

ESCUELA SUPERIOR POLITECNICA DEL LITORAL

Facultad de Ingeniería Eléctrica y Computación

“Sistema Renta de Vehículos”

TOPICO DE GRADUACION

Previo la obtención del Título de:

Licenciado en Sistemas de Información

Presentado por:

Aurora Godoy Choca
Xiomara Arcentales Torres
Julio Martínez Choca

GUAYAQUIL-ECUADOR

Año: 2002

AGRADECIMIENTO

A Dios y a mis queridos padres, por su paciencia, apoyo y cariño prodigado durante esta etapa de mi vida. A mis abuelitos, a mis hermanos, a mis amigos y a todas aquellas personas que de una u otra manera han colaborado para la culminación de mis estudios.

Xiomara Arcentales

A Dios y a todos aquellos que son un claro ejemplo de solidaridad y generosidad, pues con ellos se vuelve a recordar que no hay satisfacción más grande que el dar. A todos ustedes, gracias.

Aurora Godoy

A Dios, a mi madre y a mi hermana por su confianza y apoyo incondicional, brindados en todo momento; así mismo agradezco a todas las personas que me ayudaron a finalizar esta etapa de mi carrera.

Julio Martínez

DEDICATORIA

Dedico este trabajo a aquellas personas que con sus consejos y sabiduría me han guiado por el camino de la vida.

Xiomara Arcentales

Dedico este esfuerzo a todos aquellos que luchan por ser mejores, y a aquellos que siendo mejores luchan por ser humildes.

Aurora Godoy

Dedico este logro a todas las personas que trabajan por alcanzar sus sueños e ideales para vivir en un mundo mejor.

Julio Martínez

TRIBUNAL DE GRADUACION

**Ing. Carlos Martín B., MSIG
DIRECTOR DEL TOPICO**

**Ing. Néstor Arreaga, MSIG
MIEMBRO PRINCIPAL**

**Lic. Jorge Olaya, MBA
MIEMBRO PRINCIPAL**

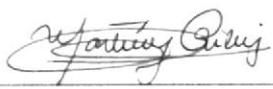
DECLARACIÓN EXPRESA

“ La responsabilidad del contenido de este trabajo de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL ”

Aurora Godoy C.



Xiomara Arcentales T.



Julio Martinez C.

INDICE GENERAL

RESUMEN	XII
INTRODUCCIÓN	1
CAPITULO 1	
1. ESPECIFICACIONES DEL PROYECTO.....	3
1.1 Justificación	4
1.2 Objetivos Generales	5
1.3 Especificaciones funcionales	6
1.4 Especificaciones no funcionales	12
1.5 Limitaciones y restricciones	14
1.6 Alcance.....	15
CAPITULO 2	
2. ANALISIS DE LA INDUSTRIA Y PRODUCTO	16
2.1 Definición de la Industria y Producto.....	17
2.2 Clientes	18
2.2.1 Comportamiento del consumidor ante el mercado convencional .	18
2.2.2 Reacción del consumidor ante el mercado virtual	19
2.2.3 Mercado convencional vs. Mercado virtual.....	20
2.3 Competidores	23
2.3.1 Funcionalidad de los competidores convencionales existentes ...	24
2.3.2 Desafíos encarados por competidores convencionales	25
2.4 Proveedores	26

2.5 Productos sustitutos	26
2.6 Análisis FODA del sistema	27
CAPITULO 3	
3. MODELO DE NEGOCIOS.....	29
3.1 Definición del Modelo de Negocios.....	30
3.2 Funcionalidad General.....	30
3.2.1 Diagramas de flujo del proceso	35
3.3 Sistema de Entrega	39
3.4 Forma de Pago.....	39
3.5 Seguridad	40
CAPITULO 4	
4. ARQUITECTURA DEL SISTEMA.....	41
4.1 Justificación de la selección del modelo	42
4.2 Tipos de Procesos Clientes	43
4.3 Tipos de Procesos Servidores	44
4.4 Tipos de Middleware.....	45
4.5 Tecnologías y herramientas escogidas para la implementación	48
4.6 Requerimientos de hardware y software para la puesta en producción	51
CAPITULO 5	
5. IMPLEMENTACION DE LA CAPA DE DATOS.....	53
5.1 Modelo Lógico de Datos	54

5.2 Modelo Físico de Datos	60
5.3 Diagrama Entidad Relación	67
5.4 Proceso Servidor de Base de Datos	67
CAPITULO 6	
6. IMPLEMENTACION DE LA CAPA DE NEGOCIOS	75
6.1 Estructura de los componentes de Acceso a datos.....	76
6.2 Objetos distribuidos de Acceso a datos	77
6.3 Comunicación con el proceso servidor de base de datos	84
6.4 Estructura de los componentes de Negocios	84
6.5 Objetos distribuidos de Negocios.....	86
6.6 Comunicación entre objetos distribuidos.....	94
6.7 Proceso Servidor transaccional	94
CAPITULO 7	
7. IMPLEMENTACION DE LA CAPA DE PRESENTACION	95
7.1 Diseño del Sitio Web	96
7.2 Cliente Standalone	99
7.3 HTML.....	100
7.4 Server Script – ASP (Active Server Pages).....	102
7.5 CSS	106
7.6 Comunicación con componentes de Negocios	107
7.6 Proceso Servidor WEB	108
CONCLUSIONES	109

RECOMENDACIONES	112
APENDICES	
Apéndice A Arquitectura Cliente / Servidor.....	114
Apéndice B. Programación en el Web.....	139
CSS.....	139
HTML.....	143
Apéndice C. COM/ DCOM/ COM+	147
COM.....	147
DCOM (Distributed Component Object Model)	154
COM+.....	169
Apéndice D	191
IIS.....	191
ASP	197

INDICE DE FIGURAS

	Pag.
Figura 3.1. Ingreso de Reservasiones	35
Figura 3.2. Confirmación de Reservasiones	36
Figura 3.3. Consulta de Reservasiones	37
Figura 3.4. Eliminación de Reservasiones	38
Figura 5.1 Diagrama Entidad Relación del Sistema Renta de Vehículos.....	67
Figura 7.1. Flujo de páginas dentro del Sistema Renta de Vehículos.....	98
Figura 7.2. Pantalla Principal de la aplicación standalone	99

INDICE DE TABLAS

	Pag.
Tabla 1. Comparación entre el Mercado Convencional y Mercado Virtual.....	21
Tabla 2. Diseño Lógico de la Entidad Agencia	54
Tabla 3. Diseño Lógico de la Entidad Parámetros	55
Tabla 4. Diseño Lógico de la Entidad Marca	55
Tabla 5. Diseño Lógico de la Entidad Tarjeta	56
Tabla 6. Diseño Lógico de la Entidad Cliente	56
Tabla 7. Diseño Lógico de la Entidad Vehículo	57
Tabla 8. Diseño Lógico de la Entidad Reservación	58
Tabla 9. Diseño Lógico de la Entidad Reservaciones Perdidas....	59
Tabla 10. Diseño Físico de la tabla de Agencias.....	60
Tabla 11. Diseño Físico de la tabla de Parámetros.....	61
Tabla 12. Diseño Físico de la tabla de Marcas	61
Tabla 13. Diseño Físico de la tabla de Clientes	62
Tabla 14. Diseño Físico de la tabla de Vehículos	63
Tabla 15. Diseño Físico de la tabla de Tarjetas	64
Tabla 16. Diseño Físico de la tabla de Reservaciones.....	65
Tabla 17. Diseño Físico de la tabla de Reservaciones Perdidas.....	66
Tabla 18. Componentes de Acceso a Datos.....	76
Tabla 19. Estructura del Objeto de Datos Cliente.....	77

Tabla 20.	Estructura del Objeto de Datos Vehículo	78
Tabla 21.	Estructura del Objeto de Datos Reservación	79
Tabla 22.	Estructura del Objeto de Datos Reservaciones Perdidas	80
Tabla 23.	Estructura del Objeto de Datos Parámetro	81
Tabla 24.	Estructura del Objeto de Datos Tarjeta	82
Tabla 25.	Estructura del Objeto de Datos Marca	83
Tabla 26.	Estructura del Objeto de Datos General	84
Tabla 27.	Componentes de Negocios	85
Tabla 28.	Estructura del Objeto de Negocios Tarjeta	86
Tabla 29.	Estructura del Objeto de Negocios Cliente	87
Tabla 30.	Estructura del Objeto de Negocios Reservación	89
Tabla 31.	Estructura del Objeto de Negocios Parámetro.....	91
Tabla 32.	Estructura del Objeto de Negocios Vehículo	92
Tabla 33.	Llamadas de las páginas ASP a componentes de negocio	107

RESUMEN

En base al análisis situacional de la empresa y al resultado del examen de las variables de estudio se justificó la decisión de desarrollar una aplicación Web que le permite a la empresa ingresar al mundo del comercio electrónico ofreciendo sus servicios desde Internet.⁴

El sistema "Renta de vehículos" marca el inicio del proceso de negociación del servicio de renta de vehículos entre el cliente y la empresa cuando éste es solicitado desde Internet. Empleando el sistema, un usuario en cualquier parte del mundo se puede convertir en un nuevo cliente para la empresa y seguidamente registrar su reservación. Cumpliendo todos los requisitos exigidos para reservar un vehículo, el cliente puede estar seguro de contar con él en el tiempo que lo solicita.

El sistema permitirá a la empresa alcanzar los objetivos trazados en esta estrategia de desarrollo:

- Ofrecer una nueva alternativa para realizar una reservación de vehículos
- Captar nuevos clientes
- Promocionar la empresa

objetivos y estar siempre vigilante de los cambios que se suscitan sobre los principios y paradigmas del mundo de los negocios.

Como recomendación final, la empresa debe considerar en una próxima versión del sistema la solución de las debilidades y limitaciones con que cuenta actualmente la aplicación.

- Mejorar la imagen de la empresa
- Brindar información útil y oportuna sobre los servicios de la empresa

El sistema se convierte para la empresa en un vendedor virtual que ofrece el servicio de renta a un ilimitado número de consumidores, sin restricciones de horarios y con una disponibilidad del 100%, lo cual sería casi imposible de obtener con el recurso humano de la organización. Esto da como resultado que la empresa aumente su productividad, ya que existe una nueva fuente de captación de operaciones, y un aumento en su rentabilidad, obteniendo beneficios tangibles e intangibles, al momento de generar más ingresos económicos cuando se cristalizan las operaciones que nacieron desde la aplicación Web y el hecho de mejorar su imagen ante el mundo, respectivamente.

El análisis y diseño que se empleó para la elaboración del sistema cumple con las normas de calidad establecidas para el desarrollo de las aplicaciones como son: eficiencia, escalabilidad, confiabilidad, reusabilidad y disponibilidad.

Recomendamos a la empresa continuar con esa visión de mejorar su competitividad, eficacia, productividad y rentabilidad estando en un continuo aprendizaje, llevando a ejecución planes estratégicos para alcanzar sus

INTRODUCCIÓN

Como parte de los requisitos para la obtención del título de Licenciado en Sistemas de Información se exige la aprobación del Tópico de Graduación, siendo uno de sus requerimientos la elaboración de un sistema en donde se aplique el contenido enseñado en el mismo.

El Tópico de Graduación está compuesto por las materias:

Programación ASP (Active Server Page)

Programación con componentes COM (Component Object Model)

A nuestro equipo de trabajo le fue asignado la elaboración de un sistema de información que permita realizar reservaciones de vehículos desde Internet.

A lo largo del documento nos referiremos a esta aplicación web como "sistema" o "Sistema Renta de Vehículos".

Simulamos que el sistema fue desarrollado para una empresa llamada "**Max Rent a Car**" que tiene su oficina en Guayaquil y se dedica a la renta de vehículos. Pero, para la elaboración del sistema nuestro equipo de trabajo hizo el levantamiento de información y el respectivo análisis situacional sobre una empresa real que brinda este servicio y está localizada en la ciudad de Guayaquil. Cabe entonces resaltar que el sistema desarrollado

está sujeto a los requerimientos reales que una empresa de renta de vehículos necesita satisfacer.

En el contenido de este documento explicaremos detalladamente las características, objetivos, funcionalidades y alcance del sistema desarrollado; la arquitectura, estructura, y modelos utilizados; las especificaciones de hardware y software; e información general de lo que se impartió en la materia presentados como apéndices.

CAPITULO 1

1. ESPECIFICACIONES DEL PROYECTO.

En este capítulo se detallan los lineamientos generales que se determinaron para la elaboración del Sistema Renta de Vehículos. Se justifica el desarrollo del proyecto, se enumeran los objetivos generales que desea alcanzar la empresa por medio del proyecto, se detallan las especificaciones funcionales y no funcionales, y se describe el alcance, limitaciones y restricciones del proyecto.

1.1 Justificación

El estrepitoso crecimiento del comercio electrónico que se ha desarrollado a través de Internet, representa ya un evento histórico por sus implicaciones y consecuencias en los negocios. Por esto, la empresa **Max Rent a Car** decidió ingresar al mundo del Internet desarrollando una aplicación Web que ofrezca sus servicios y con esto formar parte del círculo de empresas que aprovechan el potencial del avance tecnológico para mejorar sus operaciones y responder mejor a sus clientes.

El resultado de las variables de estudio -tales como calidad, servicio, competidores, tecnología, imagen, disponibilidad, promociones, etc.- justifica el desarrollo de un sistema de información que le permita a la empresa entrar a un nuevo mercado.

Contar con un sitio Web diseñado de una manera funcional y atractiva, mejora la imagen de una empresa desde el punto de vista tecnológico a sus clientes.

El estilo y diseño de un sitio Web representa en gran medida las características y el enfoque empresarial de una organización, representando la imagen que esta transmite al mundo por medio del Internet.

El tener presencia en el mundo virtual le permitirá a **MAX Rent a Car** convertirse en una empresa más competitiva, moderna y eficaz.

1.2 Objetivos Generales

El Sistema Renta de Vehículos está enfocado para lograr los siguientes objetivos:

- Ofrecer una nueva alternativa para realizar una reservación de vehículos
- Captar nuevos clientes
- Promocionar el servicio de Renta de Vehículos
- Mejorar la imagen de la empresa
- Brindar información útil y oportuna sobre los servicios de la empresa

1.3 Especificaciones funcionales

Autores

Los siguientes tipos de personas son las que interactúan con el Sistema Renta de Vehículos:

Cliente

El Cliente es aquella persona natural o jurídica que se crea como tal en el sistema. Utilizando el Sistema Renta de Vehículos, un cliente puede ingresar, consultar y eliminar sus reservaciones; y actualizar sus datos generales.

Ejecutivo de Ventas

El Ejecutivo de Ventas es aquella persona que trabaja dentro de la empresa y manipula la información que es generada por las operaciones que realizan los clientes dentro del Sistema Renta de Vehículos. El Ejecutivo de Ventas no interactúa directamente con la aplicación Web, utiliza el sistema local de la empresa para realizar sus operaciones.

Datos Persistentes

Los siguientes tipos de datos necesitan ser almacenados

Datos de Cliente

Los datos del Cliente incluyen nombre, password, tipo y número de documento de identificación, teléfono, e-mail, país, ciudad. El Id-Cliente (identificación única en el sistema) es ingresada por el cliente.

Reservaciones

Los datos de Reservación incluyen Id-Cliente, fecha de reservación, fecha de entrega, Id-Vehículo, cantidad de vehículos, tipo y número de tarjeta, valor. Una reservación es identificada de manera única por el código de agencia y por un secuencial que es generado por el sistema.

Vehículos

Los datos de Vehículo incluyen marca, descripción, características, stock, número de vehículos disponibles, valor de tarifa por día, porcentaje de descuento, número de días para aplicar el descuento. Un vehículo es identificado de

manera única por el código de la agencia y por un secuencial generado por el sistema.

Reservaciones Perdidas

Los datos de Reservaciones Perdidas incluyen Id-Cliente, fecha de reservación, fecha de entrega, Id-vehículo, cantidad de vehículos, tipo y número de tarjeta, teléfono y correo. Una reservación perdida es identificada de manera única por el código de agencia y por un secuencial que es generado por el sistema.

Escenarios

Escenarios del Cliente

CREAR NUEVO CLIENTE

El usuario que accesa al sitio puede crearse como nuevo cliente. Llena un formulario con sus datos y luego acepta para que se grabe esta información. Con la identificación única el cliente podrá manejarse dentro del sistema.

El sistema le presentará un mensaje indicando que la acción se ha realizado satisfactoriamente.

INICIAR SESION DENTRO DEL SISTEMA

Esta función se realiza cuando el cliente quiere: modificar sus datos, ingresar, consultar, o eliminar una de sus reservaciones.

El cliente ingresa su id. y password. El sistema verifica que el id y password ingresados correspondan a un cliente. Si los datos ingresados son correctos, el cliente ya ha iniciado una sesión. Si los datos son incorrectos el sistema le presenta un mensaje de error.

Si el cliente no recuerda su id y su password entonces puede ir a otra página para que ingrese 3 campos: tipo del documento de identificación, número del documento de identificación y e-mail e iniciar sesión de esta manera.

En la página para iniciar sesión, el usuario tiene la opción de crearse como nuevo cliente.

CONSULTAR VEHICULOS

El cliente no necesita haber iniciado una sesión dentro del sistema.

El sistema le muestra la flota de vehículos de la empresa, con sus características y tarifas.

INGRESAR UNA RESERVACIÓN

El cliente debe primero iniciar una sesión dentro del sistema.

Una vez ya iniciada la sesión, el sistema le presenta un formulario para que realice el ingreso de los datos de la reservación. Aparecen los siguientes datos del cliente: nombre, teléfono y correo, con opción a modificar los campos: teléfono y correo.

Los datos de la reservación que debe ingresar son: fechas y horas de reservación y de entrega, tipo de vehículo, cantidad de vehículos y datos de la tarjeta de crédito. El cliente da por aceptado estos valores.

El sistema realiza las respectivas validaciones sobre los datos ingresados. Si se cumplen todas las validaciones, se le presenta en otra pantalla el valor aproximado que va a cancelar el cliente. Si el cliente acepta esta pantalla, se procede a insertar la reservación en la base de la empresa.

El sistema muestra el número de la reservación ingresada.

CONSULTAR RESERVACIONES

El cliente debe primero iniciar una sesión dentro del sistema.

El sistema le presenta todas sus reservaciones ordenadas por fecha de reservación de mayor a menor.

ELIMINAR UNA RESERVACIÓN

El cliente debe primero iniciar una sesión dentro del sistema.

El sistema le presenta todas sus reservaciones a partir de la fecha del día, donde solo puede seleccionar una para eliminar.

MODIFICAR DATOS GENERALES

El cliente debe primero iniciar una sesión dentro del sistema. El cliente puede modificar todos sus datos a excepción de la id-cliente. El sistema le presentará un mensaje indicando que la acción se ha realizado satisfactoriamente.

Escenarios del Ejecutivo de Ventas

CONFIRMACIÓN DE LA RESERVACIÓN

El ejecutivo de ventas no utiliza la aplicación Web para confirmar la reservación, pero si manipula la información generada utilizando el sistema local de la empresa.

Se comunica con el cliente para confirmar la reservación. Si la reservación no se va a realizar, cambia el estado de la reservación o la elimina.

1.4 Especificaciones no funcionales

Requerimientos de Rendimiento

Entre los requerimientos de rendimiento consideramos los siguientes aspectos:

- Para los clientes de la aplicación Web el sistema debe tener un 99% de disponibilidad durante las 24 horas del día.
- Para los empleados la aplicación Web no debe estar disponible.
- La aplicación desarrollada brinda un servicio no tan demandante debido a la naturaleza del negocio (aquí en

Guayaquil), entonces en escenarios en los cuales aumenta el volumen de demanda la aplicación resulta escalable debido a su arquitectura de desarrollo y porque se apoya en las características de conectividad de las herramientas de Microsoft.

- El tiempo de respuesta es un factor importante más no primordial. El tiempo de respuesta de la aplicación está dado en segundos, no se necesita que sea en milisegundos.
- La confiabilidad de los datos está garantizada por el sistema, aún sin embargo se puede tolerar un grado de error ya que la aplicación no maneja transacciones vitales.
- La aplicación no interactúa con sistemas heredados.
- El tiempo promedio de respuesta calculado es 5 segundos para los clientes que se conectan a Internet por medio de un módem con una velocidad de 28.8 kb. Este nivel de requerimiento es aceptable para cualquier hardware.

Requerimientos de Desplazamiento

- Los datos estarán almacenados en la existente Base de Datos SQL Server.
- Los clientes accederán al sistema desde Internet, haciendo uso de las herramientas Internet Explorer o NetScape

Requerimientos de Seguridad

- No existen restricciones de seguridad definidos en el proceso servidor transaccional (COM+), la seguridad del sistema está dada dentro de las acciones que se realizan en ella. Se requiere que los clientes inicien sesión cuando desean realizar un ingreso, consulta o eliminación de reservaciones o en la modificación de sus datos generales.

1.5 Limitaciones y restricciones

El Sistema Renta de Vehículos tiene las siguientes limitaciones:

- Está en un solo idioma
- No permite modificar datos de las reservaciones
- No se encriptan los datos referentes a la tarjeta de crédito
- No permite la verificación del cupo de la tarjeta de crédito en línea

No existen restricciones de algún agente externo.

1.6 Alcance

El Sistema Renta de Vehículos permite realizar las siguientes operaciones:

- Ingresar nuevos clientes
- Modificar datos de clientes existentes
- Consultar información de vehículos
- Ingresar una reservación
- Consultar las reservaciones de un cliente
- Eliminar una reservación de un cliente
- Iniciar sesión dentro del sistema
- Presentar políticas de reservación de la empresa
- Presentar información sobre los servicios que ofrece la empresa
- Contactarse con la empresa por medio de un correo electrónico

CAPITULO 2

2. ANALISIS DE LA INDUSTRIA Y PRODUCTO

En este capítulo se detalla el estudio que se realizó sobre la industria del producto y los elementos que la conforman, como son los consumidores o clientes, los competidores, los proveedores y los productos sustitutos. Se describe el comportamiento del consumidor ante el mercado convencional y el virtual, y se realiza una comparación entre ambos. Al final se presenta el análisis externo e interno del sistema (FODA).

2.1 Definición de la Industria y Producto

La empresa Max Rent a Car pertenece a la industria de la renta de vehículos.

El producto que se ofrece es un servicio de renta de vehículos.

Las empresas rentadoras poseen una flota con diferentes clases de vehículos para satisfacer las diferentes necesidades de sus consumidores.

El precio que se fija generalmente es por kilometraje/día y varía dependiendo del vehículo que se renta, de la cantidad de días, del tipo de cliente que lo está rentando, pudiendo ser éste un individuo o una compañía.

El valor del servicio también puede incluir ciertos gastos adicionales producidos por daños o robo al vehículo, entrega en otra localidad, kilometraje excedente, etc.

Las formas de pago que se utilizan son: tarjeta de crédito, efectivo o cheque.

Las empresas rentadoras se cubren con una garantía que debe respaldar al cliente para poder rentarle un vehículo.

Existe un contrato en el que participan cliente y empresa, donde queda asentada la responsabilidad del cliente ante el vehículo que renta.

Como valor agregado al producto, algunas empresas incluyen ciertos servicios como llenado de tanque, accesorios para viajes, promociones, etc.

2.2 Clientes

Son clientes todas aquellas personas naturales o jurídicas con acceso a Internet, que necesiten rentar uno o varios vehículos desde la ciudad de Guayaquil y que posean tarjeta de crédito.

2.2.1 Comportamiento del consumidor ante el mercado convencional

En el mercado convencional el servicio de renta de vehículos se inicia o es generado a partir de una solicitud del cliente (consumidor) hacia la empresa proveedora, que se realiza: de forma personalizada y directa en las oficinas, vía telefónica o por correo electrónico.

Para solicitar el servicio, el cliente (consumidor) debe acercarse directamente a las oficinas de la empresa o hacerlo mediante el uso del teléfono o correo electrónico.

De la flota de vehículos que la empresa le ofrece, decide si se ajustan a sus necesidades.

Dependiendo también de las políticas del negocio relacionados al cumplimiento de garantías y formas de pago, el cliente es considerado apto o no para recibir el servicio.

El cliente sella su compromiso con la empresa mediante la firma de un contrato donde pone de manifiesto su acuerdo con la negociación a la que llegó con la misma.

De acuerdo al sistema de entrega de la empresa, el cliente cuenta con el o los vehículos que solicitó y en el tiempo concordado.

2.2.2 Reacción del consumidor ante el mercado virtual

En la actualidad los sitios destinados a ofrecer un servicio especializado, como es el caso de la empresa Max Rent a Car, no tienen un tráfico de alto volumen (normalmente estimado en millones de visitas al mes) porque el modelo del negocio no brinda un servicio genérico que cautive la atención de los millones de usuarios de la red.

Pero para aquel grupo de consumidores a los cuales está dirigido el producto, contar con una nueva alternativa para rentar un vehículo les

resulta cómodo, porque lo puede realizar desde su casa, oficina o cualquier lugar del mundo en que se encuentre y en cualquier momento; es confiable y les ahorra tiempo.

Además no se necesita que el consumidor palpe de manera directa el producto final (el vehículo) ya que con la ayuda visual que se proporciona en el sitio web -como son las fotos y características de los vehículos- se cubren las expectativas por parte del usuario de conocer el producto que va a rentar.

En base a lo mencionado anteriormente podemos concluir que los usuarios del sistema tienen una reacción favorable hacia el mismo.

2.2.3 Mercado convencional vs. Mercado virtual

Los dos mercados son comparables desde el inicio del proceso de la reservación hasta el momento en que el cliente (consumidor) firma el contrato, entrega garantías, se lleva el vehículo, paga la reservación y devuelve el vehículo, porque en estas cinco últimas instancias los procesos son iguales.

Tabla 1. Comparación entre el Mercado Convencional y Mercado Virtual

Acción	Mercado Convencional	Mercado Virtual
Solicitar información sobre vehículos	El consumidor debe acercarse a las instalaciones de la empresa, comunicarse vía telefónica o pedirla por correo electrónico. Para recibir información inmediata tiene que hacerlo en horarios de oficina	El consumidor usa la aplicación Web de la empresa para buscar información sobre los vehículos. No depende de horarios
Solicitar información sobre políticas de reservación	El consumidor debe acercarse a las instalaciones de la empresa o comunicarse vía telefónica. Para recibir información inmediata tiene que hacerlo en horarios de oficina	El consumidor usa la aplicación Web de la empresa para leer las políticas de reservación. No depende de horarios. Si desea mayor información se contacta vía correo electrónico
Conocer situación actual del vehículo	El consumidor tiene que estar presente y observar por sí mismo la situación del vehículo	El consumidor se conforma con la fotografía que está publicada en el sitio

Acción	Mercado Convencional	Mercado Virtual
Verificar garantías del cliente	La empresa verifica en presencia del cliente las garantías sobre su tarjeta de crédito, en caso que el pago sea con tarjeta de crédito. Si maneja otra forma de pago, el cliente debe presentar la garantía personalmente (efectivo, cheque certificado)	La empresa verifica la garantía de la tarjeta de crédito del cliente con los datos de la reservación
Pactar valor de reservación	El cliente puede personalmente pactar con la empresa el valor de la reservación	Un valor aproximado de la reservación es dado por la aplicación Web al momento de registrar la reservación. Para negociar valor se debe mantener comunicación vía correo electrónico
Registrar reservación	El ejecutivo de ventas de la empresa utiliza el sistema local para registrar la reservación	El consumidor usa la aplicación Web para registrar la reservación

Acción	Mercado Convencional	Mercado Virtual
Confirmar reservación	<p>Cuando el cliente se dirige a las oficinas o se comunica vía teléfono no existe confirmación.</p> <p>Cuando la comunicación se ha llevado a cabo mediante el correo electrónico se confirma por este mismo medio</p>	La empresa confirma con el cliente vía correo electrónico si la reservación es posible

2.3 Competidores

Son nuestros competidores todas aquellas empresas que brinden el servicio de renta de vehículos ya sea que esté orientado al mercado convencional o al mercado virtual.

