

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

“Análisis, Diseño e Implementación de un Sitio Web para Pago de Servicios

TESIS DE GRADO

Previa a la obtención del título de:

LICENCIADO EN SISTEMAS DE INFORMACIÓN

Presentado por:

Dayanara Altamirano Maldonado
Nancy Castro Molina
Katrina Parrales Herrera

GUAYAQUIL – ECUADOR

AÑO

2002

AGRADECIMIENTO

Agradezco primeramente a Dios, por todo, porque si en Él nada sería.

A mi Papi, por que siempre me ha dado lo mejor. Me enseñó que lo mas importante en esta vida es mi relación con Dios, que sin Él nada somos. Podemos lograr todo en la vida pero si nos olvidamos de Dios de nada sirve el resto, Gracias Papi por todo, eres el mejor, se que no suelo decirlo siempre, pero créeme, Te quiero muchísimo,

A mi Mami por ser la mejor del mundo y apoyarme a lo largo de toda mi vida, gracias por brindarme tu confianza y ser mi mejor amiga, gracias por siempre estar dispuesta a recibir todos mis amigos en la casa para hacer proyectos, de verdad que te lo agradezco de todo corazon, y gracias por ser tan divertida, siempre haciéndome ver la vida tan alegre, Te quiero muchísimo Mami.

A mi ñañita linda porque sé que siempre se ha preocupado por mí, y aunque ella diga que no, siempre ha estado dispuesta a hacerme un favor cuando lo necesitaba, Te quiero mucho ñañita.

También un especial agradecimiento a mis 2 mejores amigas y compañeras de Tesis, Katrina y Paola, gracias por que hicimos el mejor equipo del mundo, porque compartimos momentos duros, pero no se imaginan cuanto nos divertimos.

Agradezco a mis compañeros de clase, a Daniel, Karina, Juan Jose, Viviana, Enrique, Fabricio y Fernando, que no solo fuimos compañeros de clase sino que nació una linda amistad que espero que dure por siempre, de todo corazón los quiero mucho.

A una gran amiga que siempre estuvo dispuesta a ayudarnos cuando se lo pedimos...Gracias Olguita.

A mi lindo y bello perrito Rocky, porque aunque no lo crean, el mas que nadie a estado en todas mis amanecidas junto a mi cama acompañándome.

Y finalmente, agradezco de todo corazón, a una persona muy especial en mi vida, que siempre me ha apoyado en los momentos más difíciles y quien me enseñó que no hay nada imposible, todo en esta vida se puede lograr si uno lucha y persevera por cumplir sus sueños, Te quiero un mundo Víctor.

¡Muchísimas Gracias a Todos!

Dayanara Altamirano Maldonado

Gracias a Dios por darme la existencia y estar junto a mi durante toda mi vida, sin la fé en Él nada hubiera logrado.

Gracias a mi Papi por su apoyo, por su compañía en las malas noches de proyectos, por su preocupación en ayudarme a conseguir una computadora en mis primeros años de estudio en la carrera, por estar siempre pendiente de todo lo que necesito.

Gracias a mi Mami por ayudarme incondicionalmente en todo, aconsejarme siempre y acompañarme en muchas malas noches.

Gracias a la Yadi, mi hermana, por su preocupación y darme siempre su opinión.

Papi, Mami y Yadi los quiero y los extraño muchísimo.

A mis amigas Dayanara y Katrina, mi grupo de amistad y estudio ya que me apoyaron en las situaciones y momentos más importantes de mi vida.

Finalmente le agradezco a mi esposo, Víctor Hugo, quien estuvo a mi lado dandome su apoyo constante y su amor. Te amo mucho.

Nancy Castro Molina

Primeramente quiero a agradecer a Dios porque siempre ha escuchado mis oraciones además me ha dado la fuerza para seguir adelante y ha iluminado mi vida, Gracias Diosito.

A mi papi, porque siempre ha estado a mi lado dándome su apoyo constante y brindandome sus sabios consejos, además me ha enseñado que la vida es dura, pero hay que saber luchar, gracias padre mio. Te adoro. ¡Nunca cambies Papito!.

A mi mamita querida porque para mí tú has sido y serás la mejor madre del mundo, has sabido escucharme y me has hecho reflexionar ante muchas circunstancias, Gracias por tus oraciones. Sin ti no sé que habría sido de mí.

A mis queridos hermanos Johanna y Miguelito, ya que ustedes han estado junto a mí en las buenas y en las malas ayudandome en todo momento y han soportado mis amanecidas y más que nada siempre tuvieron listo mi vaso de leche.

A mi lindisima ñañita Kerly que con solo un besito me has llenado de ternura y alegría, porque es super increíble llegar a casa y saber que sus ultimas palabras antes de dormir fueron ¿A que hora llega mi ñaña?. Te amo ñañita querida tú eres una luz en mi vida, este esfuerzo lo he hecho por ti también.

A mis amiguitas del alma Dayanara y Paola porque no solo hemos sido compañeras de tesis, hemos sido como hermanas, hemos compartido alegrías y penas y en este trabajo cada amanecida fue como una historia divertida que nos ayudó a mantenernos despiertas. Con ustedes la he pasado super bien. Espero que nunca cambien, las quiero mucho.

A mi amiguita Damaris que siempre me ha apoyado y me ha hecho ver la vida de una forma más fácil, Gracias Dama.

A la Sra Ada y a Don Angel que siempre me brindaron su ayuda. Muchas gracias.

A Olguita Cornejo, por su amistad incondicional, Gracias por guiarnos siempre.

A mis compañeros de clases, especialmente Enrique, Viviana, Karina, Juan José, Daniel, Fabricio y Fernando, porque la hemos pasado super chevere en la LSI, Gracias amiguitos.

Gracias a Todos.

Katrina Parrales Herrera

DEDICATORIA

Dedicamos este trabajo primeramente a Dios por que Él más que nadie sabe cuan importante ha sido en nuestras vidas y cuanto nos ha ayudado en este proyecto.

Gracias Dios.

A nuestras familias por su apoyo incondicional durante el transcurso de nuestra carrera.

