las mismas. DHTML es una característica de Netscape Communicator 4.0, y Microsoft Explorer 4.0 y posteriores versiones de ambos navegadores, y está orientada al usuario. Es tarea del navegador mostrar y manipular las páginas web.

El DHTML tiene la ventaja de que es una herramienta con la que se pueden crear efectos que requieren poco ancho de banda, a la hora de bajarlos de Internet y, son estos efectos los que aumentan la funcionalidad de la página. Se puede utilizar para crear animaciones, juegos, aplicaciones, para introducir nuevas formas de navegar a través de los sitios web, y para crear un auténtico entramado de capas que con sólo el HTML sería imposible abordar. Aunque muchas de las características del DHTML se podrían duplicar con otras herramientas como Java o Flash, el DHTML ofrece la ventaja de que no requiere ningún tipo de plug-in para poder utilizarlo.

Aunque las tecnologías en las que se basa el DHTML (HTML, CSS, JavaScript) están estandarizadas, la forma en que Netscape y Microsoft las implementan difieren entre sí. Por este motivo, la creación de páginas web que usen esta tecnología, puede llegar a convertirse en una tarea muy compleja, puesto que hay que conseguir que la página se visualice perfectamente en ambos navegadores.

Css.

Los CSS son una adición al HTML que aporta a los desarrolladores una manera más sofisticada de estructurar páginas web. Esto lo consigue separando el contenido de la página web (el texto), de la visualización (colores, estilos, posicionamiento, etc).

Los Cascading Style Sheets Positioning (CSSP) son una extensión de los CSS, que permiten un control a nivel de pixels de las posiciones de los elementos HTML.

Java Script.

El JavaScript apareció, en primer lugar, para Netscape 2.0, y añadía funcionalidad a los formularios del HTML, a los frames y a las ventanas. Netscape 3.0 aportó nuevas características como el cambio de imágenes y controles audio/video. El Explorer 3.0, que salió poco después que el Netscape 3.0, también implementaba JavaScript, pero comercializado como JScript, que era esencialmente lo mismo que el JavaScript, aunque con algunas diferencias e incompatibilidades con las que Microsoft intentó hacer que los desarrolladores utilicen su versión de JavaScript.

Tanto en el Netscape 4.0 como en el Explorer 4.0, se introdujeron nuevas extensiones que ofrecían a los creadores de páginas web una forma de trabajar con el DHTML. De cualquier manera, estas extensiones no se estandarizaron antes de la aparición de estas versiones de los dos navegadores. Por este motivo, ahora tenemos dos versiones tan incompatibles de JavaScript.

Capas de Componentes Software.

Los componentes software de una arquitectura multi-capa constan de tres capas:

- ◆ La capa del **cliente** contiene los programas ejecutados por los usuarios, incluyendo navegadores Web y programas de aplicaciones de red. Estos programas se pueden escribir virtualmente en cualquier lenguaje de programación.
- ◆ La capa **media** contiene el servidor WebLogic y otros servidores que son direccionados directamente por los clientes, como servidores web existentes o servidores proxy.
- ◆ La capa **backend** contiene recursos de empresa, como sistemas de base de datos, aplicaciones de unidad central y legales, y aplicaciones de plannings de recursos de empresa empaquetados (ERP).

Las aplicaciones del cliente tienen acceso al servidor WebLogic directamente, o a través de un servidor web o un proxy. El servidor WebLogic conecta con servicios **backend** por cuenta de los clientes, pero los clientes no tienen acceso directamente a los servicios backend.

Componentes de la Capa Cliente.

Los clientes del servidor WebLogic utilizan interfaces estándares para acceder a servicios del servidor WebLogic. El servidor WebLogic tiene una completa funcionalidad de servidor web, así que un navegador web puede solicitar páginas al servidor WebLogic usando el protocolo estándar de la Web, HTTP. Los servlets de WebLogic Server y las JavaServer Pages (JSPs) producen páginas Web dinámicas, personalizadas requeridas para las aplicaciones avanzadas de comercio electrónico. Los programas del cliente escritos en Java pueden incluir interfaces gráficos de usuario altamente interactivos construidos con las clases de Java Swing. También se puede tener acceso a servicios del servidor WebLogic usando los APIs estándar del J2EE.

Todos estos servicios también están disponibles para los clientes de navegadores web desplegando servlets y páginas JSP en el servidor WebLogic. Los programas del cliente compatibles con CORBA escritos en Visual Basic, C++, Java, y otros lenguajes de programación pueden ejecutar JavaBeans Enterprise y RMI en el servidor WebLogic usando WebLogic

RMI-IIOP. Las aplicaciones del cliente escritas en cualquier lenguaje que soporten el protocolo HTTP pueden acceder a cualquier servicio del WebLogic Server a través de un servlet.

Componentes de la Capa Media.

La capa media incluye el servidor WebLogic y otros servidores Web, cortafuegos, y servidores proxy que median en el tráfico entre los clientes y el servidor WebLogic. El servidor WAP de Nokia, parte de la solución de comercio móvil de BEA, es un ejemplo de otro servidor de la capa media que proporciona una conectividad entre los dispositivos inalámbricos y el servidor WebLogic. Las aplicaciones basadas en una arquitectura multi-capas requieren confiabilidad, escalabilidad, y un alto rendimiento en la capa media. El servidor de aplicaciones que seleccionemos para la capa media es, por lo tanto, crítico para el éxito de nuestro sistema.

La opción **Cluster** del servidor WebLogic permite que distribuyamos peticiones de cliente y servicios **backend** entre varios servidores WebLogic cooperantes. Los programas en la capa del cliente acceden al cluster como si fuera un solo servidor WebLogic. Cuando la carga de trabajo aumenta, podemos agregar otros servidores WebLogic al cluster para compartir el trabajo. El cluster utiliza un algoritmo de balance de carga seleccionable para elegir el servidor WebLogic del cluster que es capaz de manejar la petición.