Existen en Guayaquil alrededor de 15 a 20 empresas que prestan el servicio. Dos de ellas tienen prestigio internacional. El 80% tiene agencias en las afueras del aeropuerto Simón Bolívar lo que constituye una ventaja competitiva pues llegan a uno de los mayores grupos de consumidores como son los clientes viajeros.

2.3.1 Funcionalidad de los competidores convencionales existentes

Los competidores directos que manejan el mercado convencional operan de la siguiente manera:

- Exponen su servicio directamente desde las oficinas y mediante las estrategias de publicidad dan a conocer su producto.
- Asesoran al cliente cuando éste llega a las oficinas al momento de solicitar el servicio. Verifican que el cliente cumpla con los requisitos para reservación: edad, licencia de conducir, tarjeta de crédito, efectivo o cheque certificado.
- Ayudan al cliente si es necesario en el momento de selección del vehículo para que rente el que más se ajuste a sus necesidades.
- Ponen a conocimiento del cliente el costo de la reservación y el valor adicional por los gastos deducibles (kilometraje adicional, daños al vehículo, etc.)

- Verifican la garantía del cliente si el pago se va a efectuar con tarjeta de crédito. Si el pago es en efectivo o con cheque certificado, verifica que el cliente tenga la cantidad necesaria para cubrir la garantía. Es común de las empresas de renta de vehículos, pedir al cliente dos vouchers en blanco, cuando el pago es con tarjeta de crédito; uno para el pago de la renta y otra como garantía.
- Llegado a un acuerdo con el cliente se firma un contrato para legalizar la negociación. Además se apoyan de un formulario donde se indica el estado en que es entregado él o los vehículos y éste también es firmado por el cliente.
- El vehículo es entregado al cliente.

2.3.2 Desafíos encarados por competidores convencionales

Con el fin de alcanzar sus objetivos, los competidores convencionales han tomado los siguientes desafíos:

- Ingreso al mercado virtual
- Tarifas más bajas
- Promociones

- Mayor kilometraje
- Nuevos servicios de entrega
- Otros tipos de garantía
- Renovación de vehículos
- Servicios agregados

2.4 Proveedores

Son proveedores las empresas que venden:

- Llantas
- Repuestos de vehículos
- Aceites

En la actualidad el poder de los proveedores es bajo, porque existen muchos proveedores que ofrecen estos productos y se puede escoger entre cualquiera de ellos.

2.5 Productos sustitutos

Son productos sustitutos:

- Alquiler de taxis
- Transporte público
- Prestación de vehículo a familiares o amigos

2.6 Análisis FODA del sistema

Fortalezas

- Disponible a un ilimitado número de usuarios por encontrarse en Internet, las 24 horas del día y todos los días del año, sin restricción de horarios.
- Fácil de usar y amigable al usuario.
- Almacena información consistente.
- Permite al usuario manipular la información de las reservaciones que ha registrado.
- Provee seguridad al usuario al exigirse que entre con su identificación y clave al momento de registrar reservaciones, consultarlas o eliminarlas.
- Controla por medio de un parámetro la disminución de la disponibilidad de un vehículo, es decir; se puede configurar a la aplicación para que rebaje o no el disponible de un vehículo cuando se ingrese una reservación. La aplicación puede ser un simple registrador de reservaciones, o un sistema inteligente que permita a la reservación separar el vehículo, disminuyendo disponibles.
- Mantiene información de las reservaciones que no pudieron concretarse por falta de disponibilidad de vehículos.
- Soporta un ilimitado número de usuarios concurrentes.

Oportunidades

- No hay mucha competencia en el mercado virtual para las empresas que ofrecen el servicio desde Guayaquil.
- Presentar el producto a una mayor cantidad de consumidores y por ende captar más clientes.
- Mejorar la imagen de la empresa.

Debilidades

- Se encuentra en un solo idioma
- No permite modificar los datos de las reservaciones
- Existe una dependencia con el proveedor del servicio de Internet

Amenazas

- Los usuarios que no conozcan el idioma español no accederán al sitio, lo que se convierte en una pérdida de consumidores potenciales.
- Al no permitir la modificación de los datos de la reservación, el cliente puede sentirse limitado y no querer continuar con la reservación.
- Si el proveedor del servicio de Internet tiene problemas, el sitio no estará disponible para los usuarios.

CAPITULO 3

3. MODELO DE NEGOCIOS.

En este capítulo se detalla la manera en que la empresa realiza los negocios y como el Sistema Renta de Vehículos sirve de medio para ello. Se describe el funcionamiento general del sistema de manera textual y gráficamente. Se explica el sistema de entrega, la forma de pago y las seguridades que implementa.

3.1 Definición del Modelo de Negocios

Un modelo de negocios es la "forma de hacer negocios", valga la redundancia, mediante la cual una empresa genera ingresos con base en su posicionamiento en la cadena de valor.

El modelo de negocios a través de Internet que la empresa **Max Rent a Car** respalda consiste en lo siguiente:

Por medio de una reservación realizada por el cliente desde la aplicación en internet, la empresa lo analiza y califica para determinar si se encuentra apto para recibir el servicio. La empresa se respalda de garantías al momento de entregar el vehículo al cliente. El cliente obtiene el servicio y paga por él al momento de entregar el vehículo.

El servicio que se presta está dirigido tanto a cliente individuales (Business to Customer, B2C) como corporativos (Business to Business, B2B).

3.2 Funcionalidad General

El Sistema Renta de Vehículos funciona de acuerdo a los siguientes procesos principales:

Ingreso de Reservaciones

El cliente accesa desde Internet al sitio Web de la empresa.

El cliente consulta información sobre los vehículos, tal como características y tarifas. Para esto no necesita estar registrado como cliente dentro del sistema.

El cliente consulta información sobre las políticas de reservación. Para esto no necesita estar registrado como cliente dentro del sistema.

Si el usuario es nuevo para el sistema entonces debe crearse como cliente e ingresar sus datos generales, tales como nombre, documento de identificación, teléfono, correo, etc. Todos los datos que se solicitan deben ser ingresados.

Cuando el cliente ya existe dentro del sistema debe iniciar la sesión, que consiste en ingresar su ID y su password, que le va a permitir poder accionar dentro del sistema.

El cliente ingresa los datos de su reservación. Todos los datos que se solicitan deben ser ingresados.

Al continuar el cliente con el ingreso de la reservación el sistema realizará las respectivas validaciones sobre las reglas del negocio:

Que la fecha de la reservación cumpla con las horas de anticipación que se necesitan para ingresar una nueva reservación

Que la cantidad de días de renta sea mayor o igual al número de días mínimo para rentar

Si los días que existen a partir de la fecha de ingreso hasta la fecha de reservación entran en el periodo de días que está especificado para afectar (restar) el disponible de un vehículo, entonces verificar que exista la suficiente cantidad de vehículos disponibles para poder satisfacer la reservación

Si alguna de estas reglas no se cumple, el sistema presentará un mensaje al cliente.

Cuando no se cumple con la regla de que exista disponibilidad del vehículo, el sistema almacena los datos de la reservación como una reservación perdida

Cuando la reservación cumple con las reglas del negocio, el sistema presentará al cliente el valor de la reservación.

El cliente acepta la reservación para que sea ingresada en el sistema.

Cuando se ingresa una reservación en el sistema:

Se guardan los datos de la reservación.

Se disminuye la disponibilidad del vehículo, si los días entre la fecha de reservación y la fecha del día, se encuentran dentro del parámetro "días de disminución de disponible del vehículo". En este caso la reservación queda con estado de "Procesada". Si no disminuye disponibilidad, la reservación se guarda con estado de "Pendiente".

Se actualizan datos del cliente en caso de que hayan sido modificados.

Confirmación de la Reservación

El ejecutivo de ventas de la empresa consulta las reservaciones que han realizado los clientes.

El ejecutivo de ventas confirma con los respectivos emisores de las tarjetas de crédito el cupo del cliente para determinar si tiene solvencia económica para pagar el valor de la reservación y la garantía.

El ejecutivo de ventas se comunica con el cliente vía correo electrónico o vía teléfono para indicarle que su reservación ha sido aprobada.

El cliente confirma con la empresa su reservación.

Consulta de Reservaciones

El cliente accesa desde Internet al sitio Web de la empresa.

El cliente debe iniciar la sesión, que consiste en ingresar su ID y su password, que le va a permitir poder consultar sus reservaciones.

El cliente consulta las reservaciones que ha ingresado. La consulta aparecerá ordenada por fecha de reservación de mayor a menor.

Eliminación de Reservaciones

El cliente accesa desde Internet al sitio Web de la empresa.

El cliente debe iniciar la sesión, que consiste en ingresar su ID y su password, que le va a permitir poder eliminar sus reservaciones

El sistema le muestra las reservaciones que tiene el cliente a partir de la fecha del día. El cliente solo puede eliminar una reservación a la vez.

3.2.1 Diagramas de flujo del proceso

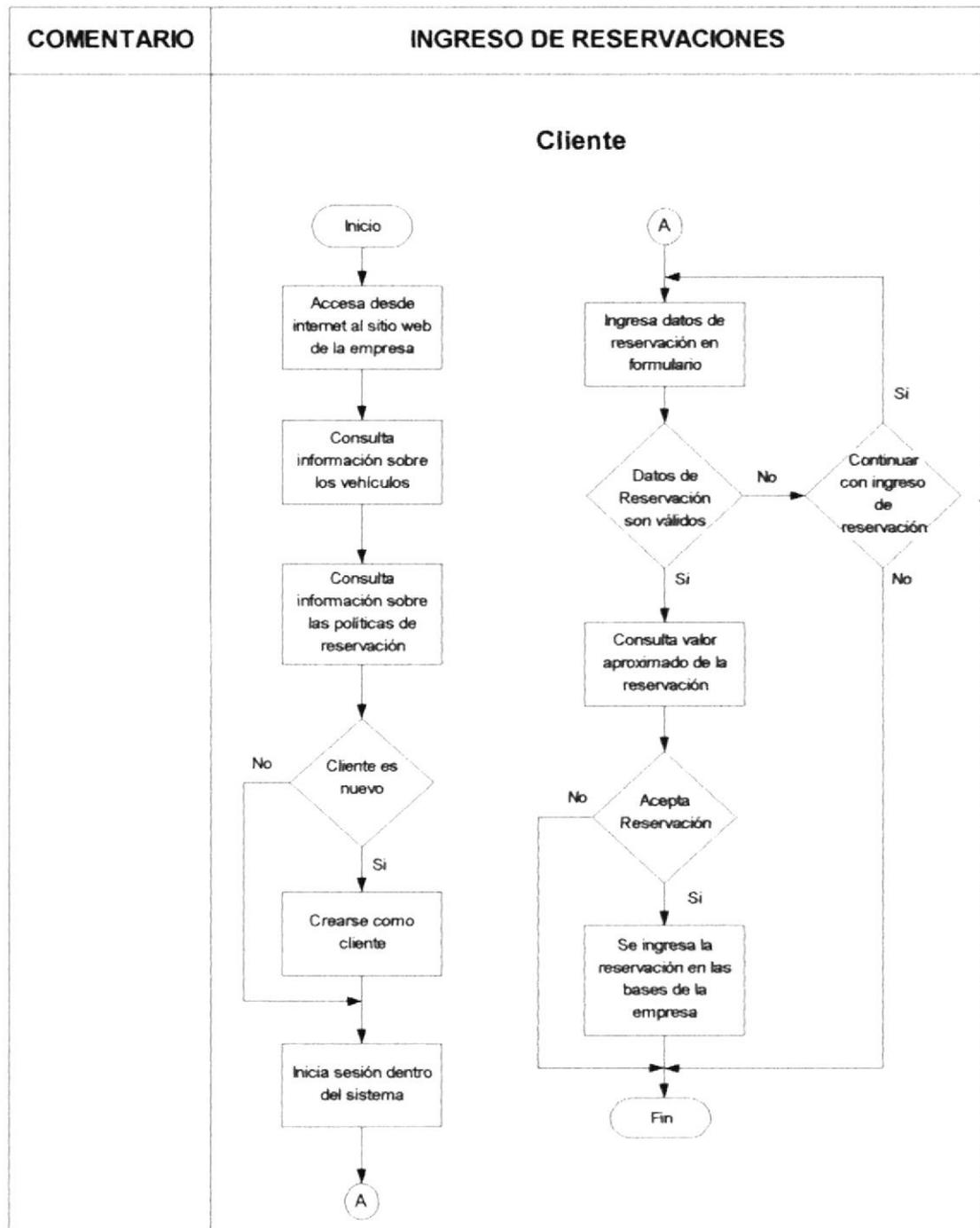


Figura 3.1. Ingreso de Reservaciones

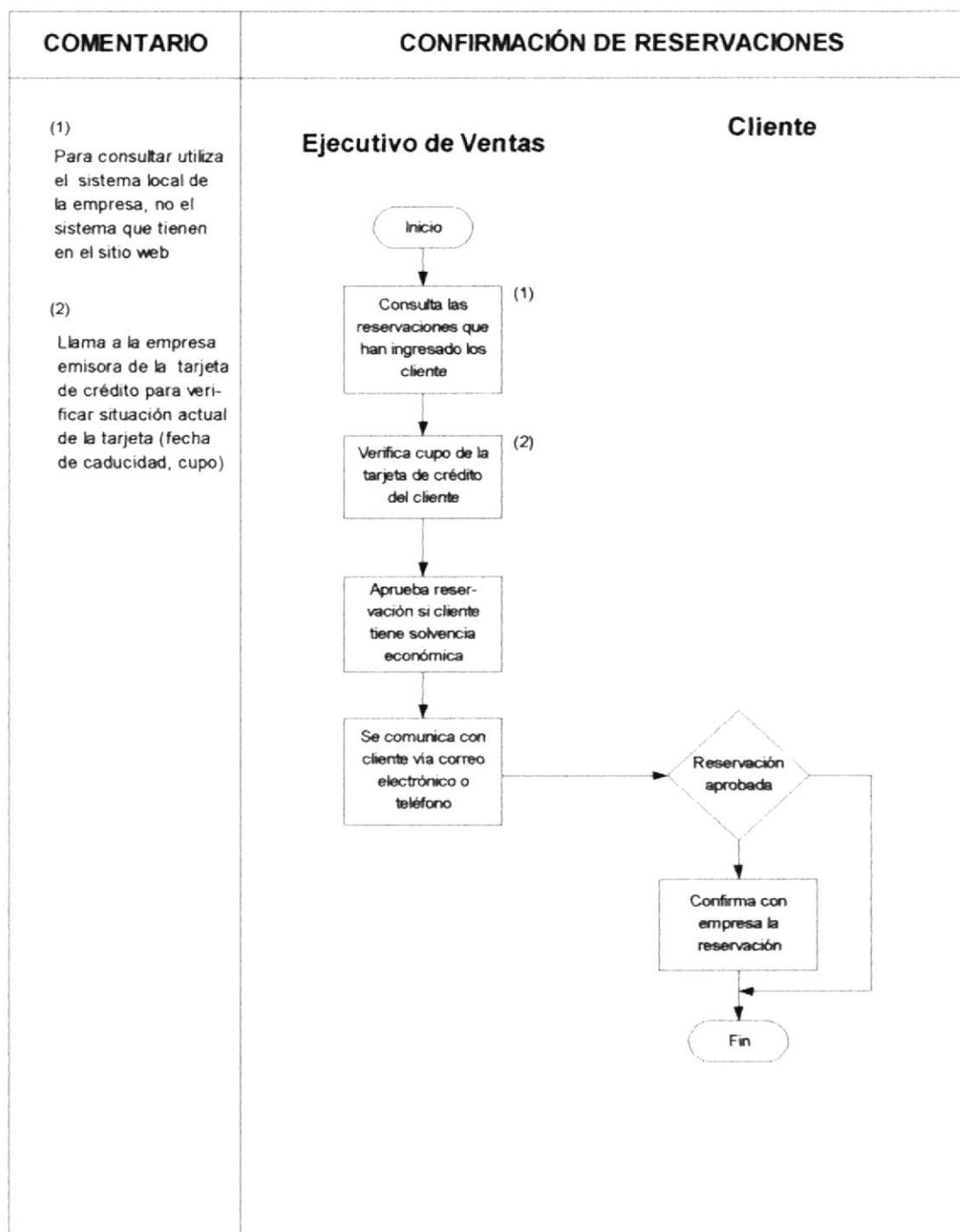


Figura 3.2. Confirmación de Reservasiones

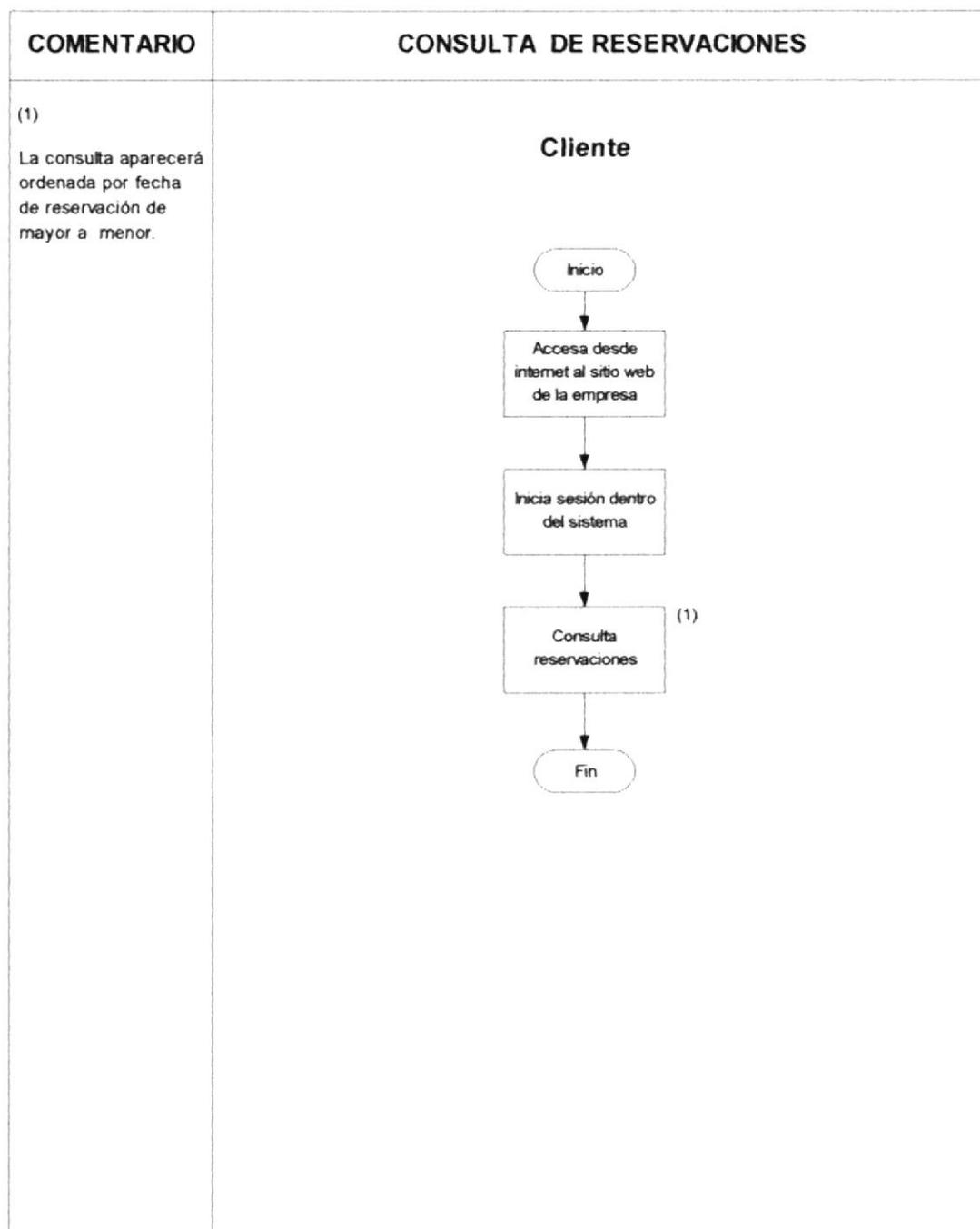


Figura 3.3. Consulta de Reservaciones

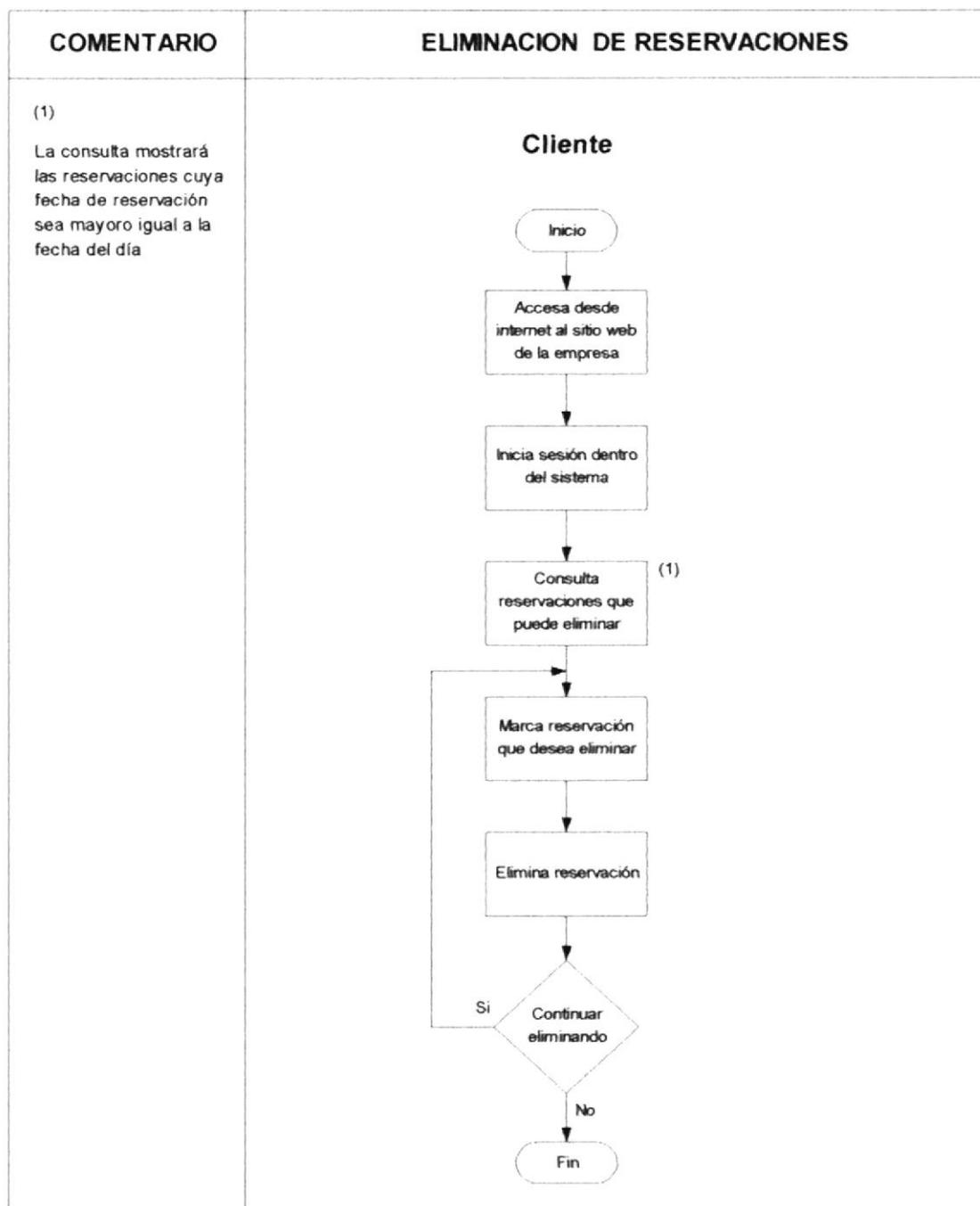


Figura 3.4. Eliminación de Reservaciones

3.3 Sistema de Entrega

El sistema de entrega no es controlado por la aplicación Web.

La empresa entrega el o los vehículos al momento que el cliente entrega las garantías, firma el contrato y firma el documento donde se especifica la situación actual del o los vehículos que el cliente ha rentado.

El vehículo es entregado generalmente en la oficina y en casos especiales en el aeropuerto.

3.4 Forma de Pago

Las formas de pago que maneja la empresa son: tarjeta de crédito, efectivo o cheque certificado.

El Sistema Renta de Vehículos al momento de ingresar los datos de la reservación, solo pide información para la forma de pago con Tarjeta de Crédito.

El sistema sólo maneja esta forma de pago porque solo con ella la empresa puede verificar la garantía y solvencia de un cliente, sin depender de su

presencia, cosa que no se puede realizar con las otras 2 formas de pago existentes (efectivo y cheque certificado).

3.5 Seguridad

La seguridad en el Sistema Renta de Vehículos es implementada para ciertas acciones. Se necesita que el cliente inicie una sesión dentro del sistema, ingresando su ID y su password.

Las acciones que requieren que el usuario haya iniciado una sesión dentro del sistema son: Ingreso, Consulta y Eliminación de reservaciones, y Modificación de datos del cliente.

CAPITULO 4

4. ARQUITECTURA DEL SISTEMA.

En este capítulo se describe la estructura del Sistema Renta de Vehículos y la justificación del modelo escogido. Se definen los tipos de procesos tanto cliente como servidores que existen en el sistema, así como también los tipos de middleware que se emplean para la comunicación entre ambos procesos. Se enumeran las tecnologías y herramientas escogidas para la implementación del sistema y los requerimientos tanto de hardware y software que se necesitan para su correcto funcionamiento.

4.1 Justificación de la selección del modelo

El desarrollo del Sistema de Renta Vehículo se realizó la bajo la arquitectura Cliente/Servidor la cual es una arquitectura de diseño de software que es el resultado de la subdivisión de una aplicación en un conjunto de procesos servidores generalmente especificados, que opera en múltiples plataformas y puede atender a un gran número de procesos clientes sobre diferentes plataformas.

Esta arquitectura brinda el equilibrio de la carga de trabajo entre el cliente y el servidor, la no dependencia de un equipo centralizado hacia los clientes y la minimización de los cuellos de botella al acceder a los datos.