A nuestros Amigos y Compañeros por todos los momentos increíbles que hemos compartido, en estudios, en nuestras clases, en nuestras amanecidas

Dayanara, Paola y Katrina

TRIBUNAL DE GRADUACIÓN

Ing. Mónica Villavicencio
Coordinadora

Ing. Carlos Martín
Director Tópico

MIEMBROS PRINCIPALES

Ing. Nestor Arreaga

LSI. Jorge Olaya

DECLARACIÓN EXPRESA

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

Srta. Dayanara Altamirano Maldonado

Sra. Nancy Castro Molina

Srta. Katrina Parrales Herrera

OBJETIVOS

Diseñar e implementar un Sitio web para pagos de servicios de Calidad que cumpla los siguientes objetivos:

- ✓ Cumplir con todos los requerimientos del Cliente, es decir que satisfaga todas sus necesidades que entre otras es tener consultas generales (de los pagos históricos) y específicas (consultas de planillas a cancelar) de manera rápida y oportuna a través del Internet.

- ✓ Funcionar de acuerdo a los requerimientos:
 - ✗ Correctamente sin errores.
 - ✗ Fácil acceso
 - ✗ Información adecuada.
 - ✗ Permitir consultar las planillas a pagar
 - ✗ Permitir realizar los pagos de servicios en línea.
 - ✗ Permitir Consultar Pagos Históricos.
 - ✗ Impresión y reimpresión de comprobantes de Pagos

- ✓ Facilitar la comunicación con el usuario

- ✓ Llegar a una Población de manera global

- ✓ Mantener información actualizada

- ✓ Mantener la seguridad en los datos.

DIRIGIDO A

Este trabajo está dirigido a aquellas personas cuyo deseo es siempre ir a la mano de los avances tecnológicos que nos ofrece el mundo informático, especialmente el Internet, y que además tienen la necesidad de agilizar procedimientos tediosos que toman tiempo, y que pueden ser realizados de forma inmediata a través del Internet desde donde se encuentre, como es, el de realizar sus pagos de servicios.

Teniendo presente que este sitio no solo será visitado por personas con conocimientos amplios en el ambiente computacional, sino también por personas con conocimientos básicos, fue diseñado tomando todas las consideraciones del caso, de un forma muy fácil y amigable de usar.

El sitio web podrá ser visitado por:

- ✓ Visitantes, público en general.

RESUMEN EJECUTIVO

El constante avance tecnológico nos exige mejorar ciertas actividades que han sido llevadas de una misma manera por años, como lo es el Pago de Servicios. Por esta necesidad nosotras hemos creado un sitio Web llamado “WebPagos” el cual le permitirá a cualquier persona consultar y efectuar el pago de sus servicios básicos necesarios desde un computador, lo cual facilitará la vida diaria de las personas, quienes para poder realizar sus pagos de servicios debían hacer colas interminables en las entidades correspondientes.

Nuestro objetivo es que el web site se convierta un sitio indispensable para la vida de los ecuatorianos, de manera que sientan que sólo a través de WebPagos faciliten sus gestiones de pago de servicios.

En este documento se especifican el análisis y los fundamentos teóricos y técnicos para el desarrollo de nuestro web site, WebPagos. Posteriormente se realiza el Diseño y la presentación de los pasos que se siguieron para la implementación.

INDICE GENERAL

	Pág.
Resumen Ejecutivo	
Indice General	
Introducción	
Capitulo 1	
1. Especificaciones Del Proyecto	1
1.1 Justificación.	1
1.2 Objetivos Generales.	4
1.3 Especificaciones Funcionales.	4
1.4 Especificaciones No Funcionales.	4
1.5 Limitaciones Y Restricciones.	5
1.6 Alcance.	5
Capitulo 2	
2. Análisis De La Industria Y Producto	7
2.1 Definición De La Industria Y Producto.	7
2.2 Clientes.	8
2.2.1 Comportamiento Del Consumidor Ante El Mercado Convencional.	8
2.2.2 Reacción Del Consumidor Ante El Mercado Virtual.	9
2.2.3 Mercado Virtual Vs. Mercado Convencional.	10
2.3 Competidores.	
2.3.1 Funcionalidad De Los Competidores Convencionales Existentes.	13
2.3.2 Desafío Encarados Por Competidores Convencionales.	14
2.4 Proveedores.	14
2.5 Productos Sustitutos.	15
2.6 Análisis Foda Del Sistema.	16
Capitulo 3	
3. Modelo De Negocios.	17
3.1 Definición Del Modelo De Negocios.	17
3.2 Funcionalidad General.	17

3.2.1 Diagrama Del Flujo De Proceso.	19
3.2.2 Esquema General De Funcionamiento.	22
3.3 Forma De Pago.	25
3.4 Seguridad.	26

Capitulo 4

4. Arquitectura Del Sistema.	28
4.1 Justificación De La Selección Del Modelo.	28
4.2 Tipos De Procesos Clientes.	28
4.3 Tipos De Procesos Servidores.	29
4.4 Tipos De Middleware.	30
4.5 Tecnologías Y Herramientas Escogidas Para La Implementación.	31
4.6 Requerimientos De Harware Y Software Para La Puesta En Producción.	31

Capitulo 5

5. Implementación De La Capa De Datos.	33
5.1 Modelo Lógico De Datos.	33
5.2 Modelo Físico De Datos.	34
5.3 Diagrama De Entidad - Relación.	35
5.4 Procedimientos Almacenados.	36
5.5 Proceso Servidor De Base De Datos.	50

Capitulo 6

6. Implementación De La Capa De Negocios.	51
6.1 Estructura De Los Componentes De Acceso A Datos.	51
6.2 Objetos Distribuidos De Acceso A Datos.	51
6.3 Comunicación Con El Proceso Servidor De Base De Datos.	52
6.4 Estructura De Los Componentes De Negocios.	54
6.5 Objetos Distribuidos De Negocios.	54
6.6 Comunicación Entre Objetos Distribuidos.	78
6.7 Proceso Servidor Transaccional.	78

Capitulo 7

7. Implementación De La Capa De Presentación.	81
7.1 Diseño Del Sitio Web - Cliente Basado En Browser.	81
7.2 Cliente Stand-Alone.	82
7.3 Cliente Html.	83
7.4 Cliente Script-Dhtml.	83

7.5 Server Script - Asp.	83
7.6 Css.	83
7.7 Comunicación Con Componentes De Negocio.	84
7.8 Proceso Servidor Web	85
Conclusiones	86
Recomendaciones	87
Bibliografía	88
Apéndices	
Apéndice A.- Arquitectura C/S	89
Apéndice B.- Programación En El Web.	96
Apéndice C.- Com/Dcom/Com+	101
Apéndice D.- IIS Y ASP	112

INTRODUCCIÓN

Frente a los avances tecnológicos como es el “Internet”, se ve la oportunidad de cubrir la necesidad de tener un medio de pago de servicios que permita una mayor rapidez, agilidad, seguridad y facilidad de acceso.