Cuando una petición falla, otro servidor WebLogic que proporciona el servicio solicitado puede asumir el control. Los fallos son transparentes siempre que sea posible, lo que reduce al mínimo la cantidad de código que se debe escribir para recuperar incidentes. Por ejemplo, el estado de la sesión de un servlet se puede replicar en un servidor secundario WebLogic de modo que si el servidor WebLogic que está manejando una petición falla, la sesión del cliente se pueda reanudar de forma ininterrumpida desde el servidor secundario. Todos los servicios de WebLogic, EJB, JMS, JDBC, y RMI están implementados con capacidades de clustering.

Componentes de la Capa Backend.

La capa **backend** contiene los servicios que son accesibles a los clientes sólo a través del servidor WebLogic. Las aplicaciones en la capa backend tienden a ser los recursos más valiosos y de misiones críticas para empresa. El servidor WebLogic los protege de accesos directos de usuarios finales. Con tecnologías tales como almacenes de conexiones y caches, el servidor WebLogic utiliza eficientemente los recursos **backend** y mejora la respuesta de la aplicación.

Los servicios **backend** incluyen bases de datos, sistemas de hojas de operación (planning) de recursos de la empresa (ERP), aplicaciones mainframe, aplicaciones legales de la empresa, y monitores de transacciones. Las aplicaciones existentes de la empresa se pueden integrar

en la capa **backend** usando la especificación de configuración del conector Java (JCA) de Sun Microsystems. El servidor WebLogic hace fácil agregar un interface Web a una aplicación **backend** integrada. Un sistema de control de base de datos es el servicio **backend** más común, requerido por casi todas las aplicaciones del servidor WebLogic. WebLogic EJB y WebLogic JMS normalmente almacena datos persistentes en una base de datos en la capa **backend**.

Un almacén de conexiones JDBC, definido en el servidor WebLogic, abre un número predefinido de conexiones a la base de datos. Una vez que estén abiertas, las conexiones a la base de datos son compartidas por todas las aplicaciones del servidor WebLogic que necesiten acceder a esa base de datos. Sólo se incurre una sola vez en la costosa sobrecarga asociada con el establecimiento de conexiones para cada conexión del almacén, por cada petición de cliente. El servidor WebLogic vigila las conexiones a la base de datos, refrescándolas cuando es necesario y asegurándose de la fiabilidad de los servicios de la base de datos para las aplicaciones.

Apéndice C

TECNOLOGÍAS COM / DCOM.

Qué es COM.

COM es una estructura para crear y utilizar componentes que:

1. Hace más fácil la escritura y la reutilización de software.
2. Proporciona la más amplia selección de servicios, herramientas, lenguajes y aplicaciones.
3. Proporciona el único mercado actualmente viable para otras compañías de componentes.
4. Tiene controles, herramientas y componentes de servidor.

Características COM.

1. Servicios utilizados de una manera estándar, independientemente de la localización.
2. Cualquier lenguaje de programación.
3. Integrado con las herramientas de desarrollo.
4. Seguridad flexible.
5. Múltiples transportes de red.

6. Especificación e implementación de referencia maduras.

Más sobre COM.

La estrategia de Microsoft Windows DNA se basa en la visión de Microsoft de los componentes cooperativos, construidos sobre el estándar binario llamado Component Object Model (COM). COM es el modelo de software basado en componentes más utilizado en el mundo, disponible hoy en día en más de 150 millones de equipos y servidores. Proporciona el mayor conjunto de servicios integrados, la más amplia gama de herramientas de uso sencillo, y el mayor conjunto de aplicaciones disponible. Además, suministra el único modo viable en el mercado de reutilizar componentes, listos para usar, de cliente y de servidor.

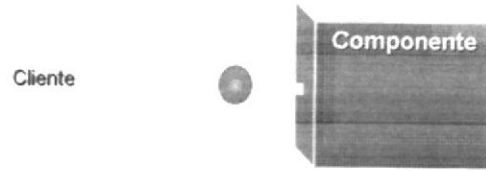
COM permite a los desarrolladores de software construir aplicaciones a partir de cualquier componente binario, que pueden ser implantadas en cualquier capa del modelo de aplicaciones. Estos componentes proporcionan los fundamentos de empaquetado, división y funcionalidad de la aplicación distribuida. COM permite que las aplicaciones sean desarrolladas usando componentes, encapsulando cualquier tipo de código o funcionalidad de la aplicación, como un control de interfaz de usuario o un objeto de negocio. Un componente puede tener uno o más interfaces; cada uno expone un conjunto de métodos y propiedades que pueden ser consultadas y modificadas desde otros componentes y aplicaciones. Por ejemplo, un

componente de clientes puede exponer propiedades tales como nombre, dirección y número de teléfono.

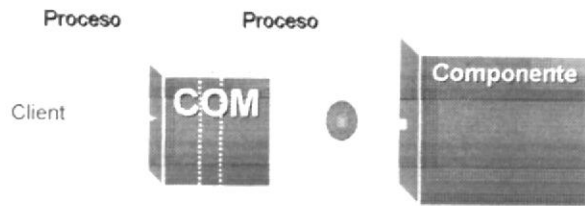
Con el modelo Microsoft Windows DNA, los componentes simplifican la complejidad de la construcción de aplicaciones multi-capas. Las aplicaciones basadas en componentes y en el modelo Windows DNA se basan en un conjunto común de servicios de infraestructura y de red proporcionado por la plataforma Windows. Los servicios de seguridad de Windows NT, por ejemplo, suministran el control de acceso a Internet Information Server, así como a servicios de control de transacciones y colas de mensajes. Otros servicios comunes incluyen gestión de sistemas, servicios de directorio, trabajo en red y soporte de hardware.

Acceso de COM.