Bajo esta arquitectura se escogió el modelo de tres capas el cual está conformado por:

- Capa uno o Capa de Presentación (Contiene la lógica de presentación)
- Capa dos o Capa de Negocios (Contiene la lógica de negocios y acceso a datos)
- Capa tres o Capa de Datos (Manejador de Base de Datos)

Este modelo presenta las siguientes ventajas:

- Cuenta con un pool de conexiones que permite acceder más eficazmente a la base de datos, ya que este se conecta lo más tarde posible y se desconecta lo más temprano.
- Minimiza cuellos de botella en el Acceso de Datos.
- Trabaja con un mayor número de usuarios de manera concurrente y no concurrente
- Mejora los tiempos de respuesta
- Mejora la confiabilidad (La capa de presentación no accesa directamente a la Base de Datos)

4.2 Tipos de Procesos Clientes

Los tipos procesos clientes son los que realizan los requerimientos de servicios y pueden ser:

Independientes Son los que no necesitan de un contenedor para ser ejecutados, pero esto los convierte en dependientes del Sistema Operativo porque utilizan los API's del S.O.

El sistema desarrollado cuenta con este tipo de proceso cliente en los programas de mantenimientos de tablas, las cuales fueron desarrolladas en Visual Basic.

Dependientes Necesitan de un software contenedor para poder ejecutarse, por ejemplo; las aplicaciones basadas en browser dependen de un programa navegador para poder levantarse. Pero a la vez este tipo de proceso es independiente del Sistema Operativo ya que está desarrollado en un código estándar que le permite funcionar en cualquier plataforma.

El Sistema Renta de Vehículos cuenta con este tipo de proceso cliente en la aplicación Web, la misma que fue desarrollado en Visual InterDev y se ejecuta a través de browser.

4.3 Tipos de Procesos Servidores

Los procesos servidores son aquellos que proveen servicios o satisfacen los requerimientos de los procesos clientes.

La aplicación desarrollada cuenta con los siguientes tipos de procesos servidores:

- **DataBase Server.** Recpta las peticiones del cliente por medio de las sentencias SQL, las procesa y las envía.
- **Transactional Server.** Se encarga de administrar procesos y transacciones. Es un mediador entre los procesos clientes y servidores, para ello se vale de colas y prioridades de trabajo.
- **Web Server.** Recibe requerimientos HTTP del proceso cliente (Web Server) y retorna al cliente la respuesta en código HTML.

4.4 Tipos de Middleware

El Middleware es un software que se encuentra entre el cliente y el servidor y permite la comunicación entre ambos.

Dependiendo del tipo de proceso servidor los middlewares varían y pueden ser:

- **Genérico.** Procesos de apoyo que extiende el alcance y la capacidad de los Sistemas Operativos tanto del Cliente como el servidor y de los protocolos de comunicación de las capas 3,4,5 del modelo OSI.
- **Específico.** Procesos muy ligados y dependientes del tipo de proceso servidor que se está operando.

El Sistema Renta de Vehículos cuenta con los siguientes tipos de middleware específicos:

- **COM/DCOM** Infraestructura de software sofisticada que permite que objetos se comuniquen a través de los límites de procesos, computadores y redes.

Es usado para la comunicación entre el proceso que se encarga de generar dinámicamente el código HTML que es enviado al Browser y los componentes de negocios que corren dentro del entorno de MTS.

También es empleado para la comunicación entre los componentes de negocios y los componentes de datos.

Los componentes de negocios atienden los requerimientos de las páginas activas del servidor, mientras que los componentes

de datos, que también corren dentro del entorno de MTS, son los que atienden los requerimientos de los componentes de negocios.

En ambas interacciones cliente-servidor se utiliza COM/DCOM como middleware.

- **HTTP** Es un protocolo a nivel de capa de aplicación dentro del modelo OSI, empleado para la interacción entre el Web Browser y el Web Server. El usuario del sistema debe emplear un Browser para invocar al Web Server que es el encargado de generar dinámicamente el código HTML que implementa la lógica de presentación.
- **ADO/OLEDB** ADO es un API único que sirve para acceder a datos.

OLEDB es una interfase de programación a nivel de sistema que sirve para acceder a toda clase de datos.

Los componentes de datos son los que necesitan enviar requerimientos de inserción, modificación, eliminación y consulta al servidor de base de datos, y OLEDB es el que atiende este tipo de requerimientos. Sin embargo en los componentes de datos, se utilizó ADO con la finalidad de que

el acceso a los datos no sea tan complejo y sea más fácil de programar.

4.5 Tecnologías y herramientas escogidas para la implementación

Para el desarrollo del Sistema Renta de Vehículos se escogieron como herramientas de desarrollo Visual Basic y Visual InterDev y se utilizó las tecnologías COM, ASP y SQL Server por los siguientes aspectos:

1. **Visual Basic** Es un lenguaje de programación de interfaz gráfica de usuario de fácil aprendizaje. Visual Basic 6.0 provee las siguientes facilidades:
 - Las características de acceso a datos permiten crear bases de datos, aplicaciones cliente, y componentes de servidor escalables para los formatos de las bases de datos más conocidas, incluidos Microsoft SQL Server y otras bases de datos de ámbito empresarial.
 - Las tecnologías ActiveX™ permiten usar la funcionalidad proporcionada por otras aplicaciones, como el procesador de textos Microsoft Word, la hoja de cálculo Microsoft Excel y

otras aplicaciones Windows. Se pueden incluso automatizar las aplicaciones y los objetos creados con la Edición Profesional o la Edición Empresarial de Visual Basic.

- Las capacidades de Internet facilitan el acceso a documentos y aplicaciones a través de Internet o intranet desde su propia aplicación, o la creación de aplicaciones de servidor para Internet.
- La aplicación terminada es un auténtico archivo.exe que utiliza una máquina virtual de Visual Basic que puede distribuir con toda libertad.

2. **Visual Interdev** Lenguaje de desarrollo de interfaz gráfica orientada a realizar aplicaciones Web dotándolas de más rapidez, riqueza y solidez. Por medio de Visual Interdev se pueden realizar:

- **Aplicaciones Web orientadas a datos**, utilizando un origen de datos compatible con ODBC o OLE DB, como los sistemas de administración de bases de datos de Microsoft.
- **Páginas Web de gran alcance, utilizando código HTML y secuencias de comandos** en aplicaciones Web que se benefician de los últimos avances en tecnología de

exploradores Web, por ejemplo, Microsoft® Internet Explorer 4.0, HTML dinámico y funciones multimedia.

- **Entorno de programación robusto** con un Modelo de objetos de secuencia de comandos, controles en tiempo de diseño (DTC) y un cuadro de herramientas ampliable para diseñar, probar y depurar las páginas más rápidamente.
 - **Equipos Web que pueden crear páginas de manera independiente** y mantener accesible una versión principal, o equipos de usuarios que no son programadores y trabajan en la versión maestra a través de Microsoft® FrontPage®.
 - **Soluciones integradas** que pueden incluir subprogramas o componentes creados con Microsoft® Visual Basic®, Visual C++®, Visual J++™ y Visual FoxPro®.
3. **COM** Tecnología de apoyo para el desarrollo de aplicaciones distribuidas, cuya infraestructura de software permite que objetos se comuniquen a través de los límites de procesos, computadoras y redes. La tecnología COM provee un conjunto de estándares para asegurar la interoperabilidad entre los componentes

4. **ASP** Active Server Pages (ASP) es un entorno de secuencias de comandos del servidor que se puede utilizar para crear páginas Web dinámicas o para generar eficaces aplicaciones Web. Las páginas ASP son archivos que contienen etiquetas HTML, texto y comandos de secuencias de comandos. Las páginas ASP pueden llamar a componentes ActiveX para que realicen tareas, como la conexión con bases de datos o cálculos comerciales. Con ASP se puede agregar contenidos interactivos a páginas Web o generar aplicaciones Web completas que utilicen páginas HTML como interfaz con su cliente.

4.6 Requerimientos de hardware y software para la puesta en producción

Para el normal funcionamiento del Sistema Renta de Vehículos se necesitan los siguientes requerimientos:

Del lado del Cliente:

Hardware

128 Mg o superior en memoria RAM

Procesador Pentium o superior

Software

Windows 9X, Windows Me, Windows NT
WorkStation 4.0 o superior, como sistema
operativo.

Internet Explorer o NetScape como el Web
Browser

Del lado del Servidor:**Hardware**

Procesador Pentium III o superior

40 Gigabytes de disco duro

256 Megabytes en memoria RAM

1.2 GigaHertz en Velocidad

Software

Internet Information Server 4.0 ó superior

SQL Server 7.0 ó superior

MTS 2.0 ó COM+

Windows NT ó Windows 2000, ó versiones
superiores

CAPITULO 5

5. IMPLEMENTACION DE LA CAPA DE DATOS.

En este capítulo se describe como se implementó la capa de datos llamada también *capa tres* dentro del modelo de 3 capas de la arquitectura cliente/servidor. Se describen los modelos de datos tanto físico como lógicos, el diagrama entidad relación y una explicación del proceso servidor de base de datos que usa el sistema.

5.1 Modelo Lógico de Datos

En el modelo lógico de datos se definen las entidades con sus respectivos atributos. Los atributos son características de las entidades.

A continuación se detalla el modelo lógico de datos del Sistema Renta de Vehículos:

Tabla 2. Diseño Lógico de la Entidad Agencia

Max Rent a Car		Diccionario de Datos	Gquil. 22/07/2002
Entidad: Agencia		Sistema: Renta de Vehículos	
Descripción: Contiene información sobre las localidades donde opera la empresa			
No	Atributo	Tipo de Dato	
1	Código	Alfanumérico	
2	Nombre	Alfanumérico	
3	Dirección	Alfanumérico	
4	R.U.C	Numérico	
5	Teléfono	Numérico	

Tabla 3. Diseño Lógico de la Entidad Parámetros

Max Rent a Car		Diccionario de Datos	Gquil. 22/07/2002
Entidad: Parámetros		Sistema: Renta de Vehículos	
Descripción: Contiene información sobre los parámetros generales de la empresa			
No	Atributo	Tipo de Dato	
1	Código Agencia	Alfanumérico	
2	Número de Horas mínimo antes de reservar	Numérico	
3	Hora de Entrada laborable del personal	Numérico	
4	Hora de Salida Laborable del personal	Numérico	
5	Número de días mínimo para reservar	Numérico	
6	Número de días para afectar disponible	Numérico	

Tabla 4. Diseño Lógico de la Entidad Marca

Max Rent a Car		Diccionario de Datos	Gquil. 22/07/2002
Entidad: Marca		Sistema: Renta de Vehículos	
Descripción: Contiene información sobre las marcas de vehículos que maneja la empresa			
No	Atributo	Tipo de Dato	
1	Código	Numérico	
2	Descripción	Alfanumérico	

Tabla 5. Diseño Lógico de la Entidad Tarjeta

Max Rent a Car			Diccionario de Datos			Gquil. 22/07/2002		
Entidad: Tarjeta				Sistema: Renta de Vehículos				
Descripción: Contiene información sobre los tipos de tarjetas de crédito con que negocia la empresa								
No	Atributo					Tipo de Dato		
1	Tipo de Tarjeta					Alfanumérico		
2	Descripción					Alfanumérico		

Tabla 6. Diseño Lógico de la Entidad Cliente

Max Rent a Car			Diccionario de Datos			Gquil. 22/07/2002		
Entidad: Cliente				Sistema: Renta de Vehículos				
Descripción: Contiene información sobre los clientes del sistema								
No	Atributo					Tipo de Dato		
1	Código					Alfanumérico		
2	Clave					Alfanumérico		
3	Tipo de Documento					Alfanumérico		
4	Número de Documento					Numérico		
5	Nombre					Alfanumérico		
6	Teléfono					Numérico		
7	Correo					Alfanumérico		
8	País					Alfanumérico		
9	Ciudad					Alfanumérico		
10	Fecha último movimiento					Fecha / Hora		
11	Usuario último movimiento					Alfanumérico		

Tabla 7. Diseño Lógico de la Entidad Vehículo

Max Rent a Car		Diccionario de Datos	Gquil. 22/07/2002
Entidad: Vehículo		Sistema: Renta de Vehículos	
Descripción: Contiene información sobre los vehículos que se rentan en la empresa			
No	Atributo	Tipo de Dato	
1	Código de agencia	Alfanumérico	
2	Código de vehículo	Numérico	
3	Marca	Numérico	
4	Descripción	Alfanumérico	
5	Características	Alfanumérico	
6	Cantidad de stock	Numérico	
7	Cantidad disponible	Numérico	
8	Tarifa por día	Numérico	
9	Porcentaje de descuento	Numérico	
10	Días para afectar descuento	Numérico	
11	Valor de Garantía	Numérico	
12	Fecha de último movimiento	Fecha / Hora	
13	Ruta Imagen número 1	Alfanumérico	
14	Ruta Imagen número 2	Alfanumérico	

Tabla 8. Diseño Lógico de la Entidad Reservación

Max Rent a Car		Diccionario de Datos	Gquil. 22/07/2002
Entidad: Reservación		Sistema: Renta de Vehículos	
Descripción: Contiene información sobre las reservaciones de vehículos			
No	Atributo	Tipo de Dato	
1	Código agencia	Alfanumérico	
2	Código de reservación	Numérico	
3	Código del cliente	Alfanumérico	
4	Fecha de reservación	Fecha / Hora	
5	Fecha de entrega	Fecha / Hora	
6	Código de vehículo	Numérico	
7	Cantidad	Numérico	
8	Tipo de tarjeta	Alfanumérico	
9	Número de tarjeta	Numérico	
10	Valor de la reservación	Fecha / Hora	
11	Estado de la reservación	Alfanumérico	
12	Estado afecta disponible	BIT	
13	Fecha de ingreso	Fecha / Hora	
14	Fecha de último movimiento	Fecha / Hora	
15	Usuario de último movimiento	Alfanumérico	

Tabla 9. Diseño Lógico de la Entidad Reservas Perdidas

Max Rent a Car		Diccionario de Datos	Gquil. 22/07/2002
Entidad: Reservas Perdidas		Sistema: Renta de Vehículos	
Descripción: Contiene información sobre las reservas que no se pudieron realizar por falta de stock en el inventario de vehículos			
No	Atributo	Tipo de Dato	
1	Código agencia	Alfanumérico	
2	Código de reservación	Numérico	
3	Código del cliente	Alfanumérico	
4	Fecha de reservación	Fecha / Hora	
5	Fecha de entrega	Fecha / Hora	
6	Código de vehículo	Numérico	
7	Cantidad	Numérico	
8	Tipo de tarjeta	Alfanumérico	
9	Número de tarjeta	Numérico	
10	Fecha de ingreso	Fecha / Hora	
11	Teléfono	Numérico	
12	Correo	Alfanumérico	

5.2 Modelo Físico de Datos

En el modelo físico de datos se definen las tablas del sistema con sus respectivos campos. A continuación se detalla el modelo físico de datos del Sistema Renta de Vehículos:

Tabla 10. Diseño Físico de la tabla de Agencias

Max Rent a Car		Diseño de Tablas		Gquil. 22/07/2002	
Tabla: revtb_Agencias			Sistema: Renta de Vehículos		
Descripción: Contiene información sobre las localidades donde opera la empresa					
No	Campo	Descripción	Tipo de Dato	Reglas de Validación	
1	c_idagencia	Agencia	Char(2)	Not Null	
2	c_nombre	Nombre	Varchar(40)	Not Null	
3	c_dirección	Dirección	Varchar(60)	Not Null	
4	c_ruc	Identificación fiscal	Char(13)	Not Null	
5	c_telefono	Teléfono	Char(20)	Not Null	
Campos Claves					
Primarios			Foráneos		
C_idagencia					

Tabla 11. Diseño Físico de la tabla de Parámetros

Max Rent a Car		Diseño de Tablas		Gquil. 22/07/2002	
Tabla: revtb_Parametros			Sistema: Renta de Vehículos		
Descripción: Contiene información sobre los parámetros generales de la empresa					
No	Campo	Descripción	Tipo de Dato	Reglas de Validación	
1	c_idagencia	Agencia	Char(2)	Debe existir en revtb_Agencias	
2	n_hrs_antic_reserv	# horas mínimas antes de reservar	Smallint	Not Null	
3	c_hora_ini_entrada	Hora de entrada laborable	Char(4)	Not Null	
4	c_hora_fin_entrada	Hora de salida laborable	Char(4)	Not Null	
5	n_dias_min_renta	# días mínimo para reservar	Smallint	Not Null	
6	n_dias_afecta_disp	Días mínimo p' restar existencias	Smallint	Not Null	
Campos Claves					
Primarios			Foráneos		
c_idagencia			c_idagencia en revtb_Agencias		

Tabla 12. Diseño Físico de la tabla de Marcas

Max Rent a Car		Diseño de Tablas		Gquil. 22/07/2002	
Tabla: revtb_Marca			Sistema: Renta de Vehículos		
Descripción: Contiene información sobre las marcas de los vehículos que maneja la empresa					
No	Campo	Descripción	Tipo de Dato	Reglas de Validación	
1	n_marca	Código	Smallint	Not Null	
2	c_descripcion	Descripción	Char(20)	Not Null	
Campos Claves					
Primarios			Foráneos		
n_marca					

Tabla 13. Diseño Físico de la tabla de Clientes

Max Rent a Car		Diseño de Tablas		Gquil. 22/07/2002	
Tabla: revtb_Cliente			Sistema: Renta de Vehículos		
Descripción: Contiene información sobre los clientes del sistema					
No	Campo	Descripción	Tipo de Dato	Reglas de Validación	
1	c_idcliente	Código	Char(15)	Not Null	
2	c_password	Clave	Char(15)	Not Null	
3	c_tipodoc	Tipo de documento	Char(2)	Not Null	
4	n_nrodoc	Número de documento	Char(15)	Not Null	
5	v_nombre	Nombre	Varchar(40)	Not Null	
6	c_telefono	Teléfono	Char(20)	Not Null	
7	v_correo	Correo	Char(30)	Not Null	
8	c_pais	País	Char(15)	Not Null	
9	c_ciudad	Ciudad	Char(15)	Not Null	
10	f_fecingreso	Fecha de ingreso	Datetime	Not Null	
11	f_fecultmovi	Fecha de último movimiento	Datetime	Not Null	
12	c_usultmovi	Usuario de último movimiento	Char(15)	Not Null	
Campos Claves					
Primarios			Foráneos		
C_idcliente					

Tabla 14. Diseño Físico de la tabla de Vehículos

Max Rent a Car		Diseño de Tablas		Gquil. 22/07/2002	
Tabla: revtb_Vehículo			Sistema: Renta de Vehículos		
Descripción: Contiene información sobre los vehículos que se rentan en la empresa					
No	Campo	Descripción	Tipo de Dato	Reglas de Validación	
1	c_idagencia	Agencia	Char(2)	Debe existir en revtb_Agencias	
2	n_idvehiculo	Código vehículo	Integer	Not Null	
3	n_marca	Marca	Smallint	Debe existir en revtb_Marca	
4	c_descripcion	Descripción	Char(50)	Not Null	
5	v_caracteristicas	Características	Varchar(500)	Not Null	
6	n_cantstock	Cantidad en stock	Integer	Not Null	
7	n_cantdisp	Cantidad disponible	Integer	Not Null	
8	n_tarifaxdia	Tarifa por día	Decimal 10,2	Not Null	
9	n_porcdscto	Porcentaje de descuento	Decimal(5,2)	Not Null	
10	n_diasdscto	# días para aplicar descuento	Smallint	Not Null	
11	n_garantia	Valor de la garantía	Decimal 12,2	Not Null	
12	F_fecultmovi	Fecha de último movimiento	Datetime	Not Null	
13	v_rutaimagen1	Ruta imagen 1	Char(80)	Not Null	
14	v_rutaimagen2	Ruta imagen 2	Char(80)	Not Null	
Campos Claves					
Primarios			Foráneos		
C_idagencia, n_idvehiculo			C_idagencia en revtb_agencias N_marca en revtb_Marca		

Tabla 15. Diseño Físico de la tabla de Tarjetas

Max Rent a Car		Diseño de Tablas		Gquil. 22/07/2002	
Tabla: revtb_Tarjeta			Sistema: Renta de Vehículos		
Descripción: Contiene información sobre los tipos de tarjetas de crédito con que negocia la empresa					
No	Campo	Descripción	Tipo de Dato	Reglas de Validación	
1	c_tipotarjeta	Tipo de Tarjeta	Char(2)	Not Null	
2	c_descripción	Descripción	Char(20)	Not Null	
Campos Claves					
Primarios			Foráneos		
C_tipotarjeta					

Tabla 16. Diseño Físico de la tabla de Reservasiones

Max Rent a Car		Diseño de Tablas		Guil. 22/07/2002	
Tabla: revtb_Reservación			Sistema: Renta de Vehículos		
Descripción: Contiene información sobre las reservasiones de vehículos					
No	Campo	Descripción	Tipo de Dato	Reglas de Validación	de
1	c_idagencia	Agencia	Char(2)	Debe existir en revtb_Agencias	
2	n_idreserv	Código de reservación	Decimal(10)	Not Null	
3	c_idcliente	Código de Cliente	Char(15)	Debe existir en revtb_Cliente	
4	f_fecreserv	Fecha de reservación	Datetime	Not Null	
5	f_fecentrega	Fecha de entrega	Datetime	Not Null	
6	n_idvehiculo	Código de vehículo	Integer	Debe existir en revtb_Vehiculo	
7	n_cantidad	Cantidad	Integer	Not Null	
8	c_tipotarjeta	Tipo de tarjeta	Char(2)	Debe existir en revtb_Tarjeta	
9	c_nrotarjeta	Número de tarjeta	Char(20)	Not Null	
10	n_valor	Valor de la reservación	Decimal 10,2	Not Null	
11	c_estado	Estado de la reservación	Char(1)	Not Null	
12	l_senia_restadisp	Estado afecta disponible	BIT	Not Null	
13	f_fecingreso	Fecha de ingreso	Datetime	Not Null	
14	f_fecultmovi	Fecha de último movimiento	Datetime	Not Null	
15	c_usrultmovi	Usuario de último movimiento	Char(15)	Not Null	
Campos Claves					
Primarios			Foráneos		
C_idagencia, n_idreserv			C_idagencia en revtb_Agencias c_idagencia, n_idvehiculo en revtb_Vehiculo c_tipotarjeta en revtb_Tarjeta		

Tabla 17. Diseño Físico de la tabla de Reservas Perdidas

Max Rent a Car		Diseño de Tablas		Guil. 22/07/2002	
Tabla: revtb_ReservPerdidas			Sistema: Renta de Vehículos		
Descripción: Contiene información sobre las reservas que no se pudieron realizar por falta de stock en el inventario de vehículos					
No	Campo	Descripción	Tipo de Dato	Reglas de Validación	
1	c_idagencia	Agencia	Char(2)	Debe existir en revtb_Agencias	
2	n_idreserv	Código de reservación	Decimal(10)	Not Null	
3	c_idcliente	Código de Cliente	Char(15)	Debe existir en revtb_Cliente	
4	f_fecreserv	Fecha de reservación	Datetime	Not Null	
5	f_fecentrega	Fecha de entrega	Datetime	Not Null	
6	n_idvehiculo	Código de vehículo	Integer	Debe existir en revtb_Vehículo	
7	n_cantidad	Cantidad	Integer	Not Null	
8	c_tipotarjeta	Tipo de tarjeta	Char(2)	Debe existir en revtb_Tarjeta	
9	c_nrotarjeta	Número de tarjeta	Char(20)	Not Null	
10	f_fecingreso	Fecha de ingreso	Datetime	Not Null	
11	c_telefono	Teléfono	Char(20)	Not Null	
12	v_correo	Correo	Char(30)	Not Null	
Campos Claves					
Primarios			Foráneos		
c_idagencia, n_idreserv			C_idagencia en revtb_Agencias c_idagencia, n_idvehiculo en revtb_Vehículo c_tipotarjeta en revtb_Tarjeta		

5.3 Diagrama Entidad Relación

A continuación se presenta el diagrama entidad relación del Sistema Renta de Vehículos:

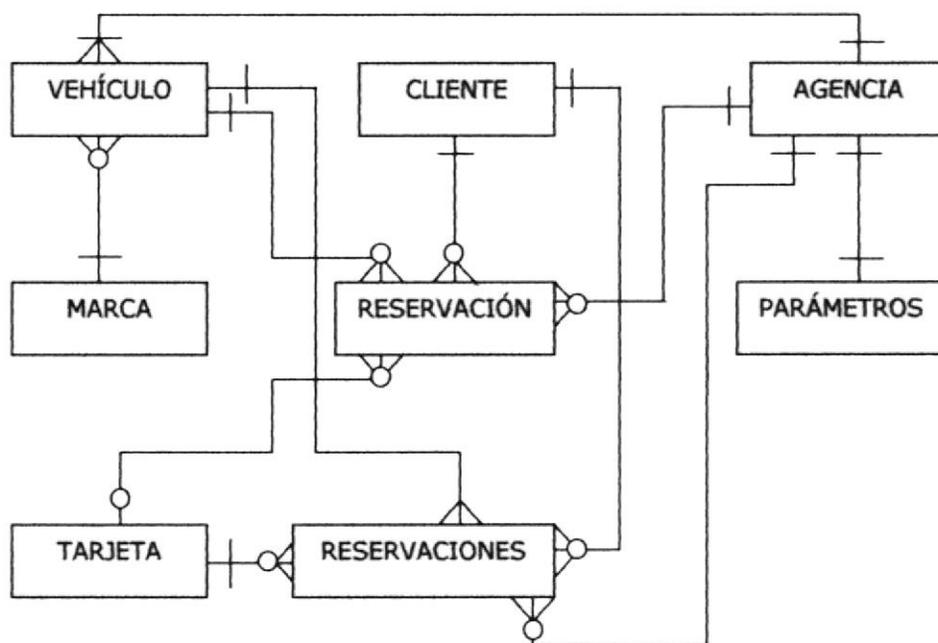


Figura 5.1 Diagrama Entidad Relación del Sistema Renta de Vehículos

5.4 Proceso Servidor de Base de Datos

Para la elaboración del Sistema Renta de Vehículos nuestro grupo de trabajo se decidió por el servidor de base de datos relacional SQL Server 7.0 ya que es un poderoso manejador y ofrece mejores resultados en rendimiento, escalabilidad y fiabilidad.

Entre las principales características que proporciona SQL Server 7.0 mencionamos las siguientes:

- Escalable desde un computador portátil a cluster de multiprocesador
- Bloqueo dinámico de filas
- Auto-gestión dinámica
- Amplia ordenación de opciones de replicación
- Desktop de SQL Server
- Servicios OLAP integrados
- Servicios de Transformación de Datos
- Microsoft English Query
- Microsoft Repository
- Integración con Microsoft Office 2000

SQL Server 7.0 escala desde bases de datos portátiles y servidores de pequeñas empresas a bases de datos de tamaño terabyte. Se integra con aplicaciones existentes, y aporta un entorno de coste efectivo para la personalización y la construcción de nuevas aplicaciones que cumplen requisitos únicos.

Escalabilidad

SQL Server 7.0 organiza más datos, transacciones y usuarios con más facilidad.

- **Escalable desde computadoras portátiles a un cluster de multiprocesador.** Puede adaptarse para organizar terabytes de datos y miles de usuarios.
- **Procesador de consultas.** Incluye un procesador de consultas rediseñado que proporciona un potente soporte para grandes bases de datos y consultas complejas.
- **Bloqueo dinámico de filas.** Soporta el bloqueo dinámico de filas, lo que lo convierte en la mejor elección en sistemas de alta calidad de online transaction processing (OLTP) y data warehousing. El administrador de bloqueos ajusta dinámicamente los recursos que utiliza para bases de datos de gran tamaño, lo que significa que no necesitará ajustar la configuración del servidor manualmente.