Por lo tanto presentamos nuestro trabajo, que fue desarrollado con la finalidad de suplir las necesidades actuales de las personas en el pago de sus servicios, presentando y exponiendo las diferentes inquietudes y necesidades de cada uno de nuestras fuentes de información(personas, empresas de servicios, investigaciones, sobre medios de pagos, Internet, etc).

Además podrá encontrar un Marco Teórico para su mayor comprensión, el estudio del mercado de los clientes y competidores, el análisis del Sitio donde se expondrá las necesidades, seguridad, Soluciones y beneficios. y la Tecnología utilizada en el diseño y desarrollo del mismo.

CAPITULO 1

1. ESPECIFICACIONES DEL PROYECTO

1.1 JUSTIFICACIÓN.

Previo a la justificación de nuestro proyecto queremos dar una breve explicación acerca de que es un pago a través del Internet.

Qué es un pago a través del Internet?

Comprar y vender cualquier bien es algo tan común que ya hoy pertenece a nuestro inconsciente. Salir a la calle a realizar compras sin efectivo, cheques, o tarjetas de crédito, parece simplemente absurdo.

En la actualidad, utilizamos diversos medios de pagos, siendo los de uso más frecuente los siguientes:

- ✓ Efectivo (Cash)
- ✓ Cheque
- ✓ Tarjetas de débito y crédito
- ✓ Tarjetas Pre-Pago
- ✓ Smart-Cards
- ✓ Money Orders

Los medios de pagos en Internet son la utilización de los instrumentos mencionados anteriormente en actividades de índole comercial, bien sea para la adquisición de bienes o servicios, utilizando para ello la red de redes (Internet),

con las mismas características de uso que en un ambiente de relaciones comerciales tradicionales.

Un conjunto de características diferenciales, al no desarrollarse la actividad en el mundo real sino en el virtual, son necesarias para el éxito, desarrollo e incremento del comercio electrónico. Estas características son:

- ✓ **Facilidad de uso:** Comodidad y rapidez en las transacciones en línea.
- ✓ **Universalidad:** Aceptación en todas partes, permitiendo pagar cualquier compra o servicio.
- ✓ **Liquidez del instrumento:** Debe permitir pagar cualquier tipo de monto sin importar el tamaño del mismo.
- ✓ **Fraccionamiento:** Habilitado para pagos exactos en céntimos.
- ✓ **Intimidad:** Debe borrar todo rastro del usuario que lo ha utilizado, así como también debe ocultar el tipo de compra, la adquisición efectuada y el importe de la misma.
- ✓ **Seguridad en el uso:** En caso de robo o extravío, si llegase a ser utilizada, no va a funcionar, y en caso extremo de hacer funcionar el instrumento, no podrían pagarse las compras.
- ✓ **Garantía de que el dinero lo reciba el acreedor y no otra persona:**
Autenticación de la fuente y del receptor del pago
- ✓ **Acreditación del pago:** Emisión de un recibo para las partes involucradas.
- ✓ **Transacciones libres de costo:** Que no haya intermediarios entre el vendedor y el comprador.

- ✓ **Facilidad y bajo costo de implementación:** Para las empresas que adopten estos medios, debe ser de fácil implementación y a bajo costo, tanto de instalación como de mantenimiento.

Se define un pago electrónico de un producto o servicio como uno en el cual el usuario inicia y completa la sesión de pago entera, vía Internet (incluyendo la vía e-mail): la forma como el comerciante selecciona procesar la orden de transacción (por ejemplo, manual o electrónicamente).

Tomando en cuenta el análisis de la investigación especificada anteriormente nace la oportunidad de desarrollar un proyecto, que permita cubrir la necesidad de tener un medio de pago de servicios que pueda ser realizado con una mayor rapidez, agilidad, seguridad y facilidad de acceso, como lo es, a través del Internet, desde cualquier lugar donde se encuentre.

1.2 OBJETIVOS GENERALES.

Crear un sitio Web que le permita a cualquier persona poder consultar y pagar todos los servicios básicos y necesarios para facilitarle su vida diaria, quienes para realizar sus pago debian hacer colas interminables en las entidades correspondientes.

Nuestro objetivo es que el sitio web se convierta un sitio indispensable para la vida de los ecuatorianos, de manera que sientan que sólo a través de WebPagos faciliten sus gestiones de pago de servicios.

1.3 ESPECIFICACIONES FUNCIONALES.

Crear un sitio virtual que permita realizar pagos de Servicios a traves del Internet de una de forma sencilla y amigable. El cliente accesará a nuestro sitio www.WebPagos.com el cual deberá ser un sitio seguro y confiable. Este deberá permitir que el cliente pueda realizar consultas especificas de sus deudas pendientes y sus pagos realizados.

1.4 ESPECIFICACIONES NO FUNCIONALES.

En nuestro sitio web se registra automáticamente todas las transacciones realizadas por el cleinte, permitiendo en cualquier momento obtener estadísticas si así el cliente lo desea.

Debido a la tecnología implementada en nuestro sitio web, ofrecemos escalabilidad a nivel de usuarios, sin afectar el rendimiento de la aplicación.

1.5 LIMITACIONES Y RESTRICCIONES.

Entre las limitaciones tenemos las siguientes:

- ✓ Un usuario no podrá cancelar ninguna planilla si no se encuentra registrado en el sistema.
- ✓ Ningún usuario podrá realizar el pago de una planilla si no tiene saldo disponible.
- ✓ No se podrá cancelar una planilla si su fecha tope de pago ha vencido.

Entre las restricciones tenemos las siguientes:

- ✓ No poder adquirir la tarjeta prepago para la cancelación de una planilla.
- ✓ Que el usuario no tenga un computador acceso a Internet.

1.6 ALCANCE.

El desarrollo del proyecto de WebPagos, primeramente es de alcance nacional, y deberá permitir las consultas y pagos de servicios públicos, básicos, masivos y frecuentes, como lo son:

- ✓ Luz
- ✓ Agua
- ✓ Teléfono
- ✓ Cable por televisión
- ✓ Cuentas de Internet

Permitiendo a su vez adicionar más servicios, aunque sean estos de otro tipo.

En cuanto a la forma de pago, se utilizaran tarjetas de Prepago, cuyo cupo será adicionado al saldo disponible del usuario.