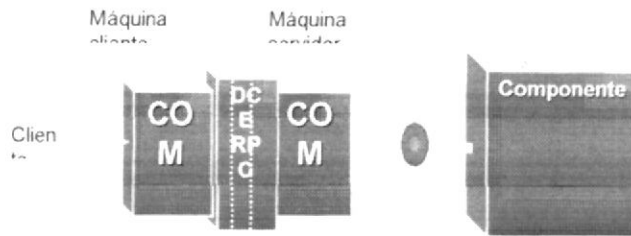
En el mismo proceso ,
llamadas directas a
funciones.



En la misma máquina.



Entre máquinas, DCE-
RPC seguras,
confiables y flexibles,
basadas en el protocolo
DCOM.



Principales Servicios COM.

1. Seguridad.	2. Administración del ciclo de vida.
3. Tipo de información.	4. Acceso a bases de datos.
5. Transferencia de datos.	6. Componentes.
7. Transacciones.	8. Comunicaciones asíncronas.
9. Registro.	10. Automatización (invocación dinámica).

El papel de COM.

Los servidores externos COM pueden intercambiar información aunque se ejecutan en un espacio de dirección diferente al del programa cliente. El proceso COM copia la información y transfiere esta copia al espacio de dirección del servidor para que los métodos del servidor puedan acceder a él. COM sustituye el indicador hacia la información original con un indicador hacia la copia en el espacio de dirección del servidor. Con IIS, es posible indicar que IIS actúe como cliente y las aplicaciones Web externas que se ejecutan en un paquete MTS separado como servidor.

Cuando finaliza el método del servidor, COM copia los datos del componente en el espacio de dirección del cliente. El proceso es transparente a las aplicaciones y puede incluso transformar tipos de datos distintos de forma automática. Ni el servidor ni el cliente se percatan de las acciones del proceso COM que se está realizando. Microsoft se refiere a este tipo de procesos que copian información de un proceso a otro como marshaling, lo que ocurre cuando una aplicación accede a una propiedad o método de un componente externo. Con IIS, el marshaling ocurre cuando se presenta una interacción con una aplicación Web. Cada vez que el navegador de un cliente solicita algo a una aplicación, por ejemplo.

DCOM

Introducción.

DCOM (Distributed Component Object Model), el Modelo de Objeto Componente Distribuido, es un juego de conceptos e interfaces de programa de Microsoft en el cual los objetos de programa del cliente pueden solicitar servicios de objetos de programa servidores en otros ordenadores dentro de una red. El modelo de Objeto Componente (COM) proporciona un juego de interfaces que permiten a los clientes y servidores comunicarse dentro del mismo ordenador (si están corriendo bajo un sistema Windows 95 o NT).

Por ejemplo, uno puede crear una página para un sitio web que contenga un script o programa que puede ser procesado (antes de ser enviado a otro usuario que lo solicite) no en el servidor del sitio web, sino en otro servidor más especializado de la red. Usando interfaces DCOM, el servidor del programa del sitio web (actuando ahora como un objeto cliente) puede iniciar una Llamada de Procedimiento Remoto (Remote Procedure Call, RPC) al objeto del servidor especializado, que proporciona el procesamiento necesario y devuelve el resultado al servidor del sitio web. Este pasa el resultado al usuario que está viendo la página Web.

Servicio de Microsoft Message Queue.

El servicio Microsoft Message Queue (MSMQ) constituye otro elemento importante de la plataforma. MSMQ es un servicio de software intermedio que facilita el sistema de mensajería entre varios procesos en una aplicación de varios niveles. Como se ha mencionado anteriormente, la mensajería es un elemento importante porque ofrece una comunicación asincrónica y sin conexión que no proporcionan RPC ni HTTP.

MSMQ se trata de un producto de mensajería basado en la entrega asincrónica de mensajes a colas con nombre. En un nivel superior, los mensajes crean un modelo de llamadas a procedimientos entre un cliente y un servidor a no ser que ambas partes puedan realizar su actividad sin necesidad la una de la otra. La principal diferencia conceptual reside en que el mensaje se desplaza en una sola dirección mientras que la llamada a un método COM implica tanto que se envíe la petición RPC al objeto, como que se devuelva una respuesta RPC al cliente.

Con MSMQ, la aplicación cliente puede enviar mensajes de petición incluso cuando la aplicación del servidor se encuentre fuera de línea. Esto también implica que el servidor podrá responder a mensajes de petición después que todas las aplicaciones cliente se hayan desconectado. En entornos en los que las aplicaciones cliente y los servidores se pueden desconectar por

determinadas razones, esta capacidad permite que toda la aplicación distribuida se mantenga en ejecución.

Apéndice D

MTS.

Introducción.

Microsoft Transaction Server (MTS) provee de un ambiente de tiempo de ejecución bajo Windows NT, el cual usa el mecanismo de interfaces COM para proveer un ambiente flexible de desarrollo de aplicaciones. MTS es satisfactorio para crear aplicaciones cliente/servidor de múltiples capas y en el Web. MTS está basado en la prueba de métodos para el procesamiento de transacciones, pero su significado trasciende al dominio de monitores para el procesamiento de transacciones. Esto define un modelo de programación simple y un ambiente de ejecución distribuido (aplicaciones del servidor basadas en componentes).

Las aplicaciones MTS están compuestas de componentes ActiveX que provee la función de la capa de negocios de la aplicación. Estos componentes son desarrollados como fueran para único usuario. Para su instalación éstos componentes se ejecutan dentro del ambiente de MTS, la aplicación servidora automáticamente escala para soportar la concurrencia de muchos clientes con alto desempeño y confiabilidad. MTS está específicamente diseñado para permitir a las aplicaciones del servidor escalar sobre un amplio rango de usuarios, desde sistemas de usuarios

únicos hasta volúmenes altos de los servidores de internet. MTS provee la robustez e integridad tradicionalmente asociada con los sistemas de procesamientos de grandes transacciones.

Por qué Transaction Server?