Facilidad de uso

Facilita a los administradores la construcción, gestión e implementación de aplicaciones de negocios, automatiza las operaciones de base de datos estándar. Además, las nuevas y sofisticadas herramientas simplifican la gestión de las complejas operaciones de gestión.

- **Auto-gestión dinámica.** Automatiza muchas de las tareas rutinarias. Los recursos de memoria y de bloqueo se ajustan automáticamente, el tamaño de los archivos crecen y disminuyen automáticamente, y

las características de auto-ajuste adaptan dinámicamente los recursos para cambiar cargas de trabajo.

- **Herramientas de perfil y ajuste.** Simplifica el proceso de localización y arreglo de problemas de base de datos permitiéndole capturar y reproducir la actividad del servidor, un nuevo analizador de consultas gráfico hace posible a evaluar consultas más fácilmente y con detenimiento, y el nuevo Index Tuning Wizard facilita el proceso de ajuste de índices.
- **Gestión de sitios múltiples.** Puede diseñar un servidor master que comunique y distribuya mensajes de tareas, alertas y eventos a servidores determinados. Además, puede realizar cambios entre grupos de servidores utilizando operaciones de arrastre-y-suelta y comandos únicos.
- **Amplia disposición de opciones de replicación.** Facilita en gran medida la instalación, la administración, la monitorización y la localización de averías. El modelo de replicación por fusión de la nueva versión hace posible que los usuarios trabajen con datos de forma libre e independiente. Con conexión o sin ella, los usuarios pueden modificar copias distribuidas de una base de datos, y obtener posteriormente el trabajo combinado y sincronizado en un solo resultado uniforme. Otra característica notable para la utilización avanzada de SQL Server es la replicación por actualización, en la que

los datos replicados por SQL Server 7.0 pueden modificarse en varios sitios.

Internet, Intranet, y Comercio

Las innovadoras características y la integración transparente con Microsoft Windows NT® y Microsoft BackOffice® hacen de SQL Server 7.0 un factor crucial en su estrategia de Internet, intranet y comercio electrónico.

- **Búsqueda a texto completo.** Suprime la necesidad de costosos dispositivos de ampliación para gestionar búsquedas de bases de datos a texto completo, soporta la búsqueda lingüística, y le permite crear índices especiales de palabras y frases relevantes en columnas seleccionadas de tablas seleccionadas.
- **Asistente web.** Contiene una opción de asistentes y de formateo que mejora el SQL Server Web Assistant, para que pueda generar archivos de HTML desde datos de SQL Server.
- **Replicación en Internet.** La replicación en Internet es más sencilla que nunca gracias a las suscripciones anónimas y al soporte incluido para la distribución en Internet. SQL Server 7.0 soporta programas que utilicen Active Server Pages (ASP).

Sistemas fijos, móviles y distribuidos

SQL Server 7.0 fue diseñado de forma tal que las organizaciones pudieran dar a sus empleados y clientes la capacidad de trabajar con datos de forma fiable desde la oficina, en Internet o de viaje.

- **SQL Server Desktop.** Con SQL Server 7.0 para Microsoft Windows® 95, Microsoft Windows 98, y Microsoft Windows NT Workstation, puede contar con un sistema de gestión de bases de datos relacionales totalmente equipado (RDBMS) que presenta un funcionamiento correcto en aplicaciones móviles y de estación de trabajo. Un código único para todas las plataformas (desde un portátil con Windows 95 a sistemas con clusters con Windows NT Server, Enterprise Edition) aporta una compatibilidad de aplicaciones total.

Data Warehousing

La información a todos los niveles de una organización puede fluir sin dificultades ni costos excesivos, para ello se sirve de las herramientas:

Servidor OLAP Integrado

Con online analytical processing (OLAP), puede desarrollar una variedad de soluciones empresariales, como la generación de informes y análisis corporativos así como el soporte a decisiones y modelado de datos. El

soporte de implementaciones multidimensionales, relacionales e híbridas OLAP, le permite escoger el modelo de datos que más le convenga a su aplicación. Los OLAP Services también ofrecen agregaciones inteligentes, que resultan en bases de datos menores, rendimiento mejorado, y tiempos de carga iniciales e incrementales más cortos.

Data Transformation Services

Los Data Transformation Services (DTS) facilitan la importación, exportación y transformación de datos heterogéneos con OLE DB, Open Database Connectivity (ODBC), o archivos de texto. DTS también suprime la necesidad de intervención por parte del usuario ya que permite la importación y transformación de datos automáticamente y de forma regular.

Microsoft English Query

Con English Query, puede crear aplicaciones que acepten consultas en lenguaje natural (preguntas escritas en inglés sencillo) en lugar de en complejas consultas de SQL. Todas las ediciones de SQL Server 7.0 incluyen English Query.

Integración con el repositorio

SQL Server 7.0 incluye Microsoft Repository y el Open Information Model, que ayudan a integrar y compartir metas entre bases de datos de SQL Server, OLAP Services, Data Transformation Services, y English Query.

Microsoft Visual Studio

La gran integración entre SQL Server 7.0 y Microsoft Visual Studio® refuerza el potencial de las organizaciones para construir soluciones de negocios escalables. La programación de lenguaje simplificada de SQL presenta la capacidad de depurar remotamente procedimientos almacenados.

CAPITULO 6

6. IMPLEMENTACION DE LA CAPA DE NEGOCIOS.

En este capítulo se describe como se implementó la capa de negocios llamada también *capa dos* dentro del modelo de 3 capas de la arquitectura cliente/servidor. Se detalla la estructura de los componentes de negocios y de acceso a datos, y se indica la tecnología que se empleó para la comunicación entre los componentes y para la comunicación con el proceso servidor de base de datos. También se explica el proceso servidor transaccional que usa el sistema.

6.1 Estructura de los componentes de Acceso a datos

Los componentes de acceso a datos encapsulan las funciones de acceso y manipulación de los datos persistentes y no persistentes. Actúan como un intermediario entre los componentes de negocios y la capa de datos. Proveen control y acceso a los datos de tal manera que los componentes de negocios no necesitan saber dónde los datos están localizados, cómo el servicio es implementado o cómo los datos son accedados.

El Sistema Renta de Vehículos cuenta con los siguientes componentes de acceso a datos:

Tabla 18. Componentes de Acceso a Datos

Componente	Objeto
dbCliente	Cliente
dbVehiculo	Vehículo
dbReservacion	Reservacion
dbReservPerdida	ReservPerdida
dbParametro	Parametro
dbTarjeta	Tarjeta
dbMarca	Marca
dbGeneral	General

6.2 Objetos distribuidos de Acceso a datos

A continuación se detalla la estructura de los objetos distribuidos de acceso a datos:

Tabla 19. Estructura del Objeto de Datos Cliente

Objeto dbCliente.Cliente

Descripción: Permite el acceso y manipulación sobre la tabla revtb_Cliente

Librería.Interfase que implementa: LibdbCliente.LibdbCliente

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/COM+): Usa Transacción

Métodos Privados	Librería.Interfase implementada	Método en interfase implementado
Buscar	LibdbCliente.LibdbCliente	Buscar
Eliminar	LibdbCliente.LibdbCliente	Eliminar
Insertar	LibdbCliente.LibdbCliente	Insertar
Modificar	LibdbCliente.LibdbCliente	Modificar
ModificarDatosContacto	LibdbCliente.LibdbCliente	ModificarDatosContacto
ObtenerporID	LibdbCliente.LibdbCliente	ObtenerporID
ObtenerporNumCorreo	LibdbCliente.LibdbCliente	ObtenerporNumCorreo

Tabla 20. Estructura del Objeto de Datos Vehículo

Objeto dbVehiculo.Vehículo

Descripción: Permite el acceso y manipulación sobre la tabla revtb_Vehiculo

Librería.Interface que **LibdbVehiculo.IdbVehiculo** implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/COM+): Usa Transacción

Métodos Privados	Librería.Interface implementada	Método en interfase implementado
Buscar	LibdbVehiculo.IdbVehiculo	Buscar
Eliminar	LibdbVehiculo.IdbVehiculo	Eliminar
Insertar	LibdbVehiculo.IdbVehiculo	Insertar
ListarDyDescripcion	LibdbVehiculo.IdbVehiculo	ListarDyDescripcion
ListarTodos	LibdbVehiculo.IdbVehiculo	ListarTodos
Modificar	LibdbVehiculo.IdbVehiculo	Modificar
ModificarDisponible	LibdbVehiculo.IdbVehiculo	ModificarDisponible
ObtenerporId	LibdbVehiculo.IdbVehiculo	ObtenerporId

Tabla 21. Estructura del Objeto de Datos Reservación

Objeto dbReservacion.Reservacion

Descripción: Permite el acceso y manipulación sobre la tabla
revtb_Reservacion

Librería.Interfase que **LibdbReservacion.LibReservacion**
implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ Usa Transacción
COM+):

Métodos Privados	Librería.Interfase implementada	Método en interfase implementado
Eliminar	LibdbReservacion.LibReservacion	Eliminar
Insertar	LibdbReservacion.LibReservacion	Insertar
ListarporCliente	LibdbReservacion.LibReservacion	ListarporCliente
ListarporClienteyFecReserv	LibdbReservacion.LibReservacion	ListarporClienteyFecReserv
ListarporEstado	LibdbReservacion.LibReservacion	ListarporEstado
Modificar	LibdbReservacion.LibReservacion	Modificar
ModificarEstado	LibdbReservacion.LibReservacion	ModificarEstado
ObtenerporId	LibdbReservacion.LibReservacion	ObtenerporId

Tabla 22. Estructura del Objeto de Datos Reservas Perdidas

Objeto dbReservPerdida.ReservPerdida

Descripción: Permite el acceso y manipulación sobre la tabla
revtb_ReservPerdidas

Librería.Interfase que **LibdbReservPerdida.LibReservPerdida**
implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ Requiere Nueva Transacción
COM+):

Métodos privados	Librería.Interfase implementada	Método en interfase implementado
Eliminar	LibdbReservPerdida.LibReservPerdida	Eliminar
Insertar	LibdbReservPerdida.LibReservPerdida	Insertar
ListarTodas	LibdbReservPerdida.LibReservPerdida	ListarTodas

Tabla 23. Estructura del Objeto de Datos Parametro

Objeto dbParametro.Parámetro

Descripción: Permite el acceso y manipulación sobre la tabla
revtb_Parametros

Librería.Interface que **LibdbParametro.LibParametro**
implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ Usa Transacción
COM+):

Métodos Privados	Librería.Interface implementada	Método en interfase implementado
Eliminar	LibdbParametro.LibParametro	Eliminar
Insertar	LibdbParametro.LibParametro	Insertar
ListarTodos	LibdbParametro.LibParametro	ListarTodos
Modificar	LibdbParametro.LibParametro	Modificar
ObtenerDatos	LibdbParametro.LibParametro	ObtenerDatos

Tabla 24. Estructura del Objeto de Datos Tarjeta

Objeto dbTarjeta.Tarjeta

Descripción: Permite el acceso y manipulación sobre la tabla
revtb_Tarjeta

Librería.Interfase que **LibdbTarjeta.LibTarjeta**
implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ Usa Transacción
COM+):

Métodos Privados	Librería.Interfase implementada	Método en interfase implementado
Eliminar	LibdbTarjeta.LibTarjeta	Eliminar
Insertar	LibdbTarjeta.LibTarjeta	Insertar
ListarTodas	LibdbTarjeta.LibTarjeta	ListarTodas
Modificar	LibdbTarjeta.LibTarjeta	Modificar
ObtenerporId	LibdbTarjeta.LibTarjeta	ObtenerporId

Tabla 25. Estructura del Objeto de Datos Marca

Objeto dbMarca.Marca

Descripción: Permite el acceso y manipulación sobre la tabla
revtb_Marca

Librería.Interfase que **LibdbMarca.IdbMarca**
implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ Usa Transacción
COM+):

Métodos Privados	Librería.Interfase implementada	Método en interfase implementado
Eliminar	LibdbMarca.IdbMarca	Eliminar
Insertar	LibdbMarca.IdbMarca	Insertar
ListarTodas	LibdbMarca.IdbMarca	ListarTodas
Modificar	LibdbMarca.IdbMarca	Modificar
ObtenerporId	LibdbMarca.IdbMarca	ObtenerporId

Tabla 26. Estructura del Objeto de Datos General

Objeto dbGeneral.General

Descripción: Tiene funciones de uso general

Librería.Interfase que
implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ Usa Transacción
COM+):

Métodos Públicos	Parámetros	Tipo de Dato Retorno
TraerSiguienteID	ByVal IdAgencia As String ByVal NombreTabla As String	Long

6.3 Comunicación con el proceso servidor de base de datos

Para establecer la comunicación entre los objetos de acceso a datos y el servidor de base de datos el sistema utiliza ADO/OLEDB.

6.4 Estructura de los componentes de Negocios

Los componentes de negocios implementan la lógica del negocio.

La lógica del negocio (también llamada reglas del negocio) es la combinación de políticas, reglas y algoritmos que constituyen el método de la empresa para hacer negocios. Ésta controla el flujo de información desde

el usuario hasta el almacenamiento en las bases de datos de la empresa y viceversa, y asegura que los requerimientos del negocio se cumplan.

Los componentes de negocios, algunas veces conocidos como objetos de negocios, son componentes que pueden representar los elementos del mundo real de un negocio –productos, clientes y empleados, por ejemplo. Como componentes, encapsulan la lógica requerida para ser usada por las aplicaciones de negocios que son el corazón de la organización.

El Sistema Renta de Vehículos cuenta con los siguientes componentes de Negocio:

Tabla 27. Componentes de Negocios

COMPONENTE	Objeto
busTarjeta	Tarjeta
busVehiculo	Vehiculo
busReservacion	Reservacion
busParametro	Parametro
busCliente	Cliente

6.5 Objetos distribuidos de Negocios

A continuación se detalla la estructura de los objetos distribuidos de negocios:

Tabla 28. Estructura del Objeto de Negocios Tarjeta

Objeto busTarjeta.Tarjeta

Descripción: Permite operaciones sobre las tarjetas de crédito, ingresar, modificar, eliminar y consultar datos
Librería.Interfase que **LibdbTarjeta.LibTarjeta**

implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ Transacción Requerida COM+):

Métodos Privados	Librería.Interfase implementada	Método en interfase implementado
Eliminar	LibdbTarjeta.LibTarjeta	Eliminar
Insertar	LibdbTarjeta.LibTarjeta	Insertar
ListarTodas	LibdbTarjeta.LibTarjeta	ListarTodas
Modificar	LibdbTarjeta.LibTarjeta	Modificar
ObtenerporId	LibdbTarjeta.LibTarjeta	ObtenerporId
Métodos Públicos	Parámetros	Tipo de Dato Retorno
Eliminar	ByVal IdTarjeta As String	
Insertar	ByVal IdTarjeta As String ByVal Descripcion As String	
ListarTodas		ADODB.Recordset
Modificar	ByVal IdTarjeta As String ByVal Descripcion As String	
ObtenerporId	ByVal IdTarjeta As String	ADODB.Recordset

Tabla 29. Estructura del Objeto de Negocios Cliente

Objeto busCliente.Cliente

Descripción: Permite operaciones sobre clientes, tales como ingresar, modificar, eliminar, inicio de sesión (logon)

Librería.Interface que **LibbusCliente.IbusCliente**

implementa:

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ Transacción Requerida

COM+):

Métodos Privados	Librería.Interface implementada	Método en interfase implementado
Eliminar	LibbusCliente.IbusCliente	Eliminar
Insertar	LibbusCliente.IbusCliente	Insertar
Logon	LibbusCliente.IbusCliente	Logon
LogonAlternativo	LibbusCliente.IbusCliente	LogonAlternativo
Modificar	LibbusCliente.IbusCliente	Modificar
ObtenerporID	LibbusCliente.IbusCliente	ObtenerporID
Métodos Públicos	Parámetros	Tipo de Dato Retorno
Eliminar	ByVal IdCliente As String	
Insertar	ByVal IdCliente As String ByVal TipoDoc As String, ByVal NumDoc As String ByVal Nombre As String ByVal Telefono As String ByVal Correo As String ByVal Pais As String ByVal Ciudad As String ByVal Password As String	
Logon	ByVal IdCliente As String ByVal Password As String	ADODB.Recordset
LogonAlternativo	ByVal TipoDoc As String	ADODB.Recordset

	ByVal NumDoc As String
	ByVal Correo As String
Modificar	ByVal IdCliente As String
	ByVal TipoDoc As String
	ByVal NumDoc As String
	ByVal Nombre As String
	ByVal Telefono As String
	ByVal Correo As String
	ByVal Pais As String
	ByVal Ciudad As String
	ByVal Password As String
ObtenerporID	ByVal IdCliente As String

Tabla 30. Estructura del Objeto de Negocios Reservación

Objeto busReservacion.Reservación

Descripción: Permite operaciones sobre las reservaciones, tales como verificar si es válida, calcular valor, ingresar, eliminar y consultar.

Librería.Interface que implementa: IbusReservacion.IbusReservacion

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/ COM+): Transacción Requerida

Métodos Privados	Librería.Interface implementada	Método en interfase implementado
CalcularTarifa	IbusReservacion.IbusReservacion	CalcularTarifa
Eliminar	IbusReservacion.IbusReservacion	Eliminar
Insertar	IbusReservacion.IbusReservacion	Insertar
ListarReservCliente	IbusReservacion.IbusReservacion	ListarReservCliente
ListarReservparaEliminar	IbusReservacion.IbusReservacion	ListarReservparaEliminar
VerificarYObtenerTarifa	IbusReservacion.IbusReservacion	VerificarYObtenerTarifa
Métodos Públicos	Parámetros	Tipo de Dato Retorno
CalcularTarifa	ByVal TarifaDia As Double ByVal PorcDscto As Double, ByVal DiasAplicarDscto As Integer ByVal DiasRenta As Long ByVal CantVehiculos As Long	Double
Eliminar	ByVal IdAgencia As String ByVal IdReserv As Long	
Insertar	ByVal IdCliente As String ByVal FechaReserv As Date ByVal HoraReserv As String ByVal FechaEntrega As Date	Long

	ByVal HoraEntrega As String	
	ByVal IdAgencia As String	
	ByVal IdVehiculo As Long	
	ByVal CantVehiculos As Long	
	ByVal TipoTarjeta As String	
	ByVal NumTarjeta As String	
	ByVal Valor As Double	
	ByVal SeniaDisminDisp As Boolean	
	ByVal Telefono_CI As String	
	ByVal Correo_CI As String	
	ByVal Telefono_Re As String	
	ByVal Correo_Re As String	
	ByVal Usuario As String	
ListarReservCliente	ByVal IdCliente As String	ADODB.Recordset
ListarReservparaEliminar	ByVal IdCliente As String	ADODB.Recordset
VerificaryObtenerTarifa	ByVal IdCliente As String	Double
	ByVal FechaReserv As Date	
	ByVal HoraReserv As String	
	ByVal FechaEntrega As Date	
	ByVal HoraEntrega As String	
	ByVal IdAgencia As String	
	ByVal IdVehiculo As Long	
	ByVal CantVehiculos As Long	
	ByVal TipoTarjeta As String	
	ByVal NumTarjeta As String	
	ByVal SeniaDisminDisp As Boolean	
	ByVal Telefono As String	
	ByVal Correo As String	

Tabla 31. Estructura del Objeto de Negocios Parametro

Objeto busParametro.Parametro

Descripción: Permite operaciones sobre los parámetros del sistema, ingresar, modificar, eliminar y consultar datos

Librería.Interfase que implementa: LibdbParametro.LibParametro

Tipo de Instanciamiento: Multiuse

Modo de Transacción (MTS/COM+): Transacción Requerida

Métodos Privados	Librería.Interfase implementada	Método en interfase implementado
Eliminar	LibdbParametro.LibParametro	Eliminar
Insertar	LibdbParametro.LibParametro	Insertar
ListarTodos	LibdbParametro.LibParametro	ListarTodos
Modificar	LibdbParametro.LibParametro	Modificar
ObtenerDatos	LibdbParametro.LibParametro	ObtenerDatos

Métodos Públicos	Parámetros	Tipo de Dato Retorno
Eliminar	ByVal IdAgencia As String	
Insertar	ByVal IdAgencia As String ByVal HorasAnticipReserv As Integer ByVal HoraEntradaLaboral As String ByVal HoraSalidaLaboral As String ByVal DiasMinimoRenta As Integer ByVal DiasAfectarDisponible As Integer	
ListarTodos		ADODB.Recordset
Modificar	ByVal IdAgencia As String ByVal HorasAnticipReserv As Integer ByVal HoraEntradaLaboral As String ByVal HoraSalidaLaboral As String	

	ByVal DiasMinimoRenta As Integer	
	ByVal DiasAfectarDisponible As Integer	
ObtenerDatos	ByVal IdAgencia As String	ADODB.Recordset

Tabla 32. Estructura del Objeto de Negocios Vehículo

Objeto busVehiculo.Vehiculo

Descripción:	Permite operaciones sobre los vehículos, ingresar, modificar, eliminar y consultar datos
Librería.Interface que implementa:	LibdbVehiculo.IdbVehiculo
Tipo de Instanciamiento:	Multiuse
Modo de Transacción (MTS/ COM+):	Usa Transacción

Métodos Privados	Librería.Interface implementada	Método en interfase implementado
Buscar	LibdbVehiculo.IdbVehiculo	Buscar
Eliminar	LibdbVehiculo.IdbVehiculo	Eliminar
Insertar	LibdbVehiculo.IdbVehiculo	Insertar
ListarIDyDescripcion	LibdbVehiculo.IdbVehiculo	ListarIDyDescripcion
ListarTodos	LibdbVehiculo.IdbVehiculo	ListarTodos
Modificar	LibdbVehiculo.IdbVehiculo	Modificar
ModificarDisponible	LibdbVehiculo.IdbVehiculo	ModificarDisponible
ObtenerporId	LibdbVehiculo.IdbVehiculo	ObtenerporId
Métodos Públicos	Parámetros	Tipo de Dato Retorno
Buscar	ByVal IdAgencia As String ByVal IdVehiculo As Long	Boolean
Eliminar	ByVal IdAgencia As String ByVal IdVehiculo As Long	
Insertar	ByVal IdAgencia As String ByVal IdVehiculo As Long	

	ByVal Marca As Integer	
	ByVal Nombre As String	
	ByVal Caracteristica As String	
	ByVal CantStock As Long	
	ByVal CantDisponible As Long	
	ByVal Tarifa As Double	
	ByVal PorcDscto As Double	
	ByVal DiasAplicaDscto As Integer	
	ByVal Rutalmagen1 As String	
	ByVal Rutalmagen2 As String	
ListarIdyDescripcion	ByVal IdAgencia As String	ADODB.Recordset
ListarTodos	ByVal IdAgencia As String	ADODB.Recordset
Modificar	ByVal IdAgencia As String	
	ByVal IdVehiculo As Long	
	ByVal Marca As Integer	
	ByVal Nombre As String	
	ByVal Caracteristica As String	
	ByVal CantStock As Long	
	ByVal CantDisponible As Long	
	ByVal Tarifa As Double	
	ByVal PorcDscto As Double	
	ByVal DiasAplicaDscto As Integer	
	ByVal Rutalmagen1 As String	
	ByVal Rutalmagen2 As String	
ModificarDisponible	ByVal IdAgencia As String	
	ByVal IdVehiculo As Long	
	ByVal Cantidad As Integer	
ObtenerporId	ByVal IdAgencia As String	ADODB.Recordset
	ByVal IdVehiculo As Long	

6.6 Comunicación entre objetos distribuidos

Para establecer la comunicación entre los objetos de negocios y los objetos de datos el sistema utiliza COM/DCOM.

Para mayor información sobre esta tecnología consulte el apéndice C.

6.7 Proceso Servidor transaccional

El proceso servidor de transacciones es el administrador de las transacciones. Actúa como mediador entre los procesos clientes y servidores. Este proceso servidor permite ejecutar otros procesos.

Para el sistema de Renta de Vehículos se escogió como proceso servidor de transacciones a COM+

Para mayor información sobre esta tecnología consulte el apéndice C.

CAPITULO 7

7. IMPLEMENTACION DE LA CAPA DE PRESENTACION.

En este capítulo se describe como se implementó la capa de presentación llamada también *capa uno* dentro del modelo de 3 capas de la arquitectura cliente/servidor. Se presenta de manera gráfica la página principal de la aplicación Web y el menú principal de la aplicación StandAlone que sirve para el mantenimiento de tablas. Se detalla de la Aplicación Web: los documentos HTML, las páginas ASP, la hoja de estilos y las llamadas a los componentes de negocios dentro de las páginas ASP. También se explica el proceso servidor web que usa el sistema.

7.1 Diseño del Sitio Web

Cuando se accesa al Sistema Renta de Vehículos se carga la página Max.html. Después de cinco segundos se carga la página Inicio.htm, que está conformado por tres frames que llaman a las páginas Encabezado.htm, Menú.htm y Principal.htm.

En la página Encabezado.html se presenta el logotipo de la empresa en tamaño pequeño. En la página Principal.html se presenta el logotipo de la empresa e información general y en la página Menu.html se presenta el menú del sistema compuesto de las opciones: Vehículo, Cliente, Reservaciones, Inicio de Sesión, Quienes Somos y Nuestros Servicios.

Si el cliente escoge la opción **Vehículo** se llamará a la página ConsVehiculo.asp, la cual lista los vehículos que dispone la empresa en su flota.

Si el cliente escoge la opción **Cliente** se le abre un submenú con las opciones Nuevo y Modificar. Cada una de estas opciones (Nuevo o Modificar) llama a la página MantCliente.asp.

Si el cliente escoge la opción **Reservación** se le abre un submenú con las opciones Ingreso, Consulta y Eliminación.

Si el cliente escoge la opción **Ingreso** se llamará a la página Reservacion1.asp.

Estando el cliente en esta página y presiona o da click sobre el botón *Continuar*, la página se llamará así misma, usará un componente de negocios para realizar verificaciones y si éstas se cumplen se llamará la página IngReservacion2.asp.

Estando el cliente en la página IngReservacion2.asp y presiona o da click sobre el botón *Aceptar*, la página se llamará así misma, usará un componente de negocios para insertar la reservación y si todo resulta ok, llamará la página mensaje.asp

Si el cliente escoge la opción **Consultar** se llamará a la página ConsReservacion.asp

Si el cliente escoge la opción **Eliminar** se llamará a la página ElimReservacion.asp.

Si el cliente escoge la opción **Inicio de Sesión** se llamará a la página IniSesion.asp.

Si el cliente escoge la opción **Quienes Somos** se llamará a la página QSomos.htm.

Y como última opción si el cliente escoge del menú **Nuestros Servicios** se llamará a la página Servicios.asp.