A continuación se indican las opciones adicionales que forman parte de los beneficios que ofrece WebPagos a los clientes:

- ✓ Registro de Usuarios
- ✓ Actualización de Datos del usuario
- ✓ Activación de Tarjetas
- ✓ Consulta Histórica (de pagos realizados)
- ✓ Consulta de Saldo Disponible
- ✓ Impresión y Reimpresión de Comprobantes de Pago

CAPITULO 2

2. ANÁLISIS DE LA INDUSTRIA Y PRODUCTO

2.1 DEFINICIÓN DE LA INDUSTRIA Y PRODUCTO.

Debido al constante avance tecnológico y las necesidades de empresas de servicios y de las personas, hemos desarrollado un innovador sistema de pago virtual, llamado www.WebPagos.com; el cual permitirá a las empresas ofrecer múltiples servicios a través del Internet y a las personas el ahorro de tiempo y esfuerzo para realizar el pago de deudas, siempre teniendo la certeza de que sus transacciones son efectivas, seguras y confiables.

Luego de haber analizado las necesidades y haber recopilado información de los aspectos relacionados con el pago virtual de servicios presentamos la siguiente solución:

Diseñar e Implementar un Sitio Web para el Pago de Servicios, el cual permitirá a cualquier persona poder consultar y pagar todos los servicios básicos y necesarios a través de Internet, para facilitarle su vida diaria.

Servicios básicos tales como:

- ✓ Luz
- ✓ Agua
- ✓ Teléfono

- ✓ Cable
- ✓ Cuentas de Internet

Este sitio web nos permitirá:

- ✓ Realizar consultas de deudas que se tienen pendientes de pago.
- ✓ Realizar consultas de los pagos realizados en una fecha determinada.
- ✓ Tener la facilidad de obtener re-impresiones de los recibos de pago.
- ✓ Presentar información general de tal manera que todos puedan acceder a ella.
- ✓ Presentar consultas de saldos de cuentas.

Las empresas proveedoras de servicios que deseen integrarse a WebPagos deben firmar un convenio de afiliación.

Además WebPagos solicita que la información de los valores a cobrar nos sea enviada en un archivo plano con un formato preestablecido, de la misma manera nosotros nos comprometemos a enviar diariamente los valores recaudados.

2.2 CLIENTES.

2.2.1 COMPORTAMIENTO DEL CONSUMIDOR ANTE EL MERCADO CONVENCIONAL.

Actualmente el Pago de servicios se lo realiza de la siguiente manera:

Las planillas son llevadas al domicilio o lugar donde reside el consumidor, este a su vez tiene que acercarse a las entidades correspondientes del servicio, a cancelar sus pagos, lo cual causa una pérdida de tiempo y esfuerzo ya que debe realizar

colas interminables para ello. Además, si el consumidor por algún motivo pierde sus comprobantes de pago este deberá realizar el procedimiento antes mencionado. En ocasiones esto es una gran molestia para el consumidor.

2.2.2 REACCIÓN DEL CONSUMIDOR ANTE EL MERCADO VIRTUAL.

El consumidor al inicio puede sentir temor en la utilización de los sistemas virtuales es el Pago de Servicios, al pensar que no son confiables, que puede existir fraude electrónico o que son complicados de manejar. Pero, a través del tiempo este mito ha ido desapareciendo, ya que el consumidor comprueba que un sistema virtual es una herramienta fácil, segura, necesaria e incluso se puede convertir en algo indispensable, para su vida cotidiana.

Actualmente, El mercado virtual ha tenido un gran auge y crecimiento en el Ecuador, ya que muchas empresas realizan sus negociaciones a través de Internet.

A nivel de Latinoamérica las estadísticas han confirmado un alto crecimiento en pagos a través del Internet, lo cual indica que el consumidor esta reaccionando de manera positiva ante este tipo de mercado.

2.2.3 MERCADO VIRTUAL VS. MERCADO CONVENCIONAL.

Ventajas del Mercado Virtual

VENTAJA	DESCRIPCION
✓ Facilidad de Acceso.	✓ Permite que cualquier usuario que tenga un computador con conexión a Internet, pueda acceder a nuestro sitio WebPagos desde cualquier lugar.
✓ Agilidad al efectuar un pago.	✓ Permite que cualquier usuario pueda efectuar el pago de sus deudas de una manera más rápida sin realizar colas interminables.
✓ Comodidad al obtener información al Instante.	✓ Permite que cualquier usuario pueda hacer diversas consultas de sus transacciones realizadas en el momento que lo necesite.
✓ Reducción de costos debido a la eliminación de intermediarios.	✓ El acceso al sitio web permite que se eliminen costos operativos y administrativos ya que no existirá la presencia de personas para la atención al público.

Desventajas del Mercado Virtual

DESVENTAJA	DESCRIPCION
<ul style="list-style-type: none"> ✓ Actualización del pago no inmediato. 	<ul style="list-style-type: none"> ✓ La actualización de los pagos realizados en el sitio virtual es de tipo batch, se generará un archivo donde consten los datos de los pagos al final del día.
<ul style="list-style-type: none"> ✓ No tener conexión a Internet. 	<ul style="list-style-type: none"> ✓ Se depende de la conexión de Internet para poder acceder a las ventajas que ofrece nuestro sitio web.
<ul style="list-style-type: none"> ✓ Altos costos por uso telefónico (conexiones dial-up). 	<ul style="list-style-type: none"> ✓ En nuestro medio el costo por el uso del teléfono es alto por lo que muchas veces las personas limitan el uso de Internet en sus hogares.

Ventajas del Mercado Convencional

VENTAJA	DESCRIPCION
<ul style="list-style-type: none"> ✓ Actualización en línea del pago. 	<ul style="list-style-type: none"> ✓ La cancelación de la deuda realizada en la empresa proveedora del servicio se ve reflejada automáticamente en la base de datos.
<ul style="list-style-type: none"> ✓ Confiabilidad en el uso de este servicio. 	<ul style="list-style-type: none"> ✓ A través del tiempo las personas se han acostumbrado en la utilización del sistema convencional por lo que tienen miedo a hacer un cambio al sistema virtual.
<ul style="list-style-type: none"> ✓ Personalización en el servicio. 	<ul style="list-style-type: none"> ✓ En el sistema convencional la atención al cliente es persona a persona.

Desventajas del Mercado Convencional

DESVENTAJA	DESCRIPCION
✓ Pérdida de tiempo al realizar la transacción.	✓ Al realizar el pago de servicios de forma convencional se pierde mucho tiempo.
✓ Esfuerzo Físico del Cliente.	✓ Las largas colas que se realizan para efectuar un pago se vuelven tediosas y provocan molestias al cliente.