En la nueva versión de IIS, Microsoft ha apostado claramente por la integración del servidor *Web* con el servidor de transacciones *MTS*. El objetivo de Microsoft es ofrecer una plataforma completa y eficiente para la creación de aplicaciones *Web* que hagan uso de componentes *ActiveX* en el servidor. Cuando decimos componentes *ActiveX* hacemos referencia a componentes reutilizables que siguen el modelo de objetos COM y a los que nuestras aplicaciones *Web* van a solicitar servicios.

Infraestructura del Servidor.

Los servidores requieren una estructura sofisticada. Construir una red con servidores de aplicaciones es una tarea compleja. Implementar la función de negocios es una fracción pequeña del trabajo. En efecto, la mayoría del trabajo implica construir una infraestructura sofisticada que logre niveles aceptables de desempeño y escalabilidad.

Los desarrolladores de aplicaciones servidor usualmente deben elaborar gran parte de la infraestructura por si mismos. Por ejemplo, incluso con los servicios provistos por RPC(Remote Procedure Call) los desarrolladores de sistemas deben:

1. Desarrollar server-process para organizar la función de negocios.
2. Registrar los servers en el directorio del sistema.
3. Acceso sincronizado a datos compartidos y recursos a través de requerimientos concurrentes desde clientes. Esto requiere protocolos de seguridad sofisticados que controle los bloqueos, inanición y otros problemas de desempeño.
4. Administrar el contexto de clientes, incluye conecciones a la base de datos y estructuras de datos por usuarios (u objetos).

Qué es un componente?

El desarrollo de aplicaciones se está orientando a un escenario, Internet, en el que los programas creados deben estar preparados para poder ejecutarse en máquinas desconocidas, con requerimientos y sistemas operativos también desconocidos. En este ámbito, la industria del desarrollo se está volcando en una nueva idea: el *ComponentWare*. Este término se refiere a la estrategia de desarrollo que pretende, antes de construir programas

grandes de manera completa, dedicarse a la programación de pequeños (o no tanto) componentes que puedan colaborar entre sí, para obtener una funcionalidad mucho mayor que la que cada uno de ellos proporciona por separado.

El objetivo de esta tecnología es conseguir que los citados componentes puedan utilizarse sin preocuparse, ni conocer, el lenguaje o plataforma de desarrollo utilizada para construir cada uno de ellos. Es decir, podemos pensar en estos componentes como piezas prefabricadas que van a permitirnos, una vez ensambladas, construir el edificio de nuestra aplicación. Basta con que cada componente cumpla unas ciertas características desde el punto de vista externo y que facilite un cierto medio para que sus funcionalidades puedan ser utilizadas.

Estarás pensando que estos componentes milagrosos que estoy presentando poco a poco empiezan a parecerse peligrosamente a cualquiera de los elementos utilizados hasta el momento para la reutilización del código: sean clases, librerías estáticas o librerías dinámicas. El *componentware* propone una reutilización bastante más ambiciosa: cada unidad, es decir, cada componente, debe constituirse como un programa compilado, que posee la funcionalidad de generar y responder a eventos que le permitan interactuar con el usuario, con la máquina, o con otros componentes.

Transacciones

Transaction Server. Vamos con la primera parte del nombre. La definición más o menos clásica de transacción es "una serie de acciones que constituyen una operación atómica", esto es, que se realizan por completo o no se realizan en absoluto. Las transacciones resultan extremadamente importantes en entornos en los que existe riesgo de concurrencia entre diversos operadores que puedan estar intentando modificar el estado de ciertos datos simultáneamente, con el consiguiente peligro de inconsistencia. Sirva como ejemplo la situación en la que varias aplicaciones cliente intenten operar simultáneamente sobre un mismo conjunto de registros en un gestor de base de datos relacionales cliente/servidor.

Las transacciones deben cumplir un conjunto de propiedades que son conocidas como ACID, siglas inglesas de Atomicidad, Completitud, Aislamiento y Durabilidad. La Atomicidad hace referencia a que todas las operaciones de la transacción deban llevarse a cabo o descartarse simultáneamente mientras que el aislamiento entre transacciones está soportado por la necesidad de que las transacciones no puedan verse afectadas por las modificaciones aún no comprometidas de transacciones en curso y la durabilidad.

Para qué sirve MTS?

MTS nace con el objetivo de facilitar el desarrollo y gestión de componentes que llevan a cabo trabajos en el ámbito de transacciones. Pongámonos, en el lugar de un desarrollador que crea una aplicación que utiliza componentes COM para realizar tareas coordinadas. Supongamos que estas tareas deben realizarse todas concertadamente para conseguir que el resultado sea el esperado. Parece evidente que de la propia naturaleza de las citadas operaciones va a resultar poco menos que imprescindible definir transacciones que involucren a los citados componentes. ¿Quién coordina esas transacciones, cuando los elementos de software que realizan las tareas (los componentes) son módulos independientes que posiblemente desconozcan por completo la existencia de los otros? La respuesta es que es necesario un servidor que cuide de estos aspectos: Transaction Server.

Toda aplicación que utilice este modelo cliente servidor puede definirse como una entidad que tiene un cierto estado (por ejemplo los stocks y la facturación en un negocio) y que permite su modificación mediante una serie de operaciones definidas por la propia aplicación. Los componentes en un servidor suelen llevar a cabo todas estas operaciones que permiten que los citados datos sean coherentes, accesibles al tiempo que facilitan su actualización y consulta.

Estas reglas de coherencia, las reglas de negocio, requieren de una cierta lógica de aplicación que es el trabajo de los programadores diseñar. Este es el cuerpo del código de los componentes. El trabajo de MTS es descargar a los programadores de todos los aspectos tangenciales que no sean estrictamente la implementación de las reglas de negocio y, especialmente, de los posibles conflictos que unos componentes puedan provocar sobre los otros. Cuando un componente está controlado durante su ejecución por *MTS* todas sus operaciones son susceptibles de enmarcarse en transacciones. *MTS* se ocupa de resolver todos los problemas de concurrencia, en memoria, en lógica de programa y en gestión de recursos.