A continuación un gráfico para mostrar el flujo de las páginas dentro del sistema:

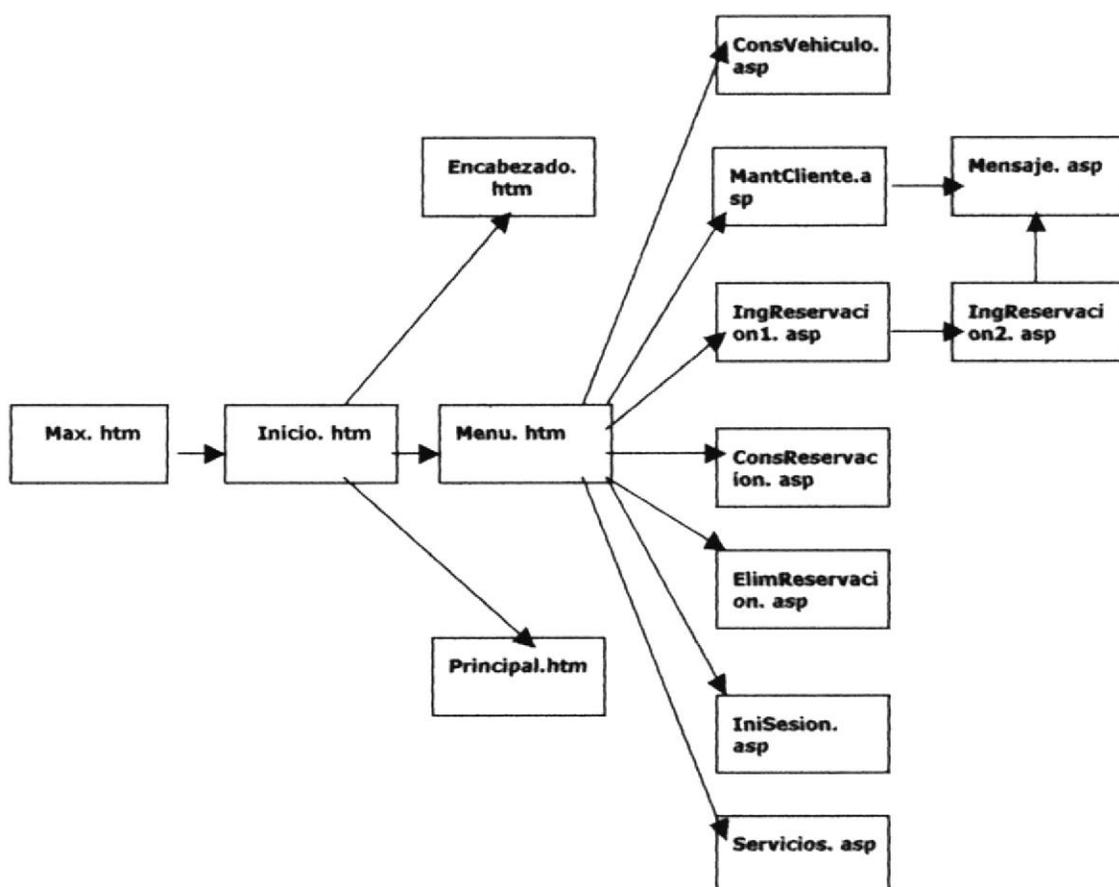


Figura 7.1. Flujo de páginas dentro del Sistema Renta de Vehículos

7.2 Cliente Standalone

La aplicación standalone permite el mantenimiento sobre algunas tablas del sistema para las operaciones de inserción, modificación, consulta y eliminación.

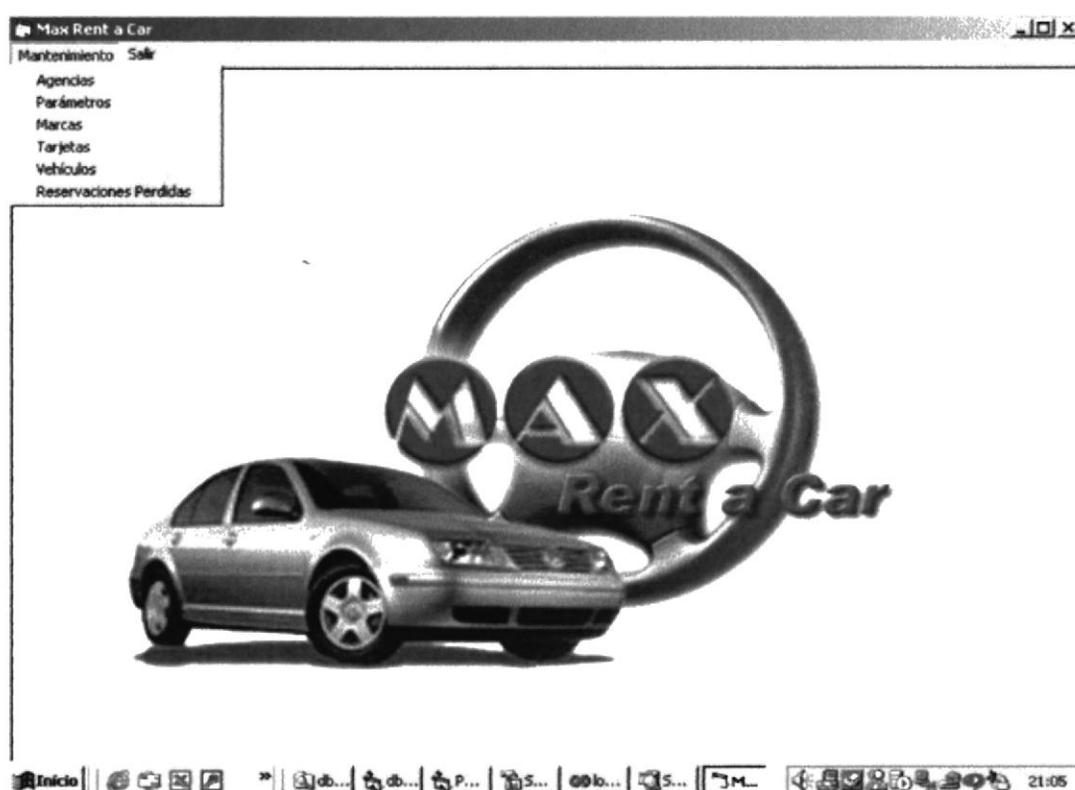


Figura 7.2. Pantalla Principal de la aplicación standalone

Tiene un menú con las siguientes opciones:

Agencias, esta opción permite el mantenimiento sobre la tabla de Agencias.

Parámetros, esta opción permite el mantenimiento sobre la tabla de Parámetros.

Marcas, esta opción permite el mantenimiento sobre la tabla de Marcas.

Tarjetas, esta opción permite el mantenimiento sobre la tabla de Tarjetas.

Vehículos, esta opción permite el mantenimiento sobre la tabla de Vehículos.

Reservaciones Perdidas, esta opción permite el mantenimiento sobre la tabla de Reservaciones perdidas.

7.3 HTML

HTML, HyperText Markup Language, es un lenguaje simple utilizado para crear documentos de hipertexto para WWW. El Sistema Renta de Vehículos utiliza los siguientes documentos html o htm:

Max.htm

Este documento da una presentación de bienvenida al sitio.

Inicio.htm

Este documento contiene tres frames y carga en ellos a los archivos:

Encabezado.htm, Menu.htm y Principal.htm.

Encabezado.htm

Este documento presenta el logotipo de la empresa en tamaño pequeño.

Principal.htm

Este documento presenta el logotipo de la empresa e información general tal como dirección, teléfono, ciudad y correo electrónico.

Menu.htm

Este documento presenta las opciones del menú del sistema y son: Vehículo, Cliente, Reservaciones, Inicio de Sesión, Quienes Somos y Nuestros Servicios. Algunas opciones del menú poseen sus respectivos submenús.

Fondo.htm

Este documento se carga como fondo en todas las demás páginas de la Aplicación.

QSomos.htm

Este documento presenta información general de la empresa

7.4 Server Script – ASP (Active Server Pages)

El Sistema Renta de Vehículos utiliza las siguientes páginas ASP:

ConsVehiculo.asp

Esta página presenta una consulta de los vehículos de la flota de la empresa. No se necesita que el usuario haya iniciado sesión. Las consultas de los vehículos permiten ver las características de los mismos y tiene un acceso directo a la página de reservaciones por medio de la foto del vehículo.

MantCliente.asp

Cuando esta página es llamada , el motor asp del lado del Web Server genera código DHTML.

Esta página permite ingresar a un nuevo cliente con sus respectivos datos o para modificar los de uno ya existente.

Si el cliente se va a crear como nuevo debe ingresar el Id y password para controlar su seguridad de acceso en futuras operaciones. Además debe ingresar el tipo del documento de identificación, número del documento de identificación, nombre, teléfono, país, ciudad y e_mail. Si existe un error al ingresar los datos, el sistema presentará un mensaje, caso contrario lo grabará en la base de datos.

Si el cliente desea modificar sus datos debe primero iniciar sesión. Puede modificar todos sus datos a excepción del Id.

El sistema presenta un mensaje al usuario cuando cualquiera de las dos opciones se ejecutaron de manera satisfactoria.

IngReservacion1.asp

Cuando esta página es llamada , el motor asp del lado del Web Server genera código DHTML.

Se verifica que el cliente haya iniciado sesión dentro del sistema.

Esta página contiene un formulario en el cual el cliente ingresa los datos de la reservación, tales como teléfono, e_mail, vehículo, cantidad de vehículos, fecha de reservación, fecha de entrega, tipo de tarjeta y número de tarjeta.

En esta página se presenta una lista con los nombres de los vehículos y los nombres de las tarjetas de crédito en controles tipo select. Para llenar esta información se usan componentes de negocios correspondientes a los objetos vehículos y tarjeta de crédito.

Cuando el cliente da click en el botón *Continuar* se valida los datos de la reservación llamando a un componente de negocios. Si no cumple con alguna de las validaciones el sistema presenta un mensaje al cliente.

Cuando todo está correcto se guardan los datos de la reservación en variables de sesión y se llama a la página *IngReservacion2.asp*.

IngReservacion2.asp

Esta página *IngReservacion2.asp* es invocada por *IngReservacion1.asp* cuando todo se ejecutó correctamente en esta última.

La página *IngReservacion2.asp* presenta el valor de la reservación. Desde esta página el cliente va a aceptar el ingreso de la reservación en las bases del sistema. Hace uso de un componente de negocios para insertar la reservación.

El sistema presenta un mensaje que incluye el número de la reservación cuando el ingreso se realizó de manera satisfactoria.

Mensaje.asp

Esta página es llamada desde otras páginas ASP y presenta un mensaje informativo al cliente.

ConsReservacion.asp

Esta página le presenta al cliente una consulta de todas sus reservaciones.

Se verifica que el cliente haya iniciado sesión dentro del sistema.

ElimReservacion.asp

Cuando esta página es llamada , el motor asp del lado del Web Server genera código DHTML.

Se verifica que el cliente haya iniciado sesión dentro del sistema.

Esta página le permite al cliente eliminar una reservación. La lista de las reservaciones que se presentan al cliente corresponden a aquellas cuya fecha de reservación es igual o mayor a la fecha del día.

IniSesion.asp

Cuando esta página es llamada , el motor asp del lado del Web Server genera código DHTML.

Esta página permite al cliente iniciar sesión dentro de la aplicación. Tiene que ingresar su Id y password y el sistema verifica que estos datos correspondan a un cliente para dar como inicio la sesión.

Servicios.asp

Esta página provee información acerca de los servicios que brinda la empresa.

7.5 CSS

CCS o (Cascading Style Sheets) es la sumatoria de procedimientos que nos llevan a controlar distintos aspectos visuales y dinámicos de las páginas web, tales como fuentes, colores, márgenes y otros. Se constituye con un pequeño archivo que se vincula a una o más páginas, con el cual se consigue un aspecto similar para todas.

El archivo **CCS** que utiliza la aplicación es **Formatocia.css**. En este archivo se incluye el formato para: etiquetas, textos, fondo de página, cabecera y detalles de tablas.

7.6 Comunicación con componentes de Negocios

Para la comunicación de las páginas ASP con los componentes de negocios se usa la tecnología COM.

Las páginas ASP que realizan llamadas a componentes de negocio son las siguientes:

Tabla 33. Llamadas de las páginas ASP a componentes de negocio

<i>Página</i>	<i>Objeto</i>	<i>Método</i>
ConsVehiculo.asp	busVehiculo.Vehiculo	ListarTodos
ConsReservacion.asp	busReservacion.Reservación	ListarReservCliente
IngReservacion1.asp	busReservacion.Reservación	VerificarYObtenerTarifa
	busVehiculo.Vehículo	ListarIDyDescripcion
	busTarjeta.Tarjeta	ListarTodas
	busParametro.Parametro	ObtenerDatos
IngReservacion2.asp	busReservacion.Reservación	Insertar
ElimReservacion.asp	busReservacion.Reservacion	ListarReservparaEliminar
		Eliminar

IniSesion.asp	BusCliente.Cliente	Logon
Servicios.asp	BusParametro.Parametro	ObtenerDatos

7.6 Proceso Servidor WEB

El proceso servidor web es parte fundamental dentro de la arquitectura del sistema. El usuario que desee usar el sistema debe 'levantar' el Web Browser y digitar la dirección URL. El Web Browser es el proceso cliente y envía requerimientos HTTP al proceso servidor que es el Web Server. La solicitud es por una página estática HTML o por un programa que se encargue de generar dinámicamente el código HTML que posteriormente será enviado al Browser.

Para el sistema de Renta de Vehículos se escogió como proceso servidor web al producto Microsoft Internet Information Server.

La aplicación se configuró con protección alta, lo que significa que se convierte en un proceso aislado al momento de ejecutarse (un proceso aislado es un proceso que corre en su propio espacio de memoria).

CONCLUSIONES

El Sistema Renta de Vehículos permite a la empresa lograr el objetivo principal de brindar a los actuales y potenciales clientes una nueva alternativa para la reservación de vehículos, permitiendo además que la empresa se encamine hacia una de las formas más actuales de negociación como es el comercio electrónico.

El haberse encaminado en las nuevas formas de hacer negocio, la transforma en una empresa más competitiva, y ante la mirada del mundo, en una empresa con avance tecnológico a la expectativa de los nuevos cambios en la actual era de la economía digital.

El sistema se convierte para la empresa en un vendedor virtual que ofrece el servicio de renta a un ilimitado número de consumidores, sin restricciones de horarios y con una disponibilidad del 100%, lo cual sería casi imposible de obtener con el recurso humano de la organización. Esto da como resultado que la empresa aumente su productividad, ya que existe una nueva fuente de captación de operaciones, y un aumento en su rentabilidad, obteniendo beneficios tangibles e intangibles, al momento de generar más ingresos económicos cuando se cristalizan las operaciones que nacieron desde la aplicación Web y el hecho de mejorar su imagen ante el mundo, respectivamente.

Por la manera funcional y atractiva en que fue diseñada la aplicación, la empresa alcanza otro de sus objetivos estratégicos que es mejorar su imagen. Esta nueva proyección puede cautivar a un mayor número de consumidores potenciales, esperando con ello, un aumento de clientes que reciban nuestros servicios.

Poner a disposición de los consumidores información útil sobre los servicios, ayuda en la relación cliente-empresa porque al mantenerlo informado sobre las políticas de reservaciones, promociones especiales, y cualquier otra información relacionada al producto, servirá como fuente de apoyo al cliente en el momento que realiza sus operaciones.

Como fortaleza tecnológica, el análisis y diseño que se empleó para la elaboración del sistema cumple con las características establecidas para el desarrollo de las aplicaciones como son: eficiencia, escalabilidad, confiabilidad, reusabilidad y disponibilidad.

Las herramientas y tecnologías utilizadas en el desarrollo de la aplicación pertenecen a una de las más importantes industrias del software, Microsoft Corp., lo que garantiza la confiabilidad y rendimiento de las mismas, un amplio soporte técnico y un gran número de ventajas y soluciones que solo una prestigiosa y líder empresa puede respaldar en sus productos.

Con lo anteriormente expuesto podemos concluir que el Sistema Renta de Vehículos es una estrategia eficaz para alcanzar los objetivos trazados por la empresa en su búsqueda de la excelencia.

RECOMENDACIONES

Con este sistema desarrollado la empresa ha emprendido su marcha hacia una de las maneras más estrepitosas y revolucionarias de hacer negocio que se inició hace unos años, el comercio electrónico.

Así como ha dado este paso, la empresa debe continuar con esa visión de mejorar su competitividad, eficacia, productividad y rentabilidad estando en un continuo aprendizaje, llevando a ejecución planes estratégicos para alcanzar sus objetivos y estar siempre vigilante de los cambios que se suscitan sobre los principios y paradigmas del mundo de los negocios.

Es necesario que la empresa mantenga la comprensión de que los datos forman parte de los activos vitales de una organización, por lo tanto la calidad, confiabilidad y disponibilidad de los mismos deben ser uno de los objetivos a alcanzar en cada sistema de información que se desarrolle o adquiera. Pero también la empresa debe mantener un estado armónico con los demás elementos que constituyen su razón de ser como son su recurso humano, sus clientes, sus proveedores y todo aquello que la conforma.

Como recomendación final, la empresa debe considerar en una próxima versión del sistema la solución de las debilidades y limitaciones con que cuenta actualmente la aplicación.

APÉNDICES

APÉNDICE A Arquitectura Cliente / Servidor

Definición

El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos estos recursos de máquina, de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe se traspasa a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor, al proceso que responde a las solicitudes.

Es el modelo de interacción más común entre aplicaciones en una red. No forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo.

Los principales componentes del esquema cliente/servidor son entonces los Clientes, los Servidores y la infraestructura de comunicaciones.

En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los Clientes interactúan con el usuario, usualmente en forma gráfica. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Los clientes realizan generalmente funciones como:

- √ Manejo de la interfase del usuario.
- √ Captura y validación de los datos de entrada.
- √ Generación de consultas e informes sobre las bases de datos.

Los Servidores proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloques, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores, la plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Por esta razón se utilizan PCs poderosas, estaciones de trabajo,

minicomputadores o sistemas grandes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema ("login"), auditoria y recuperación y contabilidad. Usualmente en los servidores existe algún tipo de servicio de bases de datos. En ciertas circunstancias, este término designará a una máquina. Este será el caso si dicha máquina está dedicada a un servicio particular, por ejemplo: servidores de impresión, servidor de archivos, servidor de correo electrónico, etc.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- √ Gestión de periféricos compartidos.
- √ Control de accesos concurrentes a bases de datos compartidas.
- √ Enlaces de comunicaciones con otras redes de área local o extensa.
- √ Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos

básicos de direccionamiento y transporte. La mayoría de los sistemas Cliente/Servidor actuales, se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Entre las principales características de la arquitectura cliente / servidor, se puede destacar las siguientes:

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

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios.

Como ejemplos de servidores pueden citarse servidores de ventanas como X-windows, servidores de archivos como NFS, servidores para el manejo de bases de datos (como los servidores de SQL), servidores de diseño y manufactura asistidos por computador, etc.

Componentes esenciales de la infraestructura Cliente/Servidor

Una infraestructura Cliente/Servidor consta de tres componentes esenciales, todos ellos de igual importancia y estrechamente ligados:

Plataforma Operativa: La plataforma deberá soportar todos los modelos de distribución Cliente/Servidor, todos los servicios de comunicación, y deberá utilizar, preferentemente, componente estándar de la industria para los servicios de distribución. Los desarrollos propios deben coexistir con las aplicaciones estándar y su integración deberá ser imperceptible para el usuario. Igualmente, podrán acomodarse programas escritos utilizando diferentes tecnologías y herramientas.

Entorno de Desarrollo de Aplicaciones: Debe elegirse después de la plataforma operativa. Aunque es conveniente evitar la proliferación de herramientas de desarrollo, se garantizará que el enlace entre éstas y el middleware no sea excesivamente rígido. Será posible utilizar diferentes herramientas para desarrollar partes de una aplicación. Un entorno de aplicación incremental, debe posibilitar la coexistencia de procesos cliente y servidor desarrollados con distintos lenguajes de programación y/o herramientas, así como utilizar distintas tecnologías (por ejemplo, lenguaje

procedural, lenguaje orientado a objetos, multimedia), y que han sido puestas en explotación en distintos momentos del tiempo.

Gestión de Sistemas: Estas funciones aumentan considerablemente el costo de una solución, pero no se pueden evitar. Siempre deben adaptarse a las necesidades de la organización, y al decidir la plataforma operativa y el entorno de desarrollo, es decir, en las primeras fases de la definición de la solución, merece la pena considerar los aspectos siguientes:

- √ ¿Qué necesitamos gestionar?
- √ ¿Dónde estarán situados los procesadores y estaciones de trabajo?
- √ ¿Cuántos tipos distintos se soportarán?
- √ ¿Qué tipo de soporte es necesario y quién lo proporciona?

No es en absoluto recomendable el intento de una infraestructura completa desde el principio, ya que las tecnologías pueden no responder a tiempo a las necesidades prioritarias del negocio. El enfoque más adecuado está en un sistema y una plataforma de aplicación conceptuales, y una arquitectura construida incrementalmente y ampliada a medida que se desarrollan nuevas aplicaciones.

La Plataforma Operativa, el Middleware y el Entorno de Desarrollo de Aplicaciones están relacionados entre sí. Las necesidades de apertura pueden condicionar la elección de la plataforma o del middleware, de igual manera que lo condiciona una determinada herramienta de desarrollo.

El software de aplicación puede influir en la plataforma del sistema, y el tiempo disponible para la primera aplicación puede implicar algún tipo de compromiso. Por lo tanto, es necesario fijar los objetivos y el modo de conseguirlos en cada caso concreto: una Metodología de Infraestructura para Sistemas Distribuidos que permita definir una infraestructura para el sistema Cliente/Servidor y evalúe la puesta en marcha del proyecto sobre una base racional.

El enfoque estructurado de dicha Metodología comprende los pasos siguientes:

- √ Captación de las necesidades: Definir, analizar y evaluar los requerimientos del negocio con las aportaciones tecnológicas.
- √ Diseño conceptual en el que se sitúan los principales bloques funcionales y de datos del sistema, mostrando la relación y comunicación entre ambos.
- √ Detalle de los principales componentes funcionales, selección de procesos, determinando los principios que deben aplicarse a la selección de software o diseño de los módulos.

Al final de los tres pasos anteriores, se definen los conceptos del sistema y la infraestructura tecnológica, sin concretar, todavía, en productos o plataformas específicos.

Por último, se llega a la selección de plataformas y principales productos y componentes para la implantación. El resultado es la descripción de una solución que incluye infraestructura tecnológica, plataformas y productos.

Características funcionales

Esta arquitectura se puede clasificar en cinco niveles, según las funciones que asumen el cliente y el servidor, tal y como se puede ver en el siguiente diagrama:

En el **primer nivel** el cliente asume parte de las funciones de presentación de la aplicación, ya que siguen existiendo programas en el servidor, dedicados a esta tarea. Dicha distribución se realiza mediante el uso de productos para el "maquillaje" de las pantallas del mainframe. Esta técnica no exige el cambio en las aplicaciones orientadas a terminales, pero dificulta su mantenimiento. Además, el servidor ejecuta todos los procesos y almacena la totalidad de los datos. En este caso se dice que hay una presentación distribuida o embellecimiento.

En el **segundo nivel**, la aplicación está soportada directamente por el servidor, excepto la presentación que es totalmente remota y reside en el cliente. Los terminales del cliente soportan la captura de datos, incluyendo

una validación parcial de los mismos y una presentación de las consultas. En este caso se dice que hay una presentación remota.

En el **tercer nivel**, la lógica de los procesos se divide entre los distintos componentes del cliente y del servidor. El diseñador de la aplicación debe definir los servicios y las interfases del sistema de información, de forma que los papeles de cliente y servidor sean intercambiables, excepto en el control de los datos, que es responsabilidad exclusiva del servidor. En este tipo de situaciones se dice que hay un proceso distribuido o cooperativo.

En el **cuarto nivel** el cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada. En esta situación se dice que hay una gestión de datos remota.

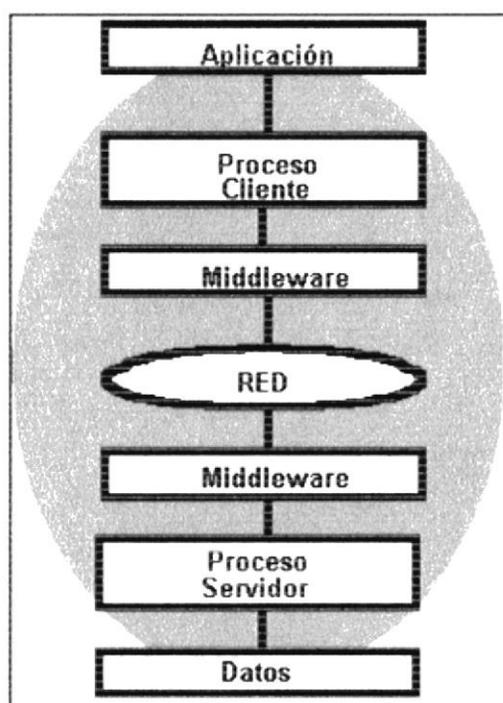
En el **quinto y último nivel**, el reparto de tareas es como en el anterior y además el gestor de base de datos divide sus componentes entre el cliente y el servidor. Las interfaces entre ambos, están dentro de las funciones del gestor de datos y, por lo tanto, no tienen impacto en el desarrollo de las aplicaciones. En este nivel se da lo que se conoce como bases de datos distribuidas.

Características físicas

El diagrama del punto anterior da una idea de la estructura física de conexión entre las distintas partes que componen una arquitectura cliente / servidor. La idea principal consiste en aprovechar la potencia de los

ordenadores personales para realizar, sobre todo, los servicios de presentación y, según el nivel, algunos procesos o incluso algún acceso a datos locales. De esta forma se descarga al servidor de ciertas tareas para que pueda realizar otras más rápidamente.

También existe una plataforma de servidores que sustituye al ordenador central tradicional y que da servicio a los clientes autorizados. Incluso a veces el antiguo ordenador central se integra en dicha plataforma como un servidor más. Estos servidores suelen estar especializados por funciones (seguridad, cálculo, bases de datos, comunicaciones, etc.), aunque, dependiendo de las dimensiones de la instalación se pueden reunir en un servidor una o varias de estas funciones.



Las unidades de almacenamiento masivo en esta arquitectura, se caracterizan por incorporar elementos de protección que evitan la pérdida de datos y permiten multitud de accesos simultáneos (alta velocidad, niveles RAID, etc.).

Para la comunicación de todos estos elementos se emplea un sistema de red que se encarga de transmitir la información entre clientes y servidores. Físicamente consiste en un cableado (coaxial, par trenzado, fibra óptica, etc.) o en conexiones mediante señales de radio o infrarrojas, dependiendo de que la red sea local (LAN o RAL), metropolitana (MAN) o de área extensa (WAN).

Para la comunicación de los procesos con la red se emplea un tipo de equipo lógico denominado middleware que controla las conversaciones. Su función es independizar ambos procesos (cliente y servidor). La interfase que presenta es la estándar de los servicios de red, hace que los procesos "piensen" en todo momento que se están comunicando con una red.

Características lógicas

Una de las principales aportaciones de esta arquitectura a los sistemas de información, es la interfase gráfica de usuario. Gracias a ella se dispone de un manejo más fácil e intuitivo de las aplicaciones mediante el uso de un dispositivo tipo ratón. En esta arquitectura los datos se presentan, editan y validan en la parte de la aplicación cliente.

En cuanto a los datos, cabe señalar que en la arquitectura cliente / servidor se evitan las duplicidades (copias y comparaciones de datos), teniendo siempre una imagen única y correcta de los mismos, disponible en línea para su uso inmediato.

Todo esto tiene como fin que el usuario de un sistema de información soportado por una arquitectura cliente / servidor, trabaje desde su estación de trabajo con distintos datos y aplicaciones, sin importarle dónde están o dónde se ejecuta cada uno de ellos.

La Importancia de un Middleware Robusto y Escalable en las Soluciones Empresariales Cliente/Servidor.