2.3 COMPETIDORES.

2.3.1 FUNCIONALIDAD DE LOS COMPETIDORES CONVENCIONALES EXISTENTES.

El sistema convencional de pago de servicios se realiza de la siguiente manera:

La empresa proveedora del Servicio a ser cancelado emite sus planillas físicas, es decir imprime el valor a pagar en un documento, el cual debe ser repartido a todos los clientes a su domicilio, lugar de trabajo, etc.

En esta planilla generalmente constan datos del abonado como: nombre, dirección, valor a pagar, fecha tope de pago, consumo del servicio, impuestos de ley, etc. El cliente una vez revisada su planilla se debe acercar a la empresa proveedora del servicio hasta la fecha tope de pago y cancelar su deuda. El servicio que ofrece la empresa es personalizado es decir el cliente recibe de forma directa su comprobante de pago.

Entre los competidores convencionales existentes tenemos los siguientes:

- ✓ Ventanillas de las empresas proveedoras de Servicios.
- ✓ Ventanillas en los Bancos.
- ✓ Ventanillas de ServiPagos.

2.3.2 DESAFÍO ENCARADOS POR COMPETIDORES CONVENCIONALES.

Nuestro sistema virtual de pago de servicios “WebPagos” no significa un desafío para las empresas proveedoras de servicios más bien deben verlo como un beneficio, el cual representa una ventaja tecnológica.

Para el caso de los Bancos y Empresas como ServiPagos, su desafío sería agilizar la atención en ventanilla y mejorar la calidad de atención al cliente.

2.4 PROVEEDORES.

Proveedor es la persona o compañía que recibirá los fondos del cliente. En nuestro caso se trata de las empresas proveedoras de servicios tales como:

- ✓ Empresas de Luz.
- ✓ Empresas de Agua.
- ✓ Empresas de Teléfono.
- ✓ Empresas de Cable.
- ✓ Empresas de Internet.

2.5 PRODUCTOS SUSTITUTOS.

Un producto sustituto que se encuentra actualmente en el mercado virtual es: www.mis-pagos.com, el cual cumple con las mismas funciones de WebPagos.

La ventaja principal que posee www.mis-pagos.com sobre WebPagos es que es un producto que se encuentra en el mercado, que ya ha ganado usuarios. Además ofrece la personalización de los pagos, es decir permite al usuario llevar un control de sus deudas por vencer mediante el envío de notificaciones vía e-mail.

La ventaja que tiene WebPagos sobre www.mis-pagos.com es que en nuestro sitio el usuario no necesita tener cuenta, ni tarjeta de débito en ninguna entidad bancaria.

2.6 ANÁLISIS FODA DEL SISTEMA.

FORTALEZAS	OPORTUNIDADES
<ul style="list-style-type: none"> ✓ Cancelación de pagos a través de tarjetas pre-pago ✓ Facilidad de Acceso. ✓ Agilidad al efectuar un pago. ✓ Pagar sus planillas a cualquier hora del día. 	<ul style="list-style-type: none"> ✓ Captar clientes que no posean cuentas bancarias. ✓ Captar clientes que se encuentren desde cualquier lugar. ✓ Captar clientes que desean pagar sus deudas de una manera más rápida. ✓ Captar clientes que trabajan o trasnochadores.

DEBILIDADES	AMENAZAS
<ul style="list-style-type: none"> ✓ No tener conexión a Internet. 	<ul style="list-style-type: none"> ✓ Perder Clientes por estar fuera de línea por un lapso de tiempo.

CAPITULO 3

3. MODELO DE NEGOCIOS.

3.1 DEFINICIÓN DEL MODELO DE NEGOCIOS.

Nuestro sitio WebPagos, esta basado en el modelo de negocios B2C (Business to Consumer), ya que se ajusta a las normas de Vender a una persona particular servicios o productos a traves del Internet.

El modelo de negocio de WebPagos consiste en brindar un valor agregado, al cliente de una empresa proveedora de servicios. Por este motivo viene a ser una herramienta puesta a servicio del cliente.

3.2 FUNCIONALIDAD GENERAL.



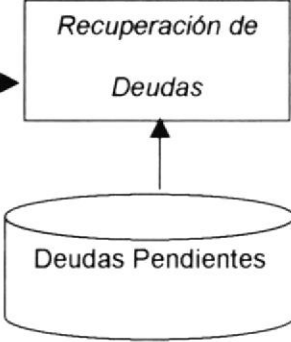
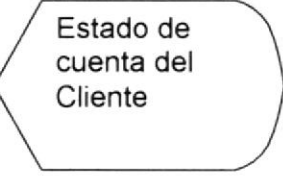
El principal propósito de WebPagos es el de permitir que cualquier persona pueda consultar y efectuar sus pagos de todo servicio basico y necesario que haya sido publicado en este sitio Web, a continuación se detalla brevemente el esquema de funcionamiento de WebPagos:


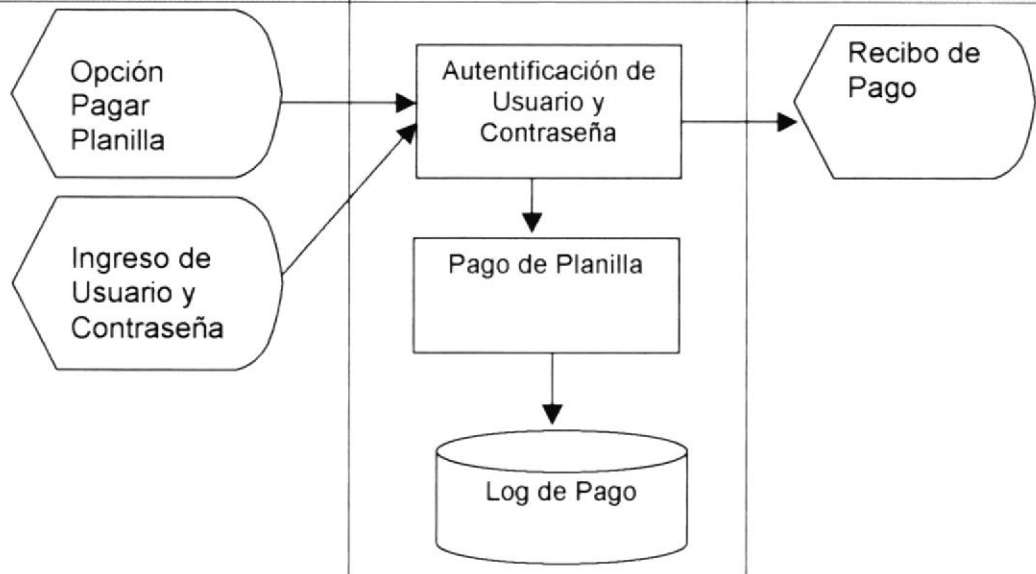
- ✓ El cliente ingresa los datos del servicio que desee consultar.
- ✓ Esta transacción es procesada, registrada por medio del Servidor de Internet.
- ✓ El Servidor de Internet se conecta a una base de datos para extraer los valores adeudados por el cliente