El modelo de operaciones en MTS.

Bien, hemos dicho que nuestro objetivo es crear una aplicación que se ubique en un servidor (o varios colaborando entre ellos) y que sea accedida por clientes que van a consultar, actualizar o gestionar los datos que conforman el estado de la citada aplicación. Las operaciones sobre el estado de la aplicación, la lógica de negocio, van a implementarse con componentes COM cuyos aspectos transaccionales van a ser controlados y gestionados por Microsoft Transaction Server.

Los componentes MTS.

Cada uno de los componentes de la aplicación, que en principio es un componente COM cualquiera, se convierte en un componente MTS. Un componente MTS es un componente COM constituido como una DLL, y que se ejecuta en el entorno de Transaction Server. Para ello los componentes deben cumplir un conjunto de características avanzadas que no vamos a exponer aquí.

Del mismo modo que una instancia de un componente COM es un objeto COM, toda instancia de un componente MTS es un objeto MTS. Cuando creamos un ejemplar de un componente MTS el servidor crea automáticamente un objeto asociado de contexto (*Context Object*) que contiene información sobre quién originó la creación del objeto y cómo se está ejecutando, principalmente desde el punto de vista de las transacciones en las que el objeto está inmerso.

Los dispensadores y gestores de recursos.

Los recursos de los que hablamos son básicamente los datos, que se ubican en uno o varios servidores, pongamos como ejemplo, y para centrar ideas, *SQL Server*. Cada uno de estos datos, que conforman el estado de nuestra aplicación, es controlado por un gestor de recursos (*Resource Manager*).

SQL Server es un ejemplo de gestor de recursos que puede atender a transacciones distribuidas.

Los gestores de recursos son consultados y tomados bajo su ámbito de acción por *MTS* para asegurar que las transacciones que operan sobre los citados datos son llevadas a cabo correctamente.

Los componentes *MTS* no acceden directamente a los gestores de recursos, sino que lo hacen a través de un módulo intermedio (el dispensador de recursos *Resource Dispenser*) que permite la independencia de la llamada a los gestores de recursos, con lo que diferentes componentes podrán llamar a diferentes gestores de recursos sin variar significativamente la tarea de programación. Este enredo quedará claro inmediatamente si ponemos un ejemplo de cada uno de estos elementos. Si SQL Server es un gestor de recursos, *ODBC* es un dispensador para él.

El explorador de Transaction Server.

Microsoft Transaction Server nos proporciona una herramienta, el explorador de Transaction Server, que nos permite ver qué es lo que está sucediendo con un conjunto de componentes *MTS* que están ejecutándose en el ámbito de *Microsoft Transaction Server*.

Los componentes en el explorador se agrupan en paquetes (*packages*). Un paquete es un conjunto de componentes COM que se agrupan para su gestión conjunta con *Transaction Server*. Normalmente se agrupan aquellos componentes que llevan a cabo tareas relacionadas en la aplicación. Los componentes de un paquete se ejecutan en el ámbito de un mismo proceso en el servidor.

El explorador presenta los componentes precisamente estructurados en paquetes según una cierta estructura jerárquica. Los paquetes están asociados a una máquina y contienen un conjunto de componentes. Cada uno de estos componentes tiene asociado interfaces y métodos. Cada una de estas entidades corresponde a un nodo del árbol jerárquico del explorador.

Creación e instalación de paquetes.

Pueden crearse paquetes con componentes ya existentes no instalados en ningún paquete (es decir, DLL que pueden o no estar registradas en la máquina), o incorporar al paquete componentes incluidos en otros. Cuando incorporamos un componente COM no registrado a un paquete, el proceso se encarga de registrar la DLL en el registro del sistema. Cada componente puede estar en múltiples paquetes.

El explorador nos proporciona un asistente para la creación de paquetes vacíos. En este modo de creación se nos solicita simplemente el nombre que deseamos dar al paquete y la cuenta de Windows NT bajo la que los componentes en el paquete se ejecutarán.

Una vez creado el paquete podemos crear componentes nuevos e incluirlos en él. Crear nuevos componentes no significa aquí desarrollar los componentes desde el punto de vista binario, sino más bien "darlos de alta" en el paquete, con lo que les damos un lugar y un ámbito en el que ejecutarse en el marco del *run time environment* del servidor de transacciones. También contamos con un asistente para la creación de nuevos componentes, en el que se nos presenta la opción de crear un nuevo componente a partir de un fichero DLL aún no registrado u obtener una lista de los componentes registrados que cumplen la condición indispensable de ser *in-process*, para que podamos elegir cual es el que deseamos añadir al paquete.

Monitorizando y gestionando los componentes de un paquete.

Cuando disponemos de un conjunto de componentes agrupados en un paquete ya estamos preparados para monitorizar el comportamiento de los citados componentes. EL proceso de instalación de *Microsoft Transaction*

Server crea un conjunto de paquetes predefinidos que pueden servirnos de aprendizaje. Uno de ellos, "Sample Bank", simula una aplicación bancaria que hace uso de componentes controlados por *MTS* para llevar a cabo las operaciones de la lógica de negocio, tales como hacer ingresos, reintegros, etc.

La instalación también genera una aplicación cliente que va a permitirnos comprobar cómo las llamadas a los componentes van a desatar la ejecución de transacciones.

El explorador nos presenta dos vistas de las propiedades de cada componentes de cada paquete: la visión de estado y de propiedad. La primera de ellas (*State View*), presenta el estado (incluyendo si están recibiendo peticiones de servicio o no) de las máquinas, los paquetes o los componentes. La vista de propiedades (*Property View*) muestra las propiedades de cada uno de los elementos, incluyendo el identificador de clase de cada componente.