Si su organización es como la mayoría de las empresas de hoy en día, con los centros de información distribuidos geográficamente, al igual que sus clientes y oportunidades de negocios, se hace necesaria una solución tecnológica en informática que cubra todos sus factores críticos de éxito.

Con el paso de los años y los adelantos en la tecnología, la forma de procesar los datos dentro de las compañías y la forma de utilizar los resultados obtenidos ha tenido un constante cambio.

El éxito futuro de las compañías y su permanencia en el mercado, está directamente relacionado con la capacidad de adecuación de estas nuevas tecnologías y su correcta utilización para satisfacer las necesidades de

información dentro de la empresa. En el proyecto de rediseño de la aplicación, la estrategia que se utiliza incluye el concepto de middleware.

Definición de Middleware

El middleware es un módulo intermedio que actúa como conductor entre dos módulos de software. Para compartir datos, los dos módulos de software no necesitan saber cómo comunicarse entre ellos, sino cómo comunicarse con el módulo de middleware.

El middleware debe ser capaz de traducir la información de una aplicación y pasarla a la otra. El concepto es muy parecido al de ORB (Object Request Broker) que permite la comunicación entre objetos y servicios de gestión básicos para aplicaciones de objetos distribuidos.

En una aplicación cliente / servidor el middleware reside entre la aplicación cliente y la aplicación del sistema host que actúa como servidor.

El módulo middleware puede definirse también en términos de programación orientada a objetos. El módulo identifica diferentes objetos y conoce qué propiedades tienen asociadas, por lo que puede responder a peticiones referentes a los mismos.

Tipos de Middleware

Middleware Genérico: Da apoyo y mejora la capacidad del Sistema Operativo de los computadores donde se están corriendo los procesos servidores y clientes;

Ejemplo: NetBios, Sockets, RPC, etc. El Middleware Genérico usan las capas 3,4 y 5 del Modelo OSI.

Middleware Especifico: Esta relacionado con el tipo del proceso servidor;

Ejemplo

Middleware Web Server: HTTP

Middleware DataBase Server: ODBC, OLEDB, ADO

Middleware Object Server: COM, CORBA

Características Generales de Middleware

Simplifica el proceso de desarrollo de aplicaciones.

- √ Es el encargado del acceso a los datos: acepta las consultas y datos recuperados directamente de la aplicación y los transmite por la red. También son responsable de enviar de vuelta a la aplicación, los datos de interés y de la generación de códigos de error.
- √ Es diferente desarrollar aplicaciones en un entorno middleware que la utilización de APIs directos del sistema. El middleware debe ser capaz de manejar todas las facilidades que posee el sistema operativo y esto, no es sencillo. Por eso, muchas veces se pierde potencia con la

utilización del middleware en lugar de las APIs del sistema operativo directamente.

- √ La adopción dentro de una organización implica la utilización de unos paquetes de software específicos para desarrollar estos módulos. Esto liga a un suministrador y a su política de actualización del producto, que puede ser distinta que la de actualización de los sistemas operativos con los que se comunica el módulo middleware.

Ventajas e Inconvenientes de Cliente/Servidor

Ventajas

a) Aumento de la productividad:

- √ Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos.
- √ Mediante la integración de las aplicaciones cliente / servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes.
- √ Una interfase gráfica de usuario consistente, reduce el tiempo de aprendizaje de las aplicaciones.

b) Menores costos de operación:

- √ La existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes.
- √ Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión. Por ejemplo, la compartición de servidores (habitualmente caros) y dispositivos periféricos (como impresoras) entre máquinas clientes, permite un mejor rendimiento del conjunto.
- √ Se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- √ Proporcionan un mejor acceso a los datos. La interfase de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información.
- √ El movimiento de funciones desde un ordenador central hacia servidores o clientes locales, origina el desplazamiento de los costos

de ese proceso hacia máquinas más pequeñas y por tanto, más baratas.

C) Mejora en el rendimiento de la red:

- √ Las arquitecturas cliente/servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Entonces, la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios.
- √ Se puede integrar PCs con sistemas medianos y grandes, sin que todas las máquinas tengan que utilizar el mismo sistema operacional.
- √ Si se utilizan interfaces gráficas para interactuar con el usuario, el esquema Cliente/Servidor presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red, pues ésta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- √ Tanto el cliente como el servidor pueden escalar para ajustarse a las necesidades de las aplicaciones. Las UCPs utilizadas en los

- respectivos equipos, pueden dimensionarse a partir de las aplicaciones y el tiempo de respuesta que se requiera.
- √ La existencia de varias UCPs proporciona una red más fiable: una falla en uno de los equipos, no significa necesariamente que el sistema deje de funcionar.
 - √ En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros, con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema.
 - √ La arquitectura modular de los sistemas cliente / servidor, permite el uso de ordenadores especializados (servidores de base de datos, servidores de ficheros, estaciones de trabajo para CAD, etc.).
 - √ Permite centralizar el control de sistemas que estaban descentralizados, como por ejemplo la gestión de los ordenadores personales que antes estuvieron aislados.
 - √ Es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets o el RPC).
 - √ El esquema Cliente/Servidor contribuye además a proporcionar a las diferentes direcciones de una institución soluciones locales, pero

permitiendo además la integración de la información relevante a nivel global.

Inconvenientes de Cliente/Servidor

Hay una alta complejidad tecnológica al tener que integrar una gran variedad de productos.

Por una parte, el mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.

- √ Requiere un fuerte rediseño de todos los elementos involucrados en los sistemas de información (modelos de datos, procesos, interfaces, comunicaciones, almacenamiento de datos, etc.). Además, en la actualidad existen pocas herramientas que ayuden a determinar la mejor forma de dividir las aplicaciones entre la parte cliente y la parte servidor.

Por un lado, es importante que los clientes y los servidores utilicen el mismo mecanismo (por ejemplo sockets o RPC), lo cual implica que se deben tener mecanismos generales que existan en diferentes plataformas.

Además de lo anterior, se cuenta con muy escasas herramientas para la administración y ajuste del desempeño de los sistemas.

- √ Es más difícil asegurar un elevado grado de seguridad en una red de clientes y servidores que en un sistema con un único ordenador centralizado. Se deben hacer verificaciones en el cliente y en el servidor. También se puede recurrir a otras técnicas como el encriptamiento.

Un aspecto directamente relacionado con el anterior, es el de cómo distribuir los datos en la red. En el caso de una empresa, por ejemplo, éste puede ser hecho por departamentos, geográficamente, o de otras maneras. Además, hay que tener en cuenta que en algunos casos, por razones de confiabilidad o eficiencia se pueden tener datos replicados, y que puede haber actualizaciones simultáneas.

- √ A veces, los problemas de congestión de la red pueden degradar el rendimiento del sistema por debajo de lo que se obtendría con una única máquina (arquitectura centralizada). También la interfase gráfica de usuario puede a veces ralentizar el funcionamiento de la aplicación.
- √ El quinto nivel de esta arquitectura (bases de datos distribuidas) es técnicamente muy complejo y en la actualidad, hay muy pocas implantaciones que garanticen un funcionamiento totalmente eficiente.
- √ Existen multitud de costos ocultos (formación en nuevas tecnologías, licencias, cambios organizativos, etc.) que encarecen su implantación.

Implicaciones del modelo Cliente/Servidor

Para llevar a cabo con éxito la implantación de un modelo Cliente/Servidor, el departamento de Sistemas de Información debe participar activamente en los proyectos. Si se quiere evitar la proliferación de distintos sistemas y plataformas de aplicación y la incompatibilidad entre los distintos desarrollos, el departamento informático debe asumir la responsabilidad corporativa en la selección de:

- √ Plataformas del sistema (por ejemplo, procesadores para los servidores y estaciones de trabajo para los usuarios, así como sistemas operativos soportados).
- √ Estándares, formatos y protocolos aplicables al hardware y software de sistemas y aplicaciones.
- √ Software para los componentes de middleware de las plataformas operativas y herramientas de desarrollo de aplicaciones.
- √ Software de aplicación estándar disponible en el mercado.

La tendencia imparable de un mayor control de la aplicación por parte del usuario, modifica las exigencias de infraestructura relativas a:

- √ Seguridad, que incluye aspectos de identificación de usuarios, control de accesos, confidencialidad de datos en las estaciones de trabajo, servidores y red.

- √ Medidas organizativas respecto a responsabilidad y propiedad de los datos en las estaciones de usuario y en los servidores.
- √ La necesaria normativa que propicie la integración de las aplicaciones de desarrollo propio con los estándares. Eso implica la definición de arquitecturas de aplicación y estándares, que deberían incluir normas sobre esa integración.
- √ Infraestructura de soporte necesaria para la gestión operativa.
- √ La necesidad de gestión y mantenimiento de aplicaciones en un sistema distribuido.

El desarrollo de aplicaciones Cliente/Servidor que ofrezcan una total flexibilidad en términos de función y de ubicación de datos requiere nuevos planteamientos y tiene implicaciones en el departamento de Sistemas de Información. Respecto a la transición de aplicaciones monolíticas a Cliente/Servidor, hay que tener en cuenta los siguientes puntos:

- √ La modelización de los procesos institucionales es clave para identificar las funciones de la aplicación que pueden implantarse y agruparse como servidores de aplicación (funciones de lógica de negocio encapsuladas).
- √ Las arquitecturas de aplicaciones y los principios de diseño deben establecerse antes de desarrollar la primera aplicación. La flexibilidad

y apertura de la aplicación Cliente/Servidor resultante, dependen de ello en gran medida.

- √ La ubicación de funciones y datos, así como la disponibilidad y el rendimiento, tendrán una dimensión diferente. Incluso en un entorno Cliente/Servidor flexible, en algún momento habrá que tomar decisiones relativas a la ubicación de funciones y datos.
- √ Las consideraciones sobre disponibilidad y rendimiento deben estar presentes en todas las etapas del proceso de desarrollo.

Pueden ser necesarios nuevos profesionales expertos en el desarrollo de interfaces gráficas de usuario, en el diseño y desarrollo de servidores de lógica de negocio y en la plataforma operativa. Esto puede exigir el uso de diferentes herramientas para desarrollar los distintos componentes de las aplicaciones.

La validación de las aplicaciones y su distribución a través de la red requiere nuevos procedimientos, ya que estamos frente a un conjunto compuesto por numerosos procesos cliente y servidor, que han sido probados individualmente.

Una de las responsabilidades de la organización de Sistemas de Información es la definición de una infraestructura Cliente/Servidor. No existen soluciones estándar Cliente/Servidor, aunque pueden existir bloques que se

puedan reutilizar. Ni siquiera se puede asegurar, de forma general, que un modelo de distribución, una herramienta o una plataforma, sean los mejores.

La solución correcta depende, en gran medida de:

- √ Los objetivos de negocio de la empresa.
- √ Los recursos disponibles actualmente de hardware, software y conocimientos.

El levantamiento de una infraestructura técnica que se aparte de las necesidades del negocio está condenado al fracaso. Los factores que determinan la solución Cliente/Servidor son:

- √ Las necesidades del negocio y cómo las tecnologías informáticas pueden soportarlas para conseguir los objetivos comerciales.
- √ El marco de tiempo disponible para la nueva puesta a punto.
- √ El estudio económico.
- √ El impacto sobre la organización y los conocimientos requeridos.
- √ Las arquitecturas y estándares adoptados por la compañía.

Pocas empresas pueden permitirse sustituir los sistemas y aplicaciones existentes en un solo paso. Normalmente es necesario un proceso gradual de implantación de las nuevas aplicaciones, lo cual exige:

- √ Una infraestructura Cliente/Servidor capaz de acomodar las aplicaciones existentes junto con las nuevas aplicaciones Cliente/Servidor.
- √ Estándares corporativos de tecnología informática.
- √ Una arquitectura de aplicaciones que permita desarrollar y poner en servicio nuevas aplicaciones, mientras siguen funcionando las existentes, preferiblemente sin tener que modificarlas.

Apéndice B. Programación en el Web

CSS

CCS o (Cascading Style Sheets) es la sumatoria de procedimientos que nos llevan a controlar distintos aspectos visuales y dinámicos de las páginas web, tales como fuentes, colores, márgenes y otros. Se constituye con un pequeño archivo que se vincula a una o más páginas, con el cual se consigue un aspecto similar para todas.

Style sheets contiene "reglas" para determinar la manera en que el estilo deseado debe ser aplicado a las páginas web. Para relacionar una página con un archivo CSS necesitamos en primer lugar un "selector", cada selector contiene "declaraciones" y las declaraciones tienen dos partes: propiedad y valor. CSS soporta 35 diferentes propiedades que pueden ser aplicadas a selectores a la hora de determinar la presentación de una página HTML.

Características y ventajas de las CSS

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a:

- Un web entero, de modo que se puede definir la forma de todo el web de una sola vez.

- Un documento HTML o página, se puede definir la forma, en un pequeño trozo de código en la cabecera, a toda la página.
- Una porción del documento, aplicando estilos visibles en un trozo de la página.
- Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación. Podemos definir, por ejemplo, varios tipos de párrafos: en rojo, en azul, con márgenes, sin ellos...
- Las hojas de estilo son compatibles con:

MS Internet Explorer 3 (parcialmente)

MS Internet Explorer 4

MS Internet Explorer 5

Netscape navigator 4 (parcialmente)

La potencia de la tecnología salta a la vista. Pero no solo se queda aquí, ya que además esta sintaxis CSS permite aplicar al documento formato de modo mucho más exacto. Si antes el HTML se nos quedaba corto para maquetar las páginas y teníamos que utilizar trucos para conseguir nuestros efectos, ahora tenemos muchas más herramientas que nos permiten definir esta forma:

- Podemos definir la distancia entre líneas del documento.
- Se puede aplicar identado a las primeras líneas del párrafo.
- Podemos colocar elementos en la página con mayor precisión, y sin lugar a errores.
- Y mucho más, como definir la visibilidad de los elementos, márgenes, subrayados, tachados...

Y seguimos mostrando ventajas, ya que si con el HTML tan sólo podíamos definir atributos en las páginas con píxeles y porcentajes, ahora podemos definir utilizando muchas más unidades como:

- Píxel (px) y porcentaje (%), como antes.
- Pulgadas (in)
- Puntos (pt)
- Centímetros (cm)

Las propiedades de las CSS incluyen "background", "font-size", "font-weight", "line-height", etc. Las reglas, entonces, están compuestas por la sumatoria de un selector y una declaración. Una regla puede usualmente presentarse de esta manera:

```
selector {propiedad: valor; propiedad: valor; ...}
```

Veamos un ejemplo para conseguir un párrafo pintado en verde:

`P {color:green}`

P - Es el selector y la regla (encerrada entre `{}` tiene una declaración)

color - Es la propiedad a la que se le va asignar un valor

green - Es el valor asignado; junto con la anterior propiedad constituye la declaración

Si hay propiedades múltiples, hay que separarlas en `{}`. Por ejemplo, el siguiente estilo definido asigna a la etiqueta `<h1>` una fuente específica (20 puntos) y un tipo de fuente (negrita):

`H1 {font-size: 15pt; font-weight: bold}`

Si queremos asignar las mismas propiedades, pero solo a dos o tres etiquetas:

`H1, H2, H3 {font-size: 15pt; font-weight: bold}`

Para que estos estilos sean aplicados hemos, por supuesto, de ubicar la regla en la sección `<head>` del documento, con lo que la haremos viable para aplicar al todo. Ahora bien, si por ejemplo, necesitamos que un determinado párrafo esté pintado de verde o que un determinado encabezado tenga tamaño 15 de fuente, necesitamos otro método.

HTML

HTML, HyperText Markup Language, es un lenguaje simple utilizado para crear documentos de hipertexto para WWW. No es un lenguaje de descripción de página como Postscript; HTML no permite definir de forma estricta la apariencia de una página, aunque una utilización algo desviada hace que se utilice en ocasiones como un lenguaje de presentación. Además, la presentación de la página es muy dependiente del browser (o programa navegador) utilizado: el mismo documento no produce el mismo resultado en la pantalla si se visualiza con un browser en modo línea, Mosaic o Netscape, o sea, HTML se limita a describir la estructura y el contenido de un documento, y no el formato de la página y su apariencia.

Una de las claves del éxito de WWW, aparte de lo atractivo de su presentación es sin duda, su organización y coherencia. Todos los documentos WWW comparten un mismo aspecto y una única interfaz, lo que facilita enormemente su manejo por parte de cualquier persona. Esto es posible porque el lenguaje HTML, en que están escritos los documentos, no solo permite establecer hiperenlaces entre diferentes documentos, sino que es un "lenguaje de descripción de página" independiente de la plataforma en que se utilice. Es decir un documento HTML contiene toda la información necesaria sobre su aspecto y su interacción con el usuario, y es luego el

browser que utilizemos el responsable de asegurar que el documento tenga un aspecto coherente, independientemente del tipo de estación de trabajo desde donde estemos efectuando la consulta.

Su simplicidad es tal que no es necesario utilizar un editor particular. Su gran permisividad exige rigor y atención en la estructura de documentos con el fin de que éstos se visualicen correctamente al margen del contexto y el browser utilizado.

Por tanto, HTML es un lenguaje muy sencillo que nos permite preparar documentos Web insertando en el texto de los mismos una serie de marcas (tags) que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos.

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.

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.

Para escribir HTML lo único que se necesita es un editor de texto ASCII, como EDIT del MS-DOS o el Bloc de notas de Windows. Las marcas o tags que controlan el comportamiento del documento son fragmentos de texto encerrados entre los signos "mayor que" y "menor que" (<marca>). Existen diferentes tipos de marcas: algunas controlan simplemente la presentación del texto del documento; otras, la forma en que se incluirán en él imágenes; otras, finalmente, los hiperenlaces con documentos o con diferentes partes

del mismo documento. Existen una serie de programas que ayudan en la elaboración de documentos HTML, como HTMLLED (shareware) o HTML Assistant, ambos para Windows, pero no son imprescindibles para escribir el código. Lo que si es necesario es un programa cliente WWW, tal como Mosaic, o Netscape, para probar el documento a medida que lo vamos desarrollando.

Las marcas funcionan muchas veces por parejas, una para indicar el inicio de enlace o formato, y otra para señalar el final. La marca de inicio consiste en una letra o una palabra (por ejemplo, estas son marcas de inicio: , <TITLE>). La marca de final es la misma letra o palabra precedida por la barra inclinada o "slash" (es decir, , </TITLE>). Existen, no obstante, algunas marcas que no requieren su pareja de cierre, como
 (que fuerza un salto de línea). Es importante señalar que las marcas, en general pueden estar indistintamente en mayúsculas o en minúsculas.

Apéndice C. COM/ DCOM/ COM+

COM

Características de COM

COM es un estándar binario que define como deben ser diseñados y construidos los objetos que cumplan su especificación. Provee la estructura necesaria para comunicarse con otras aplicaciones o componentes. Ya que COM es un estándar binario los objetos realizados mediante su especificación pueden comunicarse con otros objetos independientemente del lenguaje de desarrollo o del sistema operativo en el que se encuentren.

Independiente del lenguaje

Una aplicación cliente no debe preocuparse por el lenguaje de desarrollo del objeto COM que desea utilizar, solo debe comunicarse con él a través de una interface estándar. Este es uno de los motivos por los que la tecnología COM esta revolucionando el desarrollo de software, ya que no requiere cambios en el hardware, lenguajes de programación o arquitectura de redes (este último caso afecta a DCOM).

Comunicación entre aplicaciones

COM activa la interpelación entre aplicaciones que han sido escritas en diferentes lenguajes. Y que corran en diferentes sistemas operativos. Todos

los objetos soportan una interface estándar de manera que solo deban preocuparse por llamar o exponer sus interfaces.

Objetos COM

Los objetos COM son componentes de software reutilizables que están especialmente diseñados para ser usados por un número indeterminado de aplicaciones cliente. Al contrario de COM los objetos creados en lenguajes orientados a Objeto (como C++) pueden ser reutilizados solo por aplicaciones escritas en el mismo lenguaje. La tecnología puede encapsular los objetos OOP dentro de DLL, ActiveX(OCX.), o ejecutables para permitir el uso a diferentes aplicaciones escritas en lenguajes diferentes.

Debemos tener en cuenta que tanto OLE como ActiveX usan COM en su implementación.

Estructura de objetos e interacción

COM no especifica como los componentes están estructurados, solo especifica como interactúan entre ellos. El uso de un lenguaje de desarrollo u otro solo altera la arquitectura interna del objeto. COM hace posible que a través de las interfaces independientes del lenguaje los objetos se comuniquen entre sí.

Un objeto COM provee servicios a clientes. Un cliente no es necesariamente una aplicación, puede ser una pequeña porción de código, que accede a los

servicios del objeto. Estos servicios son accesibles a través de una interface COM que puede ser llamado por los clientes.

Existen dos tipos principales de objetos, in-process y out-of-process. El primero de ellos esta implementado como una DLL y siempre corre en la máquina local. El segundo está implementado como un ejecutable y puede residir en una máquina remota.

Interfaces COM

Como ya se ha dicho los objetos COM exponen su funcionalidad a través de una interface. La interface del objeto COM provee un puntero, o posición de memoria, a la función que el cliente quiere llamar. Esta interacción entre el cliente y el servidor se conoce como contrato. Los servicios del objeto COM se conocen como métodos o funcione miembro.

El contrato define los siguientes puntos:

- ✓ Se usa un identificador único para distinguir cada mensaje.
- ✓ La localización de la función en la interface.
- ✓ El propósito de la función.
- ✓ Los argumentos y valores de retorno de cada función.

Los siguientes puntos no están definidos en el contrato:

- ✓ Como se debe implementar la función.
- ✓ El código de la función.
- ✓ El lenguaje usado para crear el objeto.
- ✓ La plataforma en la que el objeto debe correr.

Una interface no es un objeto, ni una clase de objeto. Las interfaces son clases abstractas que se deben diseñar para que puedan ser rehusados en una variedad de contextos.

El interface IUnknown.

Un objeto COM puede tener más de una interface, cada uno de ellos puede proveer de una funcionalidad diferente para diferentes clientes.. Todos los objetos COM soportan una interface estándar llamado IUnknown. La interface IUnknown dispone de tres métodos: AddRef, Release y QueryInterface.

- ✓ AddRef
- ✓ Este método provee la actual creación de objetos COM. Cada instancia de un objeto COM usa el método para incrementar el contador de objetos. En Visual Basic el método AddRef se llama cuando se usa la función CreateObject.
- ✓ Release
- ✓ El método se usa cuando se destruye un objeto para decrementar el contador de objetos instanciados. En Visual Basic se llama cuando se igual a la variable que representa el objeto a Nothing.
- ✓ QueryInterface
- ✓ Este método enumera las interfaces que el objeto soporta. Visual Basic provee esta capacidad a través de la orden Set.

El interface Idispatch.

Es una interface que se usa para no acceder directamente a las interfaces COM, es decir provee de una manera de llamar a otra interface. Parecido a la función GetProcAddress para llamar a la función de una Dll. Es decir, se llama a una interface COM en dinámico. Soporta los siguientes métodos. El uso de esta interface se conoce como late-binding.

- ✓ GetIDsOfNames
- ✓ Busca en la interface indicado el nombre de una método, si lo encuentra facilita un puntero que podemos usar con una llamada a Invoke.
- ✓ GetTypeInfo
- ✓ Devuelve información de tipo para un objeto.
- ✓ GetTypeInfoCount
- ✓ Devuelve el número de interfaces que un objeto provee.
- ✓ Invoke
- ✓ Provee acceso a las propiedades y métodos expuestos por un objeto.

En run-time, los clientes utilizan una llamada a Idispatch.GetIDsOfNames para consultar si el método o propiedad al que quieren llamar existe. Si la propiedad o el método existen en el objeto COM se devuelve el correspondiente ID. El ID puede ser usado para llamar al método o

propiedad mediante el `Idispatch.Invoke`. Las funciones `GetTypeInfoCount` y `GetTypeInfo` obtienen información del componente acerca de interfaces, métodos y propiedades.

Virtual Tables.

Automation permite a un cliente ActiveX llamar a un método o propiedad directamente. Esta característica, llamada virtual table (VTBL) binding, no hace uso del interface `Idispatch`. El uso de la VTBL se conoce como early binding. Esta técnica requiere que una librería provea información acerca de los tipos del objeto. El cliente obtiene información desde la librería de tipos en tiempo de compilación y entonces puede llamar a los métodos y funciones directamente. Esto permite a Visual Basic realizar chequeo de la sintaxis en tiempo de compilación. En run-time, este tipo de enlace es más rápido, ya que la llamada no debe pasar por el interface `Idispatch`.

Los clientes usan punteros a instancias de interfaces. Un puntero a interface apunta a un array de punteros a funciones conocidas como VTBL. Las funciones a las que apunta son conocidas como métodos o funciones miembro de interface. Por convención los nombres de las interfaces empiezan por "I".

En Visual Basic usar una técnica u otra es muy parecido, pero puede representar diferencias de rendimiento enormes.

Código de Visual Basic que usa Late binding:

```
Dim miObjetoMio As Object  
Set miObjetoMio = CreateObject("ElObjetoCOM,ElInterface")  
MiObjetoMio.ElMetodoDelInterface()
```

Código de Visual Basic que usa early binding

```
Dim miObjetoMio As Clase.Objeto  
Set miObjetoMio = CreateObject("ElObjetoCOM.ElInterface")  
MiObjetoMio.ElMetodoDelInterface()
```

En la VTBL encontramos la lista de direcciones de todos los métodos y propiedades que son miembros de un objeto, incluyendo la función miembro de las interfaces que soporta. Los tres primeros elementos de la VTBL son los ya conocidos miembros del interface IUnknown. Las siguientes entradas son miembros de los otros interfaces soportados. OLE provee un número de interfaces que pueden ser usados y que generalmente empiezan por IOle,

DCOM (Distributed Component Object Model)

Arquitectura DCOM

En los sistemas operativos de hoy en día los procesos están aislados unos de otros. Un cliente que necesite comunicarse con un componente de otro proceso no puede realizar una llamada directa, sino que tiene que hacer uso de algún tipo de comunicación entre procesos proporcionada por el sistema operativo. COM/DCOM permite establecer esta comunicación de una manera completamente transparente interceptando la llamada del cliente y dirigiéndola a un componente en otro proceso.

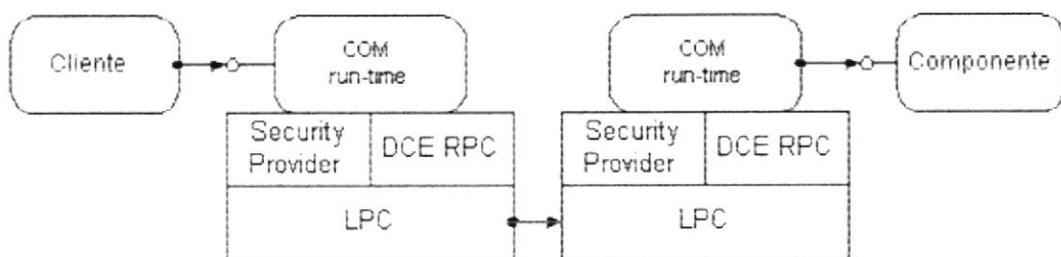


Figura 4-1. Componentes COM en diferentes procesos

Cuando cliente y componente residen en diferentes máquinas, DCOM simplemente sustituye la comunicación entre procesos por un protocolo de red, de modo que ninguno de los dos advierte este cambio. COM proporciona servicios orientados a objetos a clientes y componentes, y utiliza DCE RPC y varios sistemas de seguridad (Security Providers) para generar

paquetes de red. DCE RPC, del que DCOM toma el concepto de GUID (Globally Unique Identifier), define un estándar para la conversión de parámetros y estructuras de datos almacenadas en memoria para formar paquetes de red. Su formato de representación de datos NDR (Network Data Representation) es independiente de la plataforma.