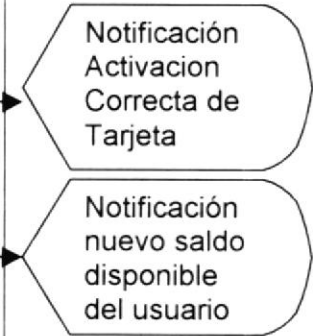
- ✓ La información de los valores que constan como deudas pendientes es enviada al cliente, para que este a libre elección, decida si desea o no realizar el pago del servicio.
- ✓ Si el cliente decide efectuar el pago, WebPagos recibe y valida esta información.
- ✓ La respuesta del pago es enviada al Servidor de Internet
- ✓ El Servidor de Internet envía la respuesta del pago a la pantalla del cliente

3.2.1 DIAGRAMA DEL FLUJO DE PROCESO.

Se presenta de manera gráfica los procesos más importantes que intervienen en el sitio WebPago

 Escuela Superior Politécnica del Litoral		Descripción de Sitio Web	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: ConsultaPlanilla.ASP	Página HTML – Página ASP		
Diagrama de Proceso: Página de Consulta de Planilla			
I	P	O	
			
Descripción de Procesos			
<ol style="list-style-type: none"> 1. Desde la pagina Consulta Servicio se escogen e ingresan los datos necesarios para poder consultar las deudas de un cliente. 2. Validar y procesar para luego presentar al cliente el valor de su deuda pendiente a través de la página Consulta Planilla. 			
Observaciones			
Las tablas que intervienen en el proceso pertenecen al sistema de Administración de Sitio Web.			

	Escuela Superior Politécnica del Litoral	Descripción de Sitio Web	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: PagoRelizado.ASP	Página ASP.		
Diagrama de Proceso Página de Pago Realizado			
<p style="text-align: center;">I</p> 	<p style="text-align: center;">P</p>	<p style="text-align: center;">O</p>	
Descripción de Procesos			
<ol style="list-style-type: none"> 1. Desde la Página de Consulta Planilla escoge opción Pago de Planilla y luego se ingresa el usuario y contraseña. 2. Verificar usuario y contraseña y todas las validaciones que se necesitan para que el pago se realice correctamente, todas estas validaciones quedan registradas en sus respectivas tablas y en logs que servirán para realizar consultas en un futuro. 3. Emitir un recibo único que sirve como comprobante del pago efectuado. 			
Observaciones			
Las tablas que intervienen en el proceso pertenecen al sistema de Administración de Sitio Web.			

	Escuela Superior Politécnica del Litoral	Descripción de Sitio Web	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: ActivarTarjeta.ASP	Pagina HTML y Página ASP.		
Diagrama de Proceso Página de Activación de Tarjeta.			
I	P		O
			
Descripción de Procesos			
<ol style="list-style-type: none"> 1. Desde la página Activación de Tarjeta deberá ingresar el código de la nueva tarjeta, el usuario y contraseña correspondiente. 2. Activar la tarjeta de prepago para un usuario específico con un saldo determinado. 3. Emitir notificación de la activación correcta de una tarjeta de prepago para un usuario. 4. Presentar el nuevo saldo disponible del usuario. 			
Observaciones			
Las tablas que intervienen en el proceso pertenecen al sistema de Administración de Sitio Web.			

3.2.2 ESQUEMA GENERAL DE FUNCIONAMIENTO.

PARA CONSULTAS

Un Cliente puede acceder a WebPagos ingresando a la siguiente dirección de Internet : www.WebPagos.com

WebPagos será un sitio confiable, ya que cumplirá con los debidos estándares de seguridad y además obtendrá certificación de Verisign la cual nos protegerá de factores externos que pudiesen ser perjudiciales para el sitio.

Primeramente el usuario tendrá disponible la opción de consulta de servicios, en donde presentará un menú con las siguientes opciones:

- ✓ Teléfono
- ✓ Luz
- ✓ Agua
- ✓ Cable por TV
- ✓ Internet

El cliente elegirá el servicio a consultar, y al darle click sobre dicho servicio, se displayarán los nombres de las compañías/empresas que otorgan este servicio, por ejemplo:

- ✓ Teléfono
 - ✓ Andinatel
 - ✓ Pacifictel

El cliente elegirá la compañía/empresa con la cual realizará su consulta/pago, y al darle click deberá aparecerle los siguientes campos a ser llenados:

Para el Servicio de Teléfono Convencional

Primeramente deberá escoger la empresa a la que el usuario esta suscrito.

- ✓ **Código de Área.**- Deberá existir una ayuda donde se liste los códigos de área del país en un combo box.
- ✓ **Número de teléfono** (deberá ingresar su número de teléfono anteponiendo el número 2 sin separaciones ni caracteres especiales)

Para los servicios de luz y agua

Primeramente deberá escoger la empresa a la que el usuario esta suscrito.

- ✓ Número de Cuenta /Medidor

Para compañías de servicio de Tv-cable

Primeramente deberá escoger la empresa a la que el usuario esta suscrito.

- ✓ Número de contrato/o membresía

Para compañías de servicio de Internet

Primeramente deberá escoger la empresa a la que el usuario esta suscrito.

- ✓ Número de contrato

Importante

En las opciones de consulta anteriormente mencionadas solo se podrá consultar la planilla del mes actual.

Una vez ingresado los datos necesarios, el sistema verificará en la base de datos el servicio indicado, el valor a pagar y lo presentará en forma de una planilla de pago.

Para poder presentar toda la información de dicha planilla es necesario realizar diariamente un proceso de carga, que tiene como finalidad subir a la base de datos de WebPagos todos los saldos que cada empresa proveedora de servicios suscrita en WebPagos maneja actualmente.