Otro aspecto monitorizable son las transacciones. El explorador dispone de dos nodos especialmente diseñados para presentar esta información. En la figura, mostramos las estadísticas de las transacciones a las que se ha visto sometido el componente *Bank* en una sesión de prueba.

Conclusión

En este artículo se pretende mostrar la necesidad y conveniencia de utilizar Microsoft Transaction Server presentando la naturaleza básica de este servidor. Transaction Server no es un elemento abordable en una pocas líneas.

Apéndice E

IIS.

Introducción.

Las empresas buscan tecnologías de Web basadas en estándares para, de una manera más eficiente, intercambiar información con clientes, socios y empleados de todo el mundo. Además, intentan implementar soluciones que aprovechen al máximo las inversiones existentes en conocimientos, formación y tecnología. Microsoft Internet Information Server (IIS) está diseñado para cubrir estas necesidades a un amplio rango de usuarios, desde grupos de trabajo y departamentos de una intranet corporativa hasta proveedores de servicios Internet que alojan sitios Web que reciben millones de visitas diarias. IIS 4.0 revoluciona las capacidades de Web del sistema operativo Microsoft Windows NT Server versión 4.0, proporcionando la forma más fácil de compartir información, crear y distribuir aplicaciones de negocio, y alojar y administrar sitios.

Ventajas

- ✓ Las características innovadoras de publicación de Web, herramientas personalizables y tecnologías de nuevos asistentes, hacen de Windows NT Server con IIS 4.0 la forma más fácil de publicar y

compartir información, de forma segura, a través de intranets corporativas y a través de Internet.

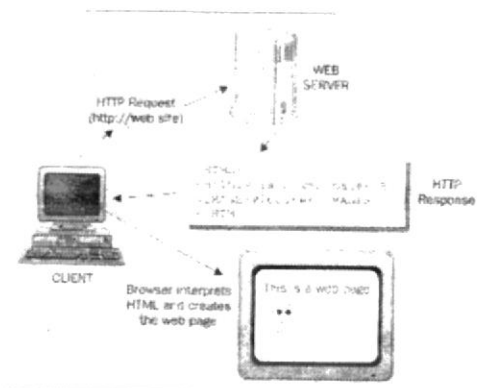
- ✓ Las aplicaciones Web pueden simplificar procesos como la creación de informes de gastos, administración de beneficios, compras, administración de inventario, marketing y ventas. IIS 4.0 proporciona todo lo necesario para distribuir aplicaciones Web fiables y escalables en Windows NT Server.
- ✓ Las herramientas personalizables de administración, opciones flexibles de administración y herramientas de análisis hacen de Windows NT Server con IIS 4.0 el servidor de Web más fácil de administrar.

Resultados al usuario den HTML. ASP permite crear formulario dinámicos que devuelven resultados a los usuarios, permite actualizar el contenido de un web site, ofrece acceso a bases de datos y devuelve los resultados en un web site, y permite acomodar a las necesidades del usuario.

Cuando una requisición por una página llega al web server, el web server interpreta la requisición, encuentra la página en el servidor , ejecuta los comandos escritos en la página y genera una página de resultados y la retorna al usuario.

Script Lenguajes

Para añadir funcionalidades dinámicas páginas en formato HTML se utilizan comandos que nacen de lenguajes de programación. El lenguaje de programación fue Visual Basic Scripts y JavaScripts y Java Applets.



Flujo de información con ASP

Microsoft recomienda el uso del ASP del lado del servidor más que un script del lado del cliente cuando sea posible elegir, ya que el script del lado del servidor dará como resultado una página HTML fácilmente desplegable. Los scripts del lado del cliente (por ejemplo, con JavaScript) pueden no funcionar de manera óptima en navegadores antiguos.

¿Porqué utilizamos ASP?

Porque ASP es una amigable y muy eficiente forma de hacer aplicaciones de comercio electrónico. Es muy poderosa y también muy fácil de usar. Si se requiere una aplicación "Ecommerce" sobre la web pero se necesita aprender rápidamente un lenguaje de procesamiento de datos, la tecnología

ASP lo es. Es también seguro que esta tecnología permanecerá durante mucho tiempo.

Active Server Pages(ASP) es un componente de Microsoft Internet Information Server(IIS). ASP fue liberado por Microsoft en 1996. ASP tiene muchas ventajas. La mayor es que reemplaza la forma tradicional de intercambiar información entre usuarios. Los métodos tradicionales de intercambio cliente-servidor involucraban la tecnología Common Gateway Interface (CGI). CGI requiere una aplicación separada desde un documento web para procesar todos los datos regresados al servidor. Esto tiene desventajas tanto en la seguridad como en el procesamiento de recursos.

Active Server Pages permiten que el código sea incrustado en un documento HTML y que corra en el servidor, es un concepto simple. El código que necesita correr en el servidor se coloca entre especiales script tags . Estas tags le dicen al servidor que no envíe el código que está entre ellas al cliente. El código no será necesario hasta que se reciba una solicitud del cliente, luego éste procesará el código en el servidor basado en la información enviada de regreso desde el cliente, el código nunca deja el servidor haciendo por lo tanto a ASP muy seguro y fácil de programar.

ASP se ejecuta sobre el servidor y envía datos de regreso al cliente (a través del browser). ASP es compatible con múltiples plataformas.

Existen varios lenguajes que se pueden usar para hacer ASP. El más comúnmente utilizado es VBScript y es por mucho el más fácil, ya que es

nativo de Microsoft. ASP se puede hacer también en Perl and Jscript (no JavaScript).

ASP ha logrado llegar a ser el líder en la construcción de aplicaciones ECommerce. Es más rápido, más eficiente, poderoso y extremadamente fácil de utilizar. Los métodos tradicionales de comercio electrónico conocidos como CGI aún existen y son usados ampliamente, sin embargo utilizan más recursos y son difíciles de implementar. También involucran el uso de sistemas operativos que son un desafío el aprenderlos.