Independencia de localización

Al implementar una aplicación distribuida real sobre una red pueden aparecer una serie de restricciones al diseño:

Algunos componentes sólo pueden ser ejecutados en algunas máquinas o en determinadas localizaciones.

Los componentes que interactúan más a menudo deberían estar situados "más cerca".

Muchos componentes pequeños incrementan la flexibilidad del diseño, pero generan mayor tráfico en la red.

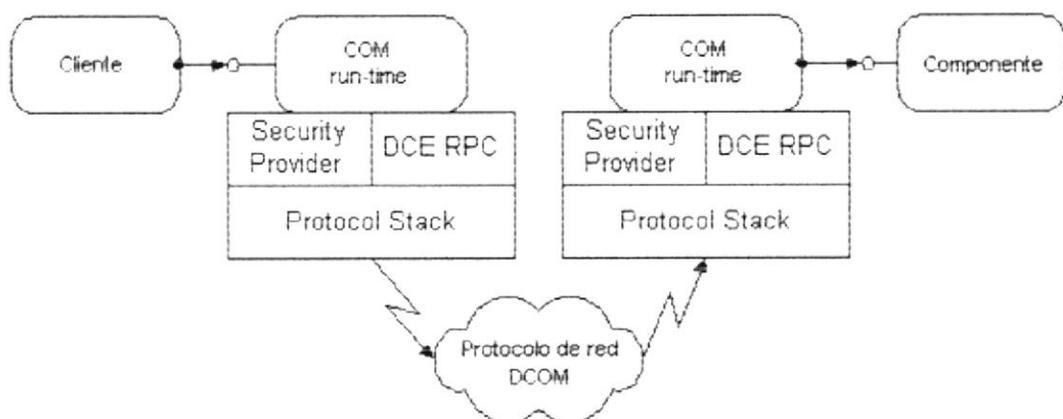


Figura 4-2. Componentes COM en diferentes máquinas

Pocos componentes grandes reducen el tráfico en la red pero también la flexibilidad del diseño.

Con DCOM es fácil evitar estas restricciones, ya que oculta la localización de los componentes, y la forma en la que el cliente se conecta a ellos y realiza una llamada a sus funciones es siempre la misma. DCOM no sólo no requiere cambios en los programas fuente, sino que tampoco es necesario precompilarlos, porque la reconfiguración cambia la forma en que los componentes se conectan entre ellos. Esto simplifica en gran medida la tarea de distribuir los diferentes componentes de la aplicación con el fin de optimizar el rendimiento.

En la figura se muestra como un mismo componente puede situarse en la máquina del cliente cuando existe suficiente capacidad en la red o en el servidor cuando el cliente accede a la aplicación a través de un enlace lento.

Facilidad de crecimiento (scalability)

Un factor crítico de las aplicaciones distribuidas es su capacidad de adaptarse al aumento del número de usuarios, volumen de datos o funcionalidades requeridas. La aplicación debería ser pequeña y rápida cuando la demanda es baja, pero también ser capaz de atender una demanda creciente manteniendo un rendimiento y una fiabilidad aceptables. DCOM proporciona algunos elementos que facilitan la capacidad de crecimiento de las aplicaciones.

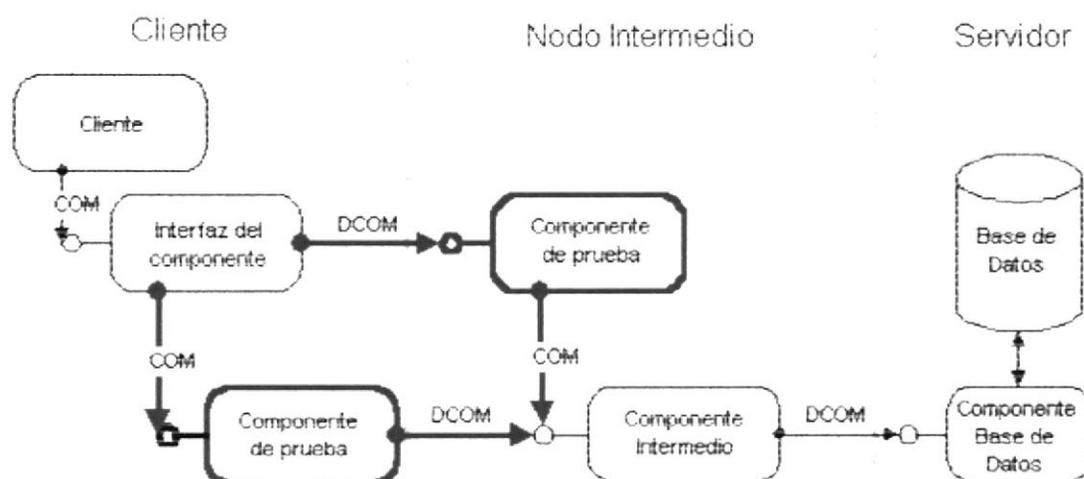


Figura 4-3. Independencia de la localización

Multiproceso simétrico

DCOM aprovecha la capacidad de proceso de Windows NT. En aquellas aplicaciones que utilizan un modelo de hebras libres (free-threads), DCOM usa un banco de hebras (thread pool) para las peticiones que llegan al sistema. En máquinas con varios procesadores, este banco de hebras se optimiza en función del número de procesadores disponibles: demasiadas hebras suponen cambios de contexto frecuentes, mientras que pocas hebras pueden desaprovechar la capacidad de algunos procesadores que quedarían inactivos. DCOM libera al diseñador de los detalles del manejo de

hebras, consiguiendo un rendimiento que sólo podría obtener programando todo el proceso manualmente.

Desarrollo flexible.

En ocasiones no es posible atender a una demanda creciente ni siquiera aumentando la velocidad que proporciona una máquina multiprocesador. La independencia de la localización de DCOM facilita la labor de distribuir componentes entre diversas máquinas, ofreciendo un modo más sencillo y barato de aumentar la capacidad de las aplicaciones. La redistribución es más sencilla cuando los componentes son "sin estado" (stateless) o no comparten su estado con otros componentes, ya que en este caso es posible ejecutar varias copias en diferentes máquinas.

La demanda puede ser distribuida de forma uniforme entre todas las máquinas o de acuerdo a criterios como capacidad y carga. Cambiar la forma en la que los clientes se conectan a los componentes y éstos últimos entre sí es fácil con DCOM, porque permite, sin necesidad de añadir código o recompilar, la redistribución dinámica de los componentes. Sólo es necesario actualizar el registro, archivo del sistema, o base de datos en la cual se almacena la situación de cada componente en un momento determinado.

Muchas aplicaciones reales tienen uno o más componentes críticos (bottleneck components) que son utilizados en la mayor parte de las operaciones. Pueden ser, por ejemplo, componentes de bases de datos a los que hay que acceder en serie de modo FCFS (first come-first served). No es posible duplicar este tipo de componentes, ya que su finalidad es precisamente establecer un punto único de sincronización para todos los usuarios de la aplicación.

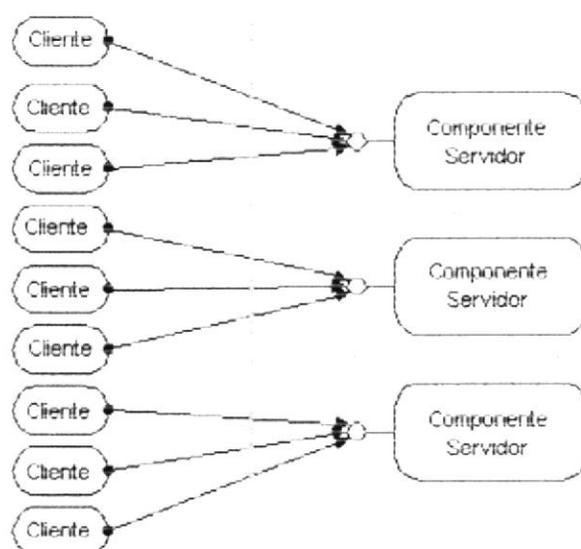


Figura 9. Distribución paralela de componentes.

Con el objetivo de mejorar el rendimiento de una aplicación distribuida estos componentes críticos deberían ejecutarse en un servidor dedicado de gran potencia y altas prestaciones. De nuevo DCOM ayuda a aislar estos

componentes críticos en las fases iniciales del proceso de diseño, colocando inicialmente todos ellos en una misma máquina para más tarde situar los componentes críticos en máquinas dedicadas.

Los componentes críticos suelen formar parte de una cadena de procesos, por ejemplo, órdenes de compra o venta en un sistema de venta electrónico: los pedidos deben ser procesados conforme se van recibiendo y en ese orden (FCFS). Una posible solución sería dividir el trabajo en componentes más pequeños y ejecutar cada uno de ellos en una máquina diferente. El efecto sería similar al que se consigue con la técnica del *pipelining* en los microprocesadores modernos: cuando llega la primera petición es procesada por el primer componente (que podría ser, por ejemplo, una comprobación de la corrección semántica de la orden) y en caso de que sea válida pasa al siguiente (para realizar, por ejemplo, una actualización de la base de datos).

Tan pronto como el primer componente mande al segundo la petición, quedará libre para procesar nuevas entradas y de esta forma tendremos dos máquinas trabajando en paralelo sobre dos peticiones

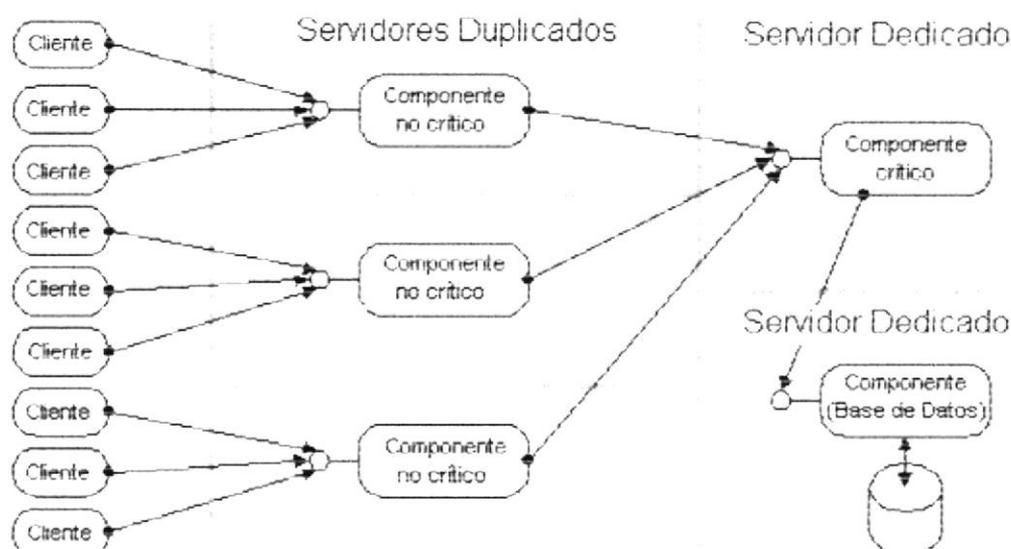


Figura 4-5. Distribución de un componente crítico.

Diferentes, a la vez que garantizamos que éstas son procesadas en el orden adecuado. Con DCOM es posible aplicar esta misma técnica cuando trabajamos en una sola máquina, ya que podemos ejecutar varios componentes en diferentes hebras o en diferentes procesos, facilitando además el crecimiento de nuestra aplicación mediante la distribución de las hebras en una máquina multiproceso o de los procesos en diferentes máquinas.

Control de la conexión

Las conexiones a través de la red son más delicadas que las conexiones dentro de una máquina, por esta razón es necesario que los componentes de una aplicación distribuida sean avisados cuando un cliente deja de estar

activo o cuando se produce un fallo en la red o en el hardware. DCOM controla las conexiones entre componentes que están dedicados a un sólo cliente así como las de aquellos que están compartidos por varios clientes. Cuando un cliente establece una conexión con un componente el contador de referencias se incrementa en una unidad, decrementándose cuando el componente libera a la conexión. En caso de que el contador sea cero el componente puede liberar la memoria que ocupa.

DCOM utiliza un protocolo de ping en el cual el cliente envía de forma periódica un mensaje para comprobar que los componentes permanecen activos. Si transcurren más de tres periodos sin que el componente reciba estos mensajes DCOM considera rota la conexión y procederá a decrementar el contador de referencias, liberando el componente si el contador llega a cero. Desde el punto de vista del componente, el proceso es el mismo independientemente de que sea el cliente el que se desconecte por sí mismo o de que se produzca un fallo en la red o en la máquina cliente. Todo este mecanismo de liberación de memoria por parte de componentes no utilizados es transparente a las aplicaciones clientes.

Ancho de banda y retardo

Aunque en teoría DCOM oculta el hecho de que en las aplicaciones distribuidas los componentes se ejecutan en diferentes máquinas, en la práctica es necesario considerar las limitaciones de una conexión a través de una red:

Ancho de banda: el tamaño de los parámetros en una llamada a una función afecta directamente al tiempo necesario para completar esa llamada.

Retardo: La distancia física y el número de elementos de la red, tales como routers o enlaces, introducen un retardo que puede llegar a ser significativo.

En el caso de una red global como Internet los retardos pueden ser del orden de segundos.

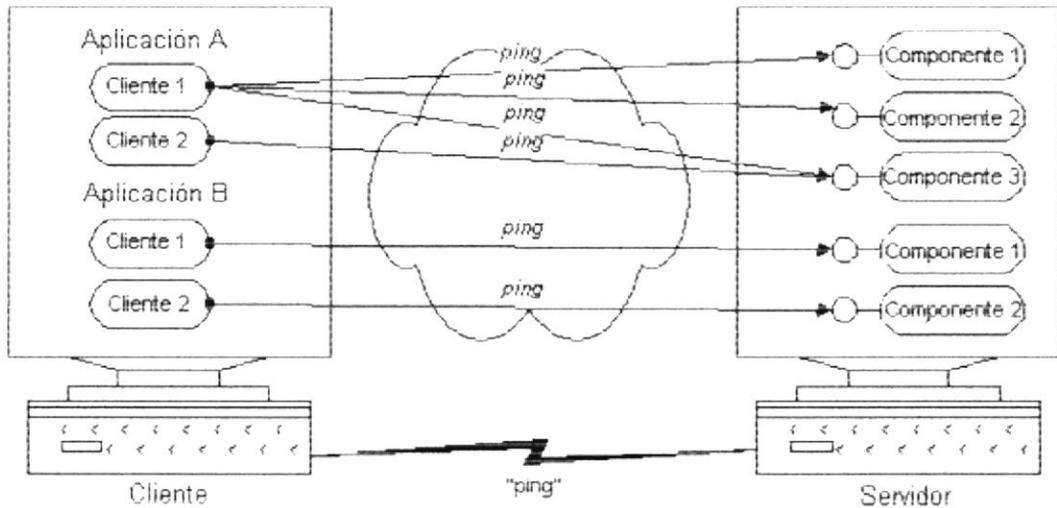


Figura 4-6. Uso de ping para controlar las conexiones.

DCOM consigue superar estas limitaciones minimizando cuando sea posible el número de saltos en la red para evitar retardos. El protocolo de transporte más usado por DCOM es el protocolo no orientado a conexión UDP, lo que permite una mayor optimización mezclando paquetes de asentimiento con datos y mensajes ping. Incluso en el caso de que DCOM emplee protocolos

orientados a conexión se consigue una significativa mejora con respecto a protocolos desarrollados específicamente para cada aplicación.

Seguridad

Al utilizar una red de comunicaciones para distribuir una aplicación no sólo nos encontramos con limitaciones físicas, ancho de banda y retardo, sino también con problemas de seguridad entre clientes y componentes. Al ser las funciones accesibles físicamente por cualquiera con acceso a la red, es necesario establecer restricciones en un nivel superior. DCOM implementa su propio sistema de seguridad con el fin de evitar que cada aplicación se vea obligada a desarrollar el suyo de manera independiente.

Una plataforma distribuida como DCOM debe facilitar un entorno seguro que permita distinguir entre clientes y grupos de clientes de modo que el sistema o la aplicación pueda conocer quién intenta acceder a determinados servicios de un componente. DCOM utiliza el sistema de seguridad de Windows NT que consiste en una serie de security providers basados en métodos tradicionales tales como dominios o claves públicas en sistemas no centralizados. Una de las partes fundamentales de este sistema de seguridad es el directorio de usuarios que almacena la información necesaria para comprobar las credenciales de éstos.

DCOM proporciona seguridad a las aplicaciones distribuidas sin necesidad de incluir en el cliente o en el componente elementos de codificación o diseño específicos. Al igual que ocurrían con la localización de los

componentes DCOM también oculta sus necesidades de seguridad. El mismo código ejecutable en un entorno centralizado, con una sola máquina, donde la seguridad no plantea ningún problema puede ser utilizado en un entorno distribuido de manera segura. El diseñador simplemente tiene que fijar los requisitos de seguridad de cada componente, indicando que usuarios tienen derecho de acceso a sus servicios. En el que caso de que el diseñador se desentienda de los aspectos de seguridad, DCOM proporciona un mecanismo de seguridad por defecto bastante eficiente.

Seguridad en Internet

Existen dos problemas de seguridad fundamentales a la hora de utilizar aplicaciones diseñadas para trabajar en Internet.

El número de usuarios puede ser varios órdenes de magnitud superior al existente en una compañía.

Los usuarios finales desearían poder usar la misma clave o password para todas las aplicaciones que utilizan, aunque sean distribuidas por diferentes compañías. Sin embargo, la aplicación o el sistema de seguridad del proveedor de los servicios puede no tener capacidad para almacenar las claves privadas de todos los posibles usuarios.

Las aplicaciones distribuidas que utilizan la arquitectura DCOM pueden aprovechar los sistemas de seguridad que ésta proporciona sin necesidad

de introducir ningún cambio en dichas aplicaciones, incluso en las más sensibles a los problemas de seguridad.

DCOM hace uso de los sistemas de seguridad (security providers) que proporciona el entorno Windows NT:

Protocolo de autenticación NTLM, empleado por la versión 4.0 y anterior de Windows NT.

Protocolo de autenticación Kerberos Version 5, que sustituye a NTLM para el acceso a recursos dentro o entre dominios de Windows NT.

Protocolo de autenticación distribuido DPA (Distributed Password Authentication), empleado por algunas de las principales organizaciones de Internet, como CompuServe o Microsoft Network.

Servicios de seguridad por canales seguros, que implementan los protocolos SSL/PCT en Windows NT 4.0.

Sistema de seguridad que utiliza DCE como entidad expendedora de claves sobre Windows NT.

Todos estos sistemas de seguridad utilizan protocolos estándar de Internet y tienen diferentes ventajas e inconvenientes. Los protocolos NTLM y Kerberos son protocolos de clave privada, son extremadamente eficientes y seguros en entornos centralizados o en dominios basados en servidores Windows NT. Existen versiones comerciales de NTLM para la mayor parte de los entornos UNIX.

Con el sistema de directorios de Windows NT 4.0, dominios controlados por varios servidores de claves pueden tener un buen comportamiento hasta aproximadamente 100.000 usuarios. Con el sistema ampliado de directorios de Windows NT 5.0 un solo servidor de claves en un dominio Windows NT puede llegar a administrar hasta un millón de usuarios. Combinando varios servidores de claves en el árbol de directorios de Windows NT 5.0, el número de usuarios en un sólo dominio es prácticamente ilimitado.

El sistema Kerberos sobre Windows NT 5.0 proporciona elementos de seguridad más avanzados como controlar qué pueden hacer los componentes mientras autentifica a los clientes, además de requerir menos recursos que NTLM.

Microsoft planea incluir en el futuro sistemas de seguridad basados en claves públicas para Windows NT, que permitirá descentralizar el control de la seguridad para cualquier aplicación de Windows NT, incluyendo aquellas basadas en DCOM. La autenticación mediante claves públicas es menos eficiente que usando claves privadas, pero no requiere almacenar las credenciales privadas de los clientes.

Rendimiento y prestaciones

Como hemos visto hasta ahora la arquitectura DCOM ofrece herramientas que proporcionan flexibilidad tanto para la integración de componentes COM ya desarrollados, como para la utilización de diferentes lenguajes de

programación, además de permitir el crecimiento de las aplicaciones. Sin embargo, podemos preguntarnos cómo afectan estas características al rendimiento de la arquitectura. En COM/DCOM el cliente nunca tiene acceso a la implementación del objeto servidor, lo que se consigue, en la mayor parte de los casos, sin tener que utilizar componentes intermedios. Esta transparencia de acceso se obtiene mediante un mecanismo de comunicación entre componentes a través de llamadas a función mediante punteros. La única sobrecarga que introduce este método frente a los utilizados en otros lenguajes de programación es la búsqueda de la dirección de función en las tablas virtuales (vtables).

Tanto si las llamadas a función se realizan entre procesos situados en la misma máquina o a través de la red, DCOM utiliza su propio mecanismo de RPCs orientado a objetos. Microsoft publica una serie de cifras que miden el rendimiento objetivo de este sistema: indican una mejora de aproximadamente el 35% sobre TCP/IP para RPCs vacías, aunque para RPCs reales este porcentaje desciende hasta el 15%.

Parameter Size	4 bytes		50 bytes	
	calls / sec	ms / call	calls / sec	ms / call
"Pentium [®] ," in-process	3,224,816	0.00031	3,277,973	0.00031
"Alpha [™] ," in-process	2,801,630	0.00036	2,834,269	0.00035
"Pentium," cross-process	2,377	0.42	2,023	0.49
"Alpha," cross-process	1,925	0.52	1,634	0.61
"Alpha," to Pentium remote	376	2.7	306	3.27

Tabla 4-1. Tiempo necesario para realizar llamadas a función. Fuente: Microsoft Corp.

COM+

Terminología y Conceptos

Si ha usado MTS en alguna ocasión, sabrá que los componentes se agrupan formando paquetes, que son la entidad mínima e instalación y configuración. Un paquete puede contener múltiples componentes COM, que comparten muchos de sus atributos y, por regla general, se ejecutan en un mismo proceso.

En COM+ el término paquete ha sido sustituido por aplicación. Una aplicación COM+ es por tanto, un grupo lógico formado por uno o más componentes y todos sus atributos se almacenarán separadamente, en una base de datos optimizada para trabajar con COM+.

Los componentes COM+ siempre estará alojado en el interior de servidores dentro de proceso. Dicho en otras palabras los componentes deben formar parte de una librería de enlaces dinámico, no de un ejecutable. Esto no implica no obstante que todos los componentes vayan a ejecutarse en el mismo espacio de proceso que el cliente. De hecho, en la mayoría de las ocasiones esto no será así, puesto que compromete la estabilidad y limita las posibilidades.

Las aplicaciones COM+ pueden ser de dos tipos: de biblioteca o de servidor. Las aplicaciones de biblioteca son equivalentes a los servidores dentro de proceso, en el sentido de que ejecutan localmente y utilizando el mismo espacio de direcciones que el proceso del cliente. Las aplicaciones de servidor, que en teoría será el tipo preferente, se ejecutan en un proceso subrogado de COM+, en lugar de hacerlo en el proceso cliente. Esto contribuirá a la estabilidad de las aplicaciones, al tiempo que aportara otras posibilidades como la ejecución remota.

Cada una de las aplicaciones COM+ puede contener, como se ha dicho antes, uno o varios componentes. Estos, a su vez, dispondrán de atributos e interfaces. Las interfaces, por último estarán compuestas de atributos y métodos. Estos últimos también dispondrán de atributos específicos.

Los componentes de una aplicación COM+ se ejecutan en un cierto contexto, definido por todos los atributos y representando por un objeto COM conocido habitualmente como objeto contexto. Dicho objeto cuenta con interfaces que ofrecen servicios diversos al componente, de tal manera que el programador puede gestionar explícitamente, aunque no sea siempre preciso, aspectos como la seguridad o las transacciones

El Catalogo COM+

Hasta la aparición de Windows 2000, el único soporte de almacenamiento para los atributos de los componentes COM era el registro de configuraciones. En esta base de datos se almacenaba, básicamente, utilizaba. No había mucho mas que almacenar, puesto que los componentes COM no contaban con mas atributos que los mencionados

En COM+ se utilizan de manera importante los atributos declarativos propiedades que definen como se comporta un componente y condicionan su uso y funcionamiento, Dichos atributos, lógicamente, deben almacenarse en algún soporte físico y persistente. Puesto que el registro de configuraciones de Windows no fue creado con el propósito de almacenar estos atributos, en Windows 2000 existe una nueva base de datos, exclusiva para los componentes COM, conocida como catalogo COM+.

En el pasado, la mayoría de las características funcionales de un componente COM debían implementarse mediante código siendo una responsabilidad plena del programador. Con COM+ se produce un cambio importante: muchas de esas características se definen mediante atributos, sin necesidad de alterar para nada el código del componente. Un mismo componente por poner un ejemplo; puede ser usado de manera sincrónica, como es tradicional, o asincrónica.

Seguridad basada en perfiles

El nivel de seguridad en los componentes es muy superior al existente para COM y, además sin necesidad de que el programador del componente se preocupe por este aspecto. Cada aplicación COM+ tendrá definido uno o más perfiles, a los que se sumaran los usuarios que sean pertinentes. Suponiendo que una aplicación COM+ este compuesta por componentes para efectuar transacciones bancarias, los posibles perfiles podrían ser administrador, interventor, cajero, etc

Los perfiles se utilizan para permitir o denegar el acceso a los componentes, a algunas de sus interfaces o, incluso a los métodos que implementan. De esta forma, es posible que dos usuarios distintos usen el mismo componente pero con limitaciones diferentes, teniendo uno acceso a toda la funcionalidad y otro solo a una parte, ya sea limitando que puede solicitar a los métodos que puede ejecutar.

Si se desea, es posible incluso afinar aun más en la seguridad de acceso a los servicios que ofrecen los componentes. Mediante las interfaces específicas del contexto del componente, el programador puede comprobar si el cliente que esta accediendo al servicio pertenece o no a un cierto perfil, adaptando su funcionamiento según este producto.

En caso de que el usuario, que actúa como cliente respecto al componente tenga que acceder a otros recursos, tales como bases de datos, COM+ también cuenta con servicios de delegación de la identificación. De esta forma, es el mismo servidor de aplicaciones el que se encarga de representar al usuario frente al servidor de datos por mencionar el caso más usual.

Transacciones

De entre los servicios heredados por COM+ de MTS, seguramente él mas conocido es el control de transacciones. La posibilidad de gestionar las transacciones al nivel de componentes en el servidor de aplicaciones, permite efectuar operaciones que de otra forma, resultarían mucho más complejas.

La mayoría de los programadores asocian el uso de transacciones al trabajo con bases de datos, al ser estas uno de los recursos en los que es indispensable asegurar la integridad de la información. Las transacciones COM+ se sitúan en un nivel superior, pudiendo abarcar operaciones no solo sobre una determinada base de datos, como es habitual, sino entre varias bases de datos que, incluso, podrían estar en diferentes servidores físicos.