PARA PAGOS

Se podrán realizar los pagos de los servicios anteriormente especificados, durante las 24 horas del día

Un Cliente puede acceder a WebPagos ingresando a la siguiente dirección de Internet : www.webpagos.com igual procedimiento al descrito en consultas

Luego de elegir el servicio, la empresa y conocer el valor a pagar podrá proceder a dar click en botón **pagar**.

Después de esto, el sistema como medida de seguridad se le solicitará que ingrese su usuario y contraseña para así poder realizar el pago, ya que solo se puede pagar una planilla si es un usuario registrado y además con saldo disponible en WebPagos y listo se ha procedido a efectuar un pago.

Importante

Después que se efectúe un pago en www.WebPagos.com aparecerá un mensaje “Su pago ha sido realizado con éxito” y un **Número de Transacción** que sirve como identificador del pago realizado, el cual que es único entre todas las transacciones realizadas en WebPagos, el mismo que se imprimirá en el comprobante de pago y se almacenará en la base de datos, para que sea enviado como un campo más dentro del archivo que se envía a las empresas con el detalle de los pagos recaudados **por Internet**

Además se presentará una imagen de Impresión del comprobante de pago, el cual constará de los datos de l cuenta cancelada, como nombre del titular de la cuenta, empresa a la que pertenece la cuenta, monto cancelado, fecha del pago y el Numero de Transacción correspondiente.

3.3 FORMA DE PAGO.

WebPagos ha implementado una innovadora forma de realizar sus pagos en internet, esto se refiere a que se ha utilizado un sistema de tarjeta de prepago, es decir que el cliente compra una tarjeta con un monto preestablecido de cupo. El cual para poder utilizarlo deberá primeramente Activar dicha Tarjeta en el sitio WebPagos, simplemente debe ingresar su usuario y contraseña y luego se le solicitará que ingrese el código secreto de su tarjeta y listo el cupo de dicha tarjeta ha sido acreditado a su cuenta y podrá hacer uso de él cuando lo necesite.

El costo de dicha tarjeta es cero, es decir que usted hace un intercambio de un monto determinado por una tarjeta del mismo valor el cual solo le servirá para hacer sus pagos de servicios básicos a través de WebPagos sin ningún recargo adicional.

A medida que usted efectúe sus pagos en WebPagos el saldo disponible que obtuvo por la activación de una tarjeta de prepago irá disminuyendo equivalentemente.

Si en algún momento usted al querer realizar sus pagos no tiene saldo disponible en su cuenta, deberá adquirir una nueva tarjeta y hacer el proceso de activación de la misma.

Al momento de activar una tarjeta de prepago, ya no puede volver a activarse. ya que dicha tarjeta en el proceso de activación queda deshabilitada, de esta manera evitamos fraudes a las empresas proveedoras de servicios suscritas en Webpagos

La tarjeta de prepago será conocida como WebPrepago y estará disponible para todo público ya que será distribuida a nivel nacional.

3.4 SEGURIDAD.

En Internet la palabra “Seguridad” cobra mayor importancia, ya que la misma debe estar presente en toda la ejecución de una transacción, debido a la naturaleza de medio público que reviste a la Red, debiéndose garantizar la privacidad de los datos de los usuarios, al efectuar transacciones.

De hecho, en la actualidad el mayor temor de los consumidores de realizar pagos a través de Internet es precisamente no tener seguridad en los mismos, sin embargo, cada vez más aumenta el número de transacciones, y los montos de las mismas.

Organizaciones, como www.fraud.org estiman que la media de dinero que los consumidores pierden debido al fraude virtual va en franco aumento, y en la misma proporción los montos de las transacciones. Lo más importante para todas estas organizaciones, es lograr disminuir de manera efectiva el fraude y que no mantenga un crecimiento proporcional con el volumen de transacciones.

Todo esto obliga al mercado a buscar distintas formas de pago que minimicen los riesgos de fraude y sirvan para ganarse la confianza de los usuarios de la red.

CAPITULO 4

4. ARQUITECTURA DEL SISTEMA.

4.1 JUSTIFICACIÓN DE LA SELECCIÓN DEL MODELO.

En el desarrollo de nuestro proyecto utilizamos la arquitectura de tres capas ya que de esta forma nos permite manejar de una manera óptima y confiable la transaccionalidad.

Mediante las tres capas hemos dividido la aplicación en secciones específicas de cliente, reglas del negocio y acceso a datos, permitiendo de esta forma lograr independencia de ubicación de nuestros componentes así como también de la herramienta de programación utilizada.

El modelo de tres capas nos ha permitido obtener óptimos resultados en cuanto a la funcionalidad de WebPagos, permitiendo a su vez escalabilidad.

4.2 TIPOS DE PROCESOS CLIENTES.

WebPagos está compuesto por dos tipos de clientes, los cuales son: dependiente e independiente.

- ✓ **Cliente Dependiente.**- Es el sitio virtual a ser utilizado por el usuario consumidor, que es a quien va dirigida la funcionalidad de WebPagos. El lenguaje de programación en que fue desarrollado es ASP, utilizando como

herramienta de soporte Visual InterDev. En este caso la aplicación se ve obligada a depender de un medio para ser ejecutada como es el browser Microsoft Internet Explorer.

- ✓ **Ciente Independiente.**- Es el sistema aplicativo en donde se realizará el mantenimiento de los datos que serán utilizados en el sitio virtual. Fue desarrollada en Microsoft Visual Basic 6.0. Esta aplicación está dirigida al usuario administrativo. La misma que no depende de un contenedor para ser ejecutada. Se utilizó este tipo de cliente debido a que no necesita ser publicado en Internet ya que será utilizado por usuarios internos

4.3 TIPOS DE PROCESOS SERVIDORES.

En el desarrollo de WebPagos se utilizó tres tipos de procesos servidores: Servidor TP, servidor web y base de datos.

- ✓ **Servidor TP.**- Es el proceso servidor que se encarga del manejo de la transaccionalidad de un sistema. Utilizamos Com+ para manejar de una forma eficiente la transaccionalidad del sitio web, ya que es fácil el distribuir, gestionar y monitorizar las aplicaciones de capas, eliminando el exceso de tener que usar varias herramientas independientes de administración.
- ✓ **Servidor web.**- Microsoft Internet Information Server (IIS) es el servidor web, que se utilizó para la publicación y administración de nuestro sitio web. Su uso es ventajoso debido a la compatibilidad y posibilidades de uso conjunto con los productos de la familia Microsoft como lo es SQL Server.