Funcionamiento de ASP.

Cuando se trabaja con IIS y ASP el servidor Web analiza las peticiones de página que le llegan. Cuando la petición tiene extensión .ASP se apoya en la aplicación ISAPI que sirve de soporte de ejecución de paginas ASP. ISAPI diferencia entre líneas HTML y los scripts. Cuando determina el lenguaje en el que encuentran los scripts (o programas) da paso a los motores de ejecución de scripts adecuados, Visual Basic Script, Java Script... Estos últimos se encargan de hacer al análisis sintáctico y la compilación de las instrucciones ejecutables. Una vez realizado esto se procede a ejecutar las instrucciones. El usuario recibe como respuesta un fichero ".htm" que se ha formado uniendo las instrucciones HTML originales de la página ".asp" con las instrucciones HTML que se han generado tras los scripts.

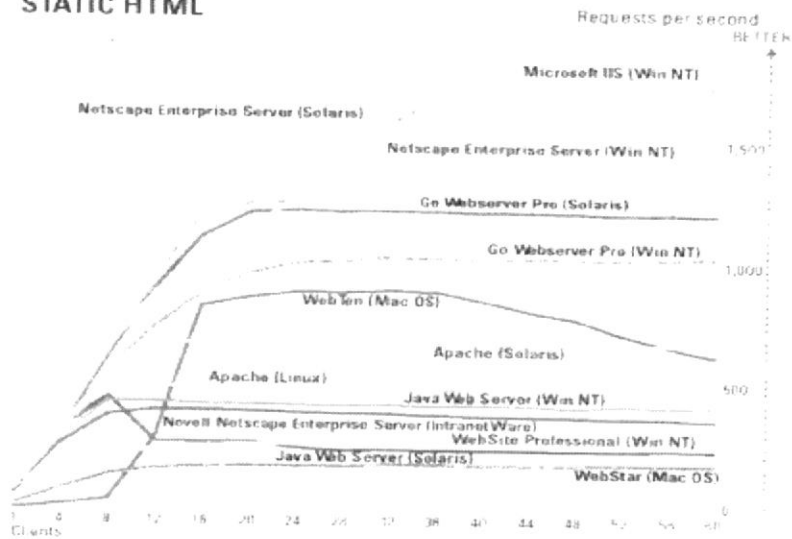
Más Sobre el IIS.

El servidor web incorporado en Microsoft® Windows NT® Server, Internet Information Server 4.0 (IIS), facilita compartir documentos e información en

una intranet o en Internet. IIS es el servidor web más rápido para Windows NT Server y está integrado completamente a sus Servicios de Directorio. Esta combinación, servicios web y sistema operativo, hace posible desplegar aplicaciones escalables y confiables basadas en el web. Esta nueva generación de soluciones de negocio en red incorpora las aplicaciones y los datos legados, al mismo tiempo permite a las compañías redefinir sus procesos internos y externos.

Windows NT Server 4.0 se integra en forma natural al web. En Windows NT Server 4.0, el web se convierte simplemente en otra parte del sistema operativo. Las Active Server Pages del IIS ofrecen un ambiente de aplicación avanzado, abierto y libre de compilación, en el cual se puede combinar HTML, scripts y componentes de servidor ActiveX reusables, para crear soluciones de negocios dinámicas y poderosas basadas en el web. En el sistema operativo están completamente integradas herramientas como Microsoft FrontPage y la búsqueda de texto dinámica y total. Crystal Reports, incluida en ISS, es una herramienta para reportes visuales que permite crear presentaciones de calidad e integrarlas en aplicaciones de bases de datos. Esta integración web permite que usted construya sitios web con calidad profesional, administre una red, incremente la flexibilidad del sistema y su escalabilidad, además de que mantenga una seguridad total y centralizada.

STATIC HTML



De acuerdo con PC Magazine, Microsoft Windows NT Server 4.0 con IIS 4.0 proporciona el nivel más elevado de desempeño en comparación con otros servidores web que hay en el mercado.

Internet Information Server 4.0 proporciona:

Servicios de aplicación confiables.

La integración de IIS, Microsoft Transaction Server y Microsoft Message Queuing en Windows NT 4.0 Option Pack hace que Windows NT Server sea la plataforma más completa para desarrollar y desplegar fácilmente aplicaciones confiables, escalables y de usuario múltiple, basadas en el web.

En específico, esto se logra a través de:

- **Crash Protection**, permite que los usuarios ejecuten con confianza varias aplicaciones en el servidor web. Si una de las aplicaciones se detiene, el servidor web y las otras aplicaciones continuarán en ejecución, mientras la aplicación con la falla se reinicia con una solicitud posterior.
- **Transactional Active Server Pages-ASP**, IIS 4.0 introduce ASP, lo que permite a las aplicaciones con scripts y componentes efectuar acciones múltiples, con todas o ninguna acción relacionada. Este aspecto es crítico en las aplicaciones para base de datos.
- **Depuración de script**, IIS 4.0 proporciona una solución integrada para depurar *scripts* y aplicaciones de ASP, por lo que facilita el desarrollo de aplicaciones de ASP.
- **Colas de mensaje integradas**, MSMQ proporciona una manera sencilla para que las aplicaciones envíen y reciban mensajes con seguridad en una red (incluso en el caso de que parte de la aplicación o de la red no estén disponibles).
- **Soporte para Java**, IIS 4.0 incluye un conjunto de clases de Java para construir componentes del lado del servidor y la Java 1.1 Virtual Machine de Microsoft. Al usar páginas ASP y Microsoft Transaction Server (MTS-Servidor de Transacción de Microsoft), usted puede construir fácilmente aplicaciones de servidor empleando Java, y ejecutarlas con eficacia en un entorno de servidor complicado.

Servicios web completos.