El control de las transacciones en MTS ya era bastante simple. Básicamente se trata de indicar que tipo de servicios de transacciones precisa un cierto

componente el cual, cuando sea preciso, indicara al sistema la finalización o interrupción, según los casos.

En COM+ se ha añadido una posibilidad: la terminación automática de una transacción. Esto puede ocurrir cuando se produce una excepción en el componente, caso en el cual la transacción se cancelaría o bien cuando el componente finaliza su trabajo, haciendo firmes las operaciones efectuadas durante la transacción. Para permitir que una transacción pueda tener lugar en un entorno distribuido, COM+ se sirve del Coordinador de transacciones distribuidas, conocido como MS DTC(Distributed Transaction Coordinator), y de unos gestores de recursos, que son los que realmente acceden a la información que se compartirá entre los distintos clientes

Balanceo de cargas

Uno de los problemas que presentan habitualmente los servidores de aplicaciones, al igual que los servidores de datos, es la escalabilidad. En un momento u otro, el número de usuarios que requieren el acceso a un componente es tan importante que el rendimiento cae, hasta llegar a tiempos de respuesta inaceptables.

Las versiones superiores de Windows 2000, concretamente las ediciones Advanced Server y DataCenter Server, incluyen un servicio de balanceo o

equilibrio de cargas para los componentes. Dicho servicio permite crear un grupo o cluster de servidores de aplicaciones que, de manera lógica, forman un solo servidor. De este grupo, una de las máquinas actúa como equilibrador de cargas, analizando cuál es el tiempo de respuesta de cada servidor y derivando la creación de componentes, a medida que el cliente lo demande, al servidor que este menos ocupado.

Que un determinado componente COM+ pueda o no ser utilizado en un entorno de equilibrio de cargas es algo que depende tan solo de un atributo que, como todos los demás es almacenado en el Catálogo COM+. Aunque el programador al crear el componente, no tiene que hacer nada especial si que debe tener en cuenta que el componente no podrá asumir que esta ejecutándose en una cierta máquina ya que, en la práctica, el entorno de ejecución dependerá de la carga que exista.

La posibilidad de crear un cluster de servidores de aplicaciones con el fin de incrementar la estabilidad y mejorar el tiempo de respuesta global también influye decisivamente en la disponibilidad de los servicios, parámetro no menos importante que la estabilidad.. En caso de que uno de los servidores de aplicaciones caiga por alguna causa, el servidor de balanceo de cargas repartirá el trabajo entre los que quedan en servicio, de tal forma que los clientes podrán seguir trabajando sin problema.

Componentes de funcionamiento asincrónico

Otra novedad importante de COM+ es la posibilidad de utilizar componentes de manera asincrónica, de tal forma que cliente y servidor no tienen por que ejecutarse en el mismo espacio de tiempo, algo que es habitual, sobre todo, en entorno móvil. El funcionamiento de los componentes asincrónico esta basado en los servicios de MSMQ(Microsoft Message Quering Services), un servidor que deberá estar funcionando en la red y que incluye en WINDOWS 2000.

Los servicios que contienen componentes asincronos, las aplicaciones COM, no se diferencian de cualquier otro servidor salvo en algunos atributos, mediante los cuales se indicaran que el componente puede ser iniciado a través de una cola de mensajes MSMQ y que los métodos pueden ser invocados a través de ese sistema. El programador del componente por tanto y una vez mas, no tiene por que preocuparse de este aspecto.

La creación de los clientes, sin embargo, si implica algunos cambios de filosofía, aunque realmente afectan tan solo a la creación del componente, porque este no se encuentra en ejecución o esta en una red a la que no hay acceso en ese momento, lo que obtiene a cambio es una referencia a un objeto que actúa como una grabadora. La grabadora, que no es mas que un objeto COM que muestra las mismas interfaces que el componente que se

quiere usar, se encarga de registrar todas las operaciones y los bien parámetros asociados. Cuando el cliente se conecta a la red en el que esta ejecutándose el componente, o bien cuando este inicia su ejecución si es que estaba detenido, otro objeto, actuando como reproductor, se encargara de tomar el registro y realizar todas las operaciones detalladas.

Otros servicios de COM+

Además de los ya mencionados COM+ cuenta con otros servicios también destacables. Al igual que MTS. COM+ cuenta con un deposito de objetos reutilizables que el administrador del sistema puede configurar según las necesidades. Los programadores también deben seguir unas ciertas normas al crear componentes que puedan reutilizarse, de tal forma que estos puedan quedar en el mencionado deposito inactivos, conocida como JIT(Just In-Time), tiene la ventaja de que lo que se usa es un componente ya existente, no hay necesidad de crear un nuevo componente y por tanto, el tiempo de respuesta es menor, mejorando el rendimiento.

Tambien hay que mencionar la existencia de un nuevo modelo de apartamento para los componentes conocido como NTA(Neutral Thread Apartment) Este viene a minimizar los problemas que se presentan cuando un componente diseñado para ejecutarse en un STA(Single Tread Apartment) es utilizado con una aplicación con múltiples hilote ejecución. Lo cual implica cambios de contexto que consumen tiempo y afectan negativamente

al rendimiento. Con el nuevo NTA este problema se minimiza al no ser preciso un cambio de contexto.

Utilizar COM+

Las tecnologías COM+ y Microsoft Windows® 2000 proporcionan una sólida plataforma de desarrollo. Esta plataforma se compone de diversas tecnologías esenciales que proporcionan los componentes básicos para generar aplicaciones empresariales de varios niveles.

COM+ ofrece a las empresas las ventajas de las aplicaciones de varios niveles y reduce su complejidad inherente.

En este artículo vamos a ver un análisis general de los servicios distribuidos que se han integrado en la plataforma COM+. Cualquier aplicación de varios niveles que pueda parecer irrelevante requiere elementos como un soporte para las transacciones, seguridad integrada, un servidor Web, un sistema de mensajería y una entrega de notificaciones de eventos. En este artículo se identificarán dónde se ubican cada uno de estos servicios COM+. De este modo, obtendremos una visión general de estas nuevas tecnologías.

Necesidad de mensajería

Un gran número de protocolos que se utilizan en las aplicaciones distribuidas, como RPC y HTTP, son sincrónicos, es decir, que las aplicaciones que se ejecutan en el equipo cliente deben esperar pacientemente durante cada petición mientras que el equipo servidor realiza

la tarea solicitada. Asimismo, si el servidor tiene numerosas peticiones atrasadas, el cliente no tiene otra opción que enfrentarse a un largo tiempo de espera.

Las aplicaciones creadas mediante protocolos sincrónicos dependen de todos los equipos que están en línea y que forman parte del procesamiento de una petición de forma simultánea. El servidor debe estar en línea para que el cliente realice sus peticiones y, de igual forma, los clientes deben estar en línea enviando peticiones para que se utilicen los ciclos de procesamiento disponibles que ofrece el servidor. Si se desconecta el servidor, los clientes no podrán enviar sus peticiones. Cuando se desconectan los clientes, el servidor permanece expectante sin realizar ninguna actividad. En resumen, todo el mundo debe estar conectado al mismo tiempo para que el sistema en general pueda funcionar.

Suele ocurrir que varias aplicaciones necesiten comunicarse sin conexión y de forma asincrónica. La mensajería es el mecanismo que permite establecer una comunicación asincrónica entre dos o más aplicaciones. Las aplicaciones cliente envían mensajes asincrónicos a las colas de la red.

Generalmente cada mensaje representa una petición del cliente o algún tipo de notificación. Las aplicaciones del servidor supervisan estas colas y procesan los mensajes en el momento en que se reciben o con

posterioridad. Los mensajes y las colas proporcionan, simplemente, una capa de indirección entre las aplicaciones que realizan las peticiones y las que las procesan.

Las aplicaciones que se crean sobre un protocolo de mensajería pueden continuar funcionando cuando varios equipos están desconectados. No obstante, no es tarea fácil la creación de una infraestructura sólida de mensajería que pueda enviar cada mensaje hacia su cola de destino con las garantías de envío necesarias. La infraestructura de mensajería debe proporcionar un amplio conjunto de subsistemas a fin de almacenar mensajes de forma transparente cuando los equipos estén fuera de línea y enviarlos con posterioridad a sus destinos cuando se pueda establecer una conexión.

La mayoría de las empresas no están dispuestas a invertir el tiempo y dinero necesarios para escribir y depurar el código de una infraestructura de mensajería personalizada. Afortunadamente, se encuentran a la venta varios productos de mensajería que ofrecen las ventajas de la comunicación asincrónica y sin conexión, como es el caso de MQSeries de IBM y MSMQ Server de Microsoft. Gracias a ello no es necesaria una infraestructura de mensajería personalizada.

Descripción de los servicios COM+

Además de proporcionar un entorno en tiempo de ejecución, COM+ y Windows 2000 también incluyen varios servicios integrados de gran importancia para los programadores que generan aplicaciones de varios niveles. Puede que algunas aplicaciones de varios niveles necesiten aplicar uno o dos de estos servicios. Otros, sin embargo, necesitarán la utilización de todos ellos. Resulta conveniente tener una perspectiva global de la interrelación entre estos componentes para poder tomar las decisiones adecuadas durante la fase inicial de diseño. En las siguientes secciones se incluye una descripción general sobre los servicios más importantes de la plataforma para las aplicaciones distribuidas.

Servicios de Internet Information Server

Servicio de Microsoft Message Queue

Servicio de eventos COM+

Servicios de Internet Information Server

Los servicios de Internet Information Server (IIS) constituyen el servidor Web de Microsoft. En un principio, este producto se creó para Windows NT Server. La versión más reciente es IIS 5.0, que se distribuye con Windows 2000. Al igual que los demás productos del servidor Web, IIS administra las peticiones entrantes de HTTP enviadas por las aplicaciones cliente. Las versiones anteriores de IIS se utilizaron fundamentalmente para páginas Web estáticas. Actualmente, numerosas aplicaciones basadas en Web se

serven de IIS para ejecutar procesamientos personalizados en todas las peticiones entrantes.

IIS exhibe una API registrada denominada Internet Server API (ISAPI) para los desarrolladores que desean crear aplicaciones Web con una lógica personalizada del lado del servidor. Aquellos programadores que programan con ISAPI crean directamente módulos de software denominados extensiones ISAPI y filtros ISAPI. Aunque la escritura de software basado en ISAPI proporciona los niveles más elevados de rendimiento y flexibilidad, este enfoque también presenta costes significativos, ya que es preciso el desarrollo en C o C++ en lugar de en Visual Basic, con lo cual, los programadores se verán obligados a tratar con detalles de infraestructura de bajo nivel, como la escritura de un administrador de agrupamiento de subprocesos.

Numerosas empresas se niegan a programar directamente con ISAPI debido a que no disponen de la experiencia necesaria o no desean invertir el tiempo o el dinero necesarios para ponerla en práctica. IIS ofrece una alternativa a ISAPI con un marco de trabajo denominado páginas Active Server (ASP). El marco de trabajo de ASP es, en sí mismo, una extensión de ISAPI que permite a los programadores escribir una lógica del lado del servidor utilizando lenguajes de secuencias de comandos y Visual Basic.

Las aplicaciones Web que se crean mediante ISAPI o ASP proporcionan soluciones HTML, ahora bien, está en nuestras manos la decisión de con qué exploradores será compatible la aplicación. Podemos aprovechar al máximo las posibilidades de exploradores, como Internet Explorer en un entorno de intranet utilizando dichas aplicaciones web.

En un gran número de empresas se han generado sitios bastante sofisticados mediante el uso de ASP únicamente. Resulta muy sencillo escribir una lógica empresarial en el servidor y un código de acceso a datos utilizando lenguajes de secuencia de comandos como Visual Basic, Scripting Edition (VBScript) y JavaScript además de una herramienta de desarrollo como Microsoft Visual InterDev®. No obstante, las empresas también han reparado en la dificultad que conlleva la reutilización, el mantenimiento y la extensión de la lógica difundida en distintas páginas, dificultad que se hace más evidente a medida que aumenta el tamaño del sitio. Un enfoque más adecuado consiste en encapsular la lógica empresarial y el código de acceso a datos dentro de componentes compilados.

La integración entre ASP y COM+ facilita la creación y la ejecución de objetos empresariales personalizados desde una página ASP. De este modo, se puede distribuir la mayor parte de la lógica del servidor utilizando componentes en lugar de secuencias de comandos incrustados en las

páginas ASP. El uso de componentes facilita la reutilización, el mantenimiento y la extensión del código. Dichos componentes se pueden crear con Visual Basic, que ofrece mejores mecanismos de comprobación y depuración que las demás herramientas de desarrollo basadas en ASP. Numerosas empresas han reparado en el hecho de que el uso de ASP con Visual Basic proporciona el mejor equilibrio entre la productividad, el mantenimiento y el rendimiento.

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.

Vamos a ver en qué casos sería necesario el uso de MSMQ. Observemos un ejemplo típico. En una aplicación para administrar pedidos de ventas, los clientes pueden enviar los pedidos a una cola de peticiones aún cuando la aplicación del servidor no se esté ejecutando. Posteriormente, la aplicación del servidor puede abrir la cola y comenzar a procesar las solicitudes de pedido. MSMQ también permite que la aplicación del servidor devuelva un mensaje de respuesta al remitente como si se tratara de la respuesta a una llamada a métodos. Debemos tener en cuenta, que el proceso se ralentiza algo más cuando está involucrado el sistema de mensajería.

MSMQ resulta muy útil para aquellas empresas en las que los usuarios utilizan equipos portátiles que están conectándose y desconectándose constantemente de la red. Con MSMQ, se crean aplicaciones cliente que

envían mensajes a una cola de la red. Si se desconecta un equipo portátil, MSMQ almacena los mensajes de forma automática en una cola local temporal. Cuando el equipo portátil se vuelve a conectar a la red, MSMQ detectará que vuelve a estar disponible y enviará de forma automática los mensajes almacenados en caché a la cola de destino adecuada. Comprobamos que los aspectos esenciales del mecanismo de almacenamiento y envío se encuentran totalmente integrados en la plataforma subordinada.

Los productos de mensajería como MSMQ ofrecen mejores garantías de entrega que los productos basados en RPC o HTTP. Un mensaje MSMQ se puede enviar dentro del ámbito de una transacción para proporcionar una semántica de entrega de "sólo una vez". Esto indica que MSMQ adopta bastantes precauciones adicionales para garantizar que los mensajes llegan finalmente a su destino.

También proporciona un soporte de infraestructura adicional que devuelve al remitente notificaciones de error cuando los mensajes exceden el tiempo de espera o no se pueden enviar hacia su destino. De este modo, estamos seguros de que MSMQ entregará los mensajes a su destino o nos informará de que no se ha podido realizar la entrega. MSMQ elimina una gran cantidad

de problemas de pérdida de mensajes que afectan a las aplicaciones basadas en RPC y HTTP.

Componentes en cola

MSMQ se incorporó por primera vez a Windows NT Server en noviembre de 1997 con la versión de Windows NT Option Pack. Desde entonces, los programadores de Visual Basic se han servido de MSMQ para aprovechar las ventajas de la comunicación asincrónica y sin conexión. Sin embargo, la programación en MSMQ requiere código adicional para crear, preparar y enviar mensajes desde las aplicaciones cliente. También es necesario que escriba una aplicación de escucha (listener) del lado del servidor que reciba estos mensajes. Si se compara con la comunicación que se realiza con las llamadas a métodos de COM, en la comunicación con MSMQ resulta más difícil enviar una petición y obtener una respuesta.

COM+ proporciona un servicio denominado "Componentes en cola" que permite obtener mayores ventajas de MSMQ sin tener que programar de forma explícita con la API MSMQ. El servicio de componentes en cola no es más que una capa de productividad generada sobre MSMQ. Los componentes en cola se modifican exactamente de la misma forma que los componentes estándar de COM+ con una o dos limitaciones. Por ejemplo, no se pueden diseñar métodos con parámetros de salida o que devuelvan valores.

Una vez que se crea y se instala un componente en cola en una aplicación COM+, es preciso configurar un atributo que indique que la interfaz que se está utilizando se supone que está en la cola. También hay que configurar la aplicación COM+ para incluirla en la cola y que reciba mensajes. Al realizar esta operación, el servicio de componentes en cola crea automáticamente una cola especial para la aplicación. Además, configura un servicio proporcionado por el sistema que reciba mensajes para administrar los mensajes entrantes a medida que se reciben.

Una vez que el componente en cola se ha configurado adecuadamente en el servidor, se pueden escribir aplicaciones cliente que lo utilicen. Las aplicaciones cliente no solicitan directamente el objeto de un componente en cola, sino que crean un objeto especial proxy en el lado del cliente denominado "recorder" (grabadora), cuyo aspecto y funcionamiento es el mismo que el del objeto real en lo que se refiere al cliente. A continuación, el cliente comienza a invocar llamadas a métodos de la forma normal. El servicio de componentes en cola proporciona una infraestructura para grabar estas llamadas a métodos en un mensaje MSMQ y transportar estos datos a través de la red hasta el equipo donde se encuentra instalado el componente en cola. La parte del servidor del servicio de componentes en cola recibe el mensaje, crea una instancia del componente en cola y vuelve a realizar las llamadas a métodos.

El principal objetivo del diseño de los componentes en cola consiste en proporcionar las llamadas a métodos de COM más adecuadas junto con las ventajas que ofrece la comunicación asincrónica y sin conexión. En resumen, los componentes en cola utilizan MSMQ como protocolo subordinado de transporte en lugar de RPC. Los componentes en cola pueden evitar de este modo la gran cantidad de limitaciones de un protocolo sincrónico y orientado a la conexión.

No es necesario utilizar este tipo de servicio para aprovechar las ventajas de MSMQ. Siempre podemos programar directamente con MSMQ. El servicio de componentes en cola constituye un marco de trabajo orientado a la productividad que elimina los incómodos detalles de la programación en MSMQ. Como suele ocurrir con cualquier otro marco de trabajo, se proporciona una mayor productividad a costa de la flexibilidad.

Los componentes en cola no admiten todas aquellas posibilidades que sí se pueden realizar con MSMQ. Se debe utilizar el servicio de componentes en cola únicamente si éste admite las funciones necesarias y supone un ahorro considerable de tiempo de programación en MSMQ. Dentro de las funciones de mayor utilidad se encuentra el servicio de escucha (listener) del lado del servidor. Resulta casi imposible crear un servicio de escucha multiproceso utilizando únicamente Visual Basic, por lo que, si desea una aplicación de

escucha del lado del servidor que pueda procesar una gran cantidad de mensajes por segundo sin recurrir al desarrollo con C++, los componentes en cola es justo lo que necesitamos.

Apéndice D

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 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 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 el servidor de Web más fácil de administrar.

Características básicas

Protección contra errores graves La protección contra errores graves permite a los clientes ejecutar varias aplicaciones en el servidor Web con fiabilidad. Si una aplicación produce un error, el servidor Web y las demás

aplicaciones continuarán ejecutándose y la aplicación que falló se reinicia con la siguiente petición.

Active Server Pages con transacciones IIS presenta Transactional Active Server Pages, que permiten a las aplicaciones con secuencias de comandos y componentes realizar múltiples acciones, aceptando todas las acciones simultáneamente, o sin aceptar ninguna. Esto es decisivo para las aplicaciones de bases de datos.

Depuración de secuencias de comandos IIS ofrece una solución integrada para depurar secuencias de comandos y aplicaciones de Active Server Pages, lo que facilita la programación de aplicaciones ASP.

Cola de mensajes integrada Microsoft Message Queue Server (MSMQ) ofrece una forma fácil para que las aplicaciones envíen y reciban mensajes a través de una red con fiabilidad, incluso en el caso en que parte de la aplicación, o la red, no estén disponibles.

Compatibilidad con Java IIS incluye un conjunto de clases de Java para crear componentes del lado del servidor y Java 1.1 Virtual Machine de Microsoft. Si utiliza Active Server Pages y Microsoft Transaction Server,

puede crear aplicaciones Java y ejecutarlas fácilmente en entornos de servidor.

Control total del contenido Ahora los administradores pueden configurar el servidor Web en diferentes niveles: por servidor, por sitio, por directorio virtual, por subdirectorio o por archivo, lo que ofrece flexibilidad total en la mayoría de las configuraciones de servidor.

Administración de contenido y análisis de sitios IIS ofrece la capacidad de crear mensajes de error personalizados y restricciones PIC. Esta versión también incluye Microsoft Site Server Express, que le permite observar patrones de uso de su sitio y visualizar la estructura de su sitio a través de un archivo de registro básico y de análisis de contenido.

Múltiples sitios Web IIS admite múltiples sitios Web en una única dirección IP, operadores de sitios Web, ajuste de ancho de banda por cada sitio Web y administración mejorada y personalizable, que permite a las organizaciones alojar múltiples sitios de intranets de departamentos o múltiples sitios públicos de Internet.

Estándares de Internet IIS incorpora los estándares existentes y los más recientes de la industria, incluidos: HTTP 1.1 para un mayor rendimiento de

Internet, administrando y emitiendo certificados digitales X.509 y enviando mensajes a servidores estándar de noticias NNTP y de correo SMTP.

Compatibilidad con administración automatizada Con los nuevos objetos de administración de IIS puede escribir secuencias de comandos y ejecutarlas desde la línea de comandos, automatizando diferentes tareas de administración. Esta versión incluye un conjunto de secuencias de comandos de ejemplo que están listas para su utilización o personalización, de manera que satisfagan sus necesidades de administración.

Características adicionales

Servidor Web más fácil de instalar y configurar Si utiliza el asistente para descarga, puede seleccionar los componentes específicos que desea instalar. Además, la nueva capacidad de instalación desatendida permite instalar IIS en varios servidores sin tener que supervisar las instalaciones.

Herramientas flexibles de administración Las herramientas flexibles de administración del servidor Web IIS incluyen una consola integrada de administración basada en Windows, administración mejorada basada en el explorador, secuencias de comandos ejecutables desde la línea de

comandos, así como interfaces programables para crear herramientas de administración personalizadas.

Seguridad IIS incluye un servidor de certificados que está estrechamente integrado con el modelo de seguridad de Windows NT Server, lo que permite a las organizaciones emitir y administrar certificados digitales X.509 estándar de Internet.

Motor de búsqueda IIS incluye capacidades de búsqueda integradas, que permiten a los usuarios crear formularios de búsqueda personalizados con Active Server Pages, objetos de datos ActiveX y consultas SQL para buscar información en el servidor Web.

ASP

En los viejos días de los sitios web, la mayoría de ellos estaban comprendidos por un número determinado de páginas HTML fijas que podían ser hojeadas por el navegante tal y como las páginas de una revista. Poco a poco comenzaron a aparecer algunos sitios que ofrecían cierta interactividad estando respaldada esta por ciertos programitas llamados cgi que estaban escritos en una gran variedad de lenguajes.

Con el surgimiento de las nuevas tecnologías, los sitios comenzaron a incorporarlas para agregar interactividad y una multitud de bondades (páginas dinámicas, conexiones con bases de datos SQL, Access, Oracle, o cualquier otro motor que disponga de driver ODBC, comercio electrónico, portales, sedes personalizables y todas aquellas aplicaciones que requieren información dinámica, etc.) que el HTML por si solo no podía brindar. Uno de los lenguajes que más rápido se está extendiendo en la actualidad es ASP, que significa Active Server Pages (Páginas de Servidor Activas) y ha sido desarrollado por Microsoft. Pero ASP no es un lenguaje sino que es un entorno de programación del servidor.

Para trabajar en tal entorno, ASP utiliza códigos comunes de VBScript y JScript, ambos de los cuales son automáticamente soportados por ASP.

Requerimientos básicos

Para procesar páginas ASP no hay restricciones del lado del cliente (tanto el IE como el NS pueden acceder tranquilamente a una página .asp). Pero desde el lado del servidor este debe ser un servidor Microsoft y poseer una librería que interpreta el código (asp.dll). El servidor más popular es el Internet Information Server (IIS) versión 3.0 o superior que se utiliza en plataformas Windows NT. También gracias al Personal Web Server, una aplicación que Microsoft distribuyó a través del Windows 95 y 98, es posible instalar un servidor en una máquina individual o en una pequeña red de oficina u hogareña. La gran popularidad de las ASP provocó la aparición de intérpretes que las adaptan para plataformas UNIX, siendo los más conocidos Chilisoft e Instant ASP.

Lenguaje Primario

En el lenguaje HTML todas las etiquetas están encerradas entre este tipo de llaves: <>. En ASP se utilizan llaves con signos de porcentual, de esta manera: <% %>. Pero fuera de esta simbología, en la práctica tenemos que los códigos de ASP pueden aparecer dentro de etiquetas HTML o antes de ellas. Como lenguajes estándares ASP utiliza VBScript y JScript, aunque puede utilizar otros lenguajes. Si otros lenguajes son utilizados, se debe especificar el lenguaje en la parte superior de la página, de esta manera:

<%@LANGUAGE=LenguajeUtilizado%>

Objetos

Los objetos de ASP permiten la interacción entre el servidor y el cliente. No es necesario instalarlos todos, pero si hay que recordar abrirlos y cerrarlos. Los objetos de ASP más utilizados son los siguientes:

Request (Petición): Permite acceder a la información de un formulario, leer cookies, certificados de clientes, leer encabezados HTTP y analizar URLs codificadas.

Response (Respuesta): Es lo que permite enviar información al usuario, y permite escribir un documento, redirigir a otra página y crear cookies.

Application (Aplicación): A través de este objeto se pueden controlar aspectos relativos al inicio y finalización de una aplicación lo mismo que el acceso a información que está almacenada en el servidor.

Server (Servidor): El objeto server permite desarrollar funciones de rutina como el mapeo de un path virtual o físico y la creación de una instancia para un componente.

Session (Sesión): Este objeto permite albergar información sobre cada usuario que está visitando el sitio.

Componentes

Los componentes son controles ActiveX que sirven como interfaz con ASP para simplificar procedimientos comunes. Los más usuales son estos:

Ad Rotator (Rotador de Publicidad): Es un componente que permite presentar banners en una página de manera aleatoria.

Browser Capabilities (Capacidad del Browser): Permite determinar el navegador que utiliza el usuario y que aspectos soporta.

Collaboration Data Objects (CDO): Teniendo en cuenta que el servidor IIS posea SMTP (Simple Mail Transport Protocol, que es un protocolo de correo simple que se utiliza para la transferencia de e-mail a través de la Red), este componente permitirá enviar un e-mail a través de una página ASP, sin necesidad de enlazar a un script de Perl o cualquier otro CGI.

Content Linking: A través de este componente se puede tejer un sistema de links entre todas las páginas de un sitio de manera que no haya necesidad de escribir los típicos "anterior" o "siguiente". Incluso genera mapas del sitio.

Content Rotator (Rotador de Contenidos): En los sitios comerciales, la necesidad de rotar los contenidos atrae tráfico. Para ello se utiliza este componente que es capaz de añadir contenidos dinámicos a cualquier página sin necesidad de usar una base de datos. Su funcionamiento está amparado en trozos de código HTML que toma de un archivo de texto y que alterna en un espacio determinado en una página. Cada vez que carga el contenido, el rotador de contenidos presenta aleatoriamente uno de los trozos.

Database Access (Acceso a Datos): A través de este componente se puede enlazar una página a una base de datos y que la página tome sus contenidos de la base.

Además de estos componentes existe una gran variedad de componentes extras desarrollados por otras empresas además de Microsoft destinados a ASP.