IIS 4.0 fue diseñado para soportar varios escenarios de servidor web, que abarcan desde los sitios web simples en una intranet corporativa, hasta grandes granjas web de ISP (Proveedores de Servicios de Internet).

- **Control total del contenido,** Ahora los administradores pueden configurar el servidor web dependiendo del servidor, sitio, directorio virtual, subdirectorio o bien, tomando como base el archivo, con lo que se ofrece una flexibilidad completa en la mayoría de las configuraciones de servidor.
- **Administración del contenido y análisis del sitio,** IIS 4.0 proporciona algunas características nuevas que facilitan la administración y el análisis del contenido en un servidor web. Entre estas características figuran mensajes de error personalizados y calificaciones de PIC. Esta versión también incluye Microsoft Site Server Express, que le permite aprender el uso de patrones en el sitio y visualizar la estructura de éste a través del análisis básico del contenido y de un archivo de bitácora.
- **Sitios web múltiples,** Al contar con soporte para varios sitios web en una sola dirección de IP, operadores de sitio web, reducción de ancho de banda por sitio web, y administración mejorada y personalizable, IIS 4.0 permite que las organizaciones hospeden fácilmente varios

sitios departamentales en una intranet, o que proporcionen servicios de hospedaje a varios sitios públicos en Internet.

- **Estándares de Internet**, IIS 4.0 incorpora los estándares industriales anteriores y más modernos, incluidos: HTTP 1.1 para desempeño mejorado en Internet, emisión y manejo de certificados digitales X.509 (que se pueden mapear hacia cuentas de usuarios de Windows NT Server), además del envío de mensajes hacia servidores estándares de correo SMTP y de noticias NNTP.
- **Soporte administrativo automatizado**, Al emplear los nuevos IIS Administration Objects (Objetos de Administración IIS), usted puede hacer *scripts* y ejecutarlos en la línea de comando, con lo que logra automatizar varias tareas administrativas. Esta versión incluye un conjunto de ejemplos de *scripts* para que usted los emplee de inmediato o los personalice y satisfaga las necesidades administrativas que tenga.

Servicios de red integrados.

IIS 4.0 proporciona los niveles más altos de integración con Windows NT Server. Al integrar el servidor web y el sistema operativo base, Windows NT Server e IIS 4.0 proporcionan a las organizaciones una plataforma única para administrar la red y los servicios web y de aplicación.

- **Configuración,** IIS es el servidor web más fácil de configurar. Al emplear el asistente para descargar el programa, usted puede elegir los componentes específicos que desee instalar. Asimismo, la nueva capacidad de instalación sin supervisión permite que usted instale IIS en varios servidores, sin tener que monitorear las instalaciones.
- **Administración,** IIS 4.0 ofrece las herramientas de administración más flexibles para manejar el servidor web. Entre ellas figuran una consola de administración integrada, basada en Windows; una administración enriquecida basada en el explorador; elaboración de *scripts* en la línea de comando; así como interfaces programables para construir herramientas de administración personalizables. Esta versión también ofrece una nueva funcionalidad para restaurar la configuración anterior, por lo que permite que usted devuelva al servidor las configuraciones guardadas previamente.
- **Seguridad,** IIS 4.0 incluye un servidor de certificación integrado que se acopla firmemente al modelo de seguridad de Windows NT Server, permitiendo que las organizaciones emitan y administren certificados digitales estándares X.509 para Internet. Esta versión también ofrece mejoras de seguridad para los bancos internacionales. Al emplear la tecnología Server Gated Crypto, los bancos internacionales pueden ofrecer el encriptamiento más fuerte (de 128 bits) en sus transacciones en línea.
- **Motores de búsqueda,** IIS 4.0 incluye capacidades de búsqueda integradas que permiten a los usuarios crear formas de búsqueda

personalizadas con Active Server Pages, ActiveX Data Objects y consultas de SQL para buscar información en el servidor web.

El trabajo conjunto de IIS y ASP.

Es necesario conocer cómo se integran las aplicaciones ASP con IIS 4.0. Una raíz virtual puede contener más de una aplicación. IIS 3.0 limita cada raíz virtual a una aplicación definida por el archivo global.asa. IIS 4.0 le ofrece más flexibilidad, pudiendo tener más de una aplicación, cada una de las cuales tendrá una carpeta con sus definiciones global.asa.

Un beneficio adicional de ejecutar una aplicación por separado es que puede descargar y reinstalar componentes que se ejecutan como parte de la aplicación Web sin tener que parar el servicio IIS. Sin embargo, sustituir componentes dentro de una aplicación ASP es problemático, ya que si la aplicación está ejecutando un proceso con IIS, deberá parar el servidor Web para sustituir el componente. Esta interrupción puede significar un gran inconveniente si necesita recopilar y sustituir un componente utilizado por la aplicación ASP. Puesto que la aplicación ASP está utilizando el componente, podrá borrarlo o sustituirlo mientras IIS esté ejecutando el componente.

Para parar IIS, deberá hacer algo más que parar el servidor Web con ISM (Internet Services Manager). Diríjase a la aplicación Servicios del Panel de control e interrumpa el servicio IIS Admin (que a su vez interrumpe el servicio Web Publishing). A continuación, sustituya el componente y reinicie el servicio. La consecuencia es que la aplicación Web se interrumpe cuando

el servicio está apagado. (Este método puede causar problemas si utiliza componentes en su servidor de producción.)

Si desea parar una aplicación aislada, abra ISM, seleccione con el botón derecho del ratón el nombre de la aplicación y seleccione Propiedades. Seleccione Descargar para parar la aplicación. A continuación, sustituya el componente. La aplicación se iniciará cuando los usuarios accedan a la aplicación.

Ejecutar aplicaciones Web dentro del espacio de procesos compartidos de IIS también produce beneficios. Experimentará una mejoría en el rendimiento del sistema, porque este método evita mucho del consumo de recursos de sistema asociados a aplicaciones externas.