Entre los servicios básicos que nos proporciona IIS hemos utilizado el WWW y el SMTP para el envío de mails.

- ✓ **Base de datos.**- Es el repositorio donde se van a almacenar los datos del sitio virtual. En nuestro caso utilizamos Microsoft SQL Server 7.0 como proceso de Base de Datos.

4.4 TIPOS DE MIDDLEWARE.

A continuación se presentan los diferentes tipos de Middleware que se utilizan en el sitio WebPagos

La tecnología COM+ se utiliza para manejar la transaccionalidad de la base de datos y para tener un óptimo control de instancias de los objetos.

La tecnología COM+ y Microsoft Windows 2000 proporcionan una sólida plataforma de desarrollo, la cual es utilizada en WebPagos. Esta plataforma se compone de diversas tecnologías esenciales que proporcionan los componentes básicos para generar aplicaciones empresariales de varios niveles y reduce su complejidad.

El Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP

OLDB es utilizado para acceder al Servidor de base de datos y además para tener más control sobre el comportamiento de la aplicación y el rendimiento

4.5 TECNOLOGIAS Y HERRAMIENTAS ESCOGIDAS PARA LA IMPLEMENTACIÓN.

A continuación se presentan las tecnologías que hemos utilizado para el desarrollo del sitio web WebPagos.

- ✓ Tecnología COM+
- ✓ Active Server Page
- ✓ Protocolo HTTP
- ✓ Tecnología OLEDB
- ✓ Tecnología ADO

Las herramientas utilizadas son las siguientes:

- ✓ Microsoft Visual Interdev 6.0
- ✓ Microsoft Visual Basic 6.0
- ✓ Microsoft Sql Server 7.0

4.6 REQUERIMIENTOS DE HARWARE Y SOFTWARE PARA LA PUESTA EN PRODUCCIÓN.

Requerimientos de Harware (Server)

Un computador con las siguientes características :

- ✓ Capacidad en DD 40GB o superior

- ✓ 256 MB en Memoria Ram o superior
- ✓ Procesador Pentium III o superior
- ✓ Con conexión a Internet dedicada.

Requerimientos de Hardware (Cliente)

Un computador con las siguientes características :

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

Requerimientos de Software (Server)

El software que se requiere es el siguiente:

- ✓ Windows 2000 que incluya Com+ e Internet Information Server (IIS).
- ✓ Microsoft SqlServer 7.0 (SP3)
- ✓ Microsoft Internet Explorer 4.0 o superior
- ✓ ADO 2.5 es utilizado para los accesos a las bases de datos SQL

Requerimientos de Software (Client)

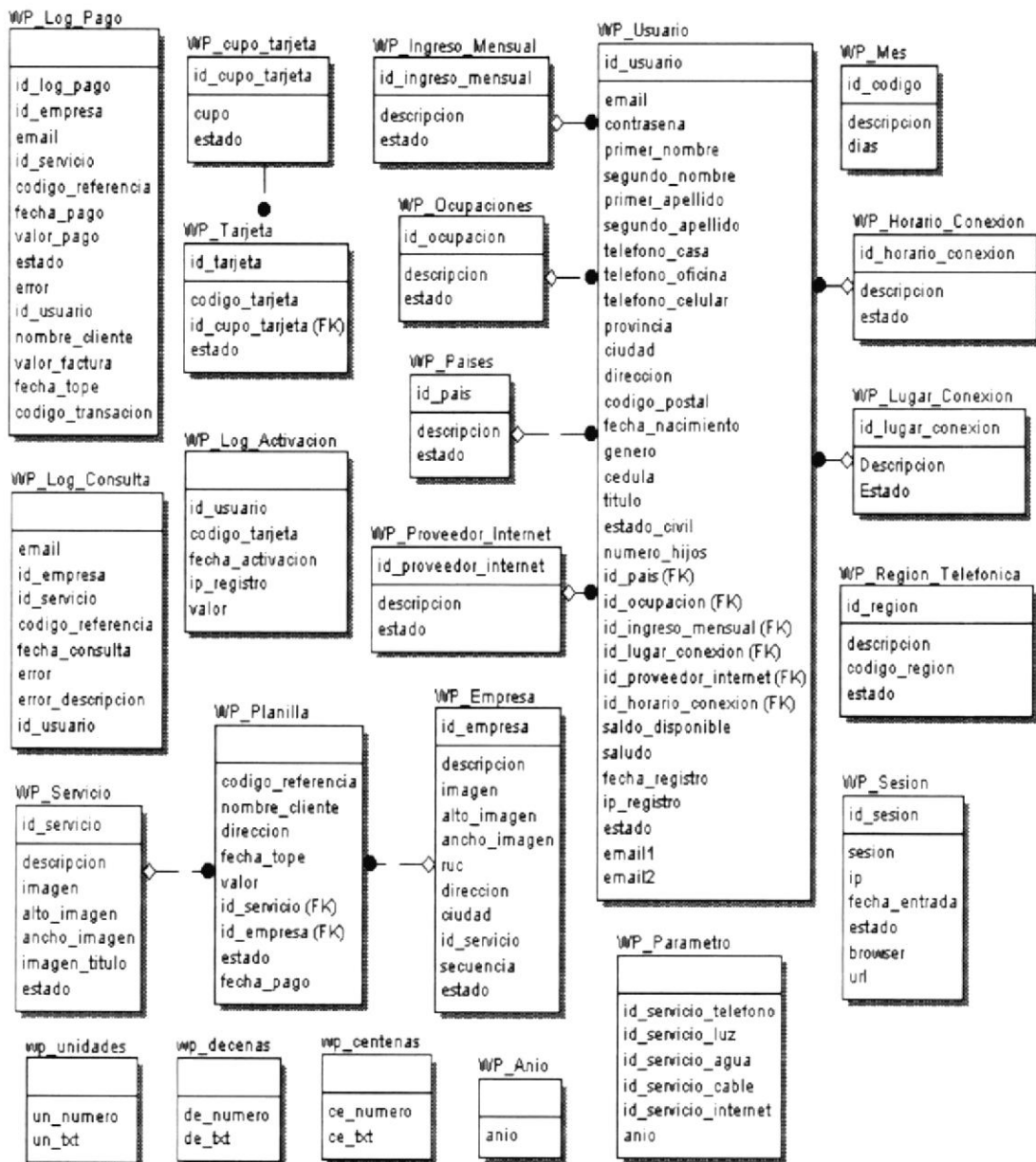
El software que se requiere es el siguiente:

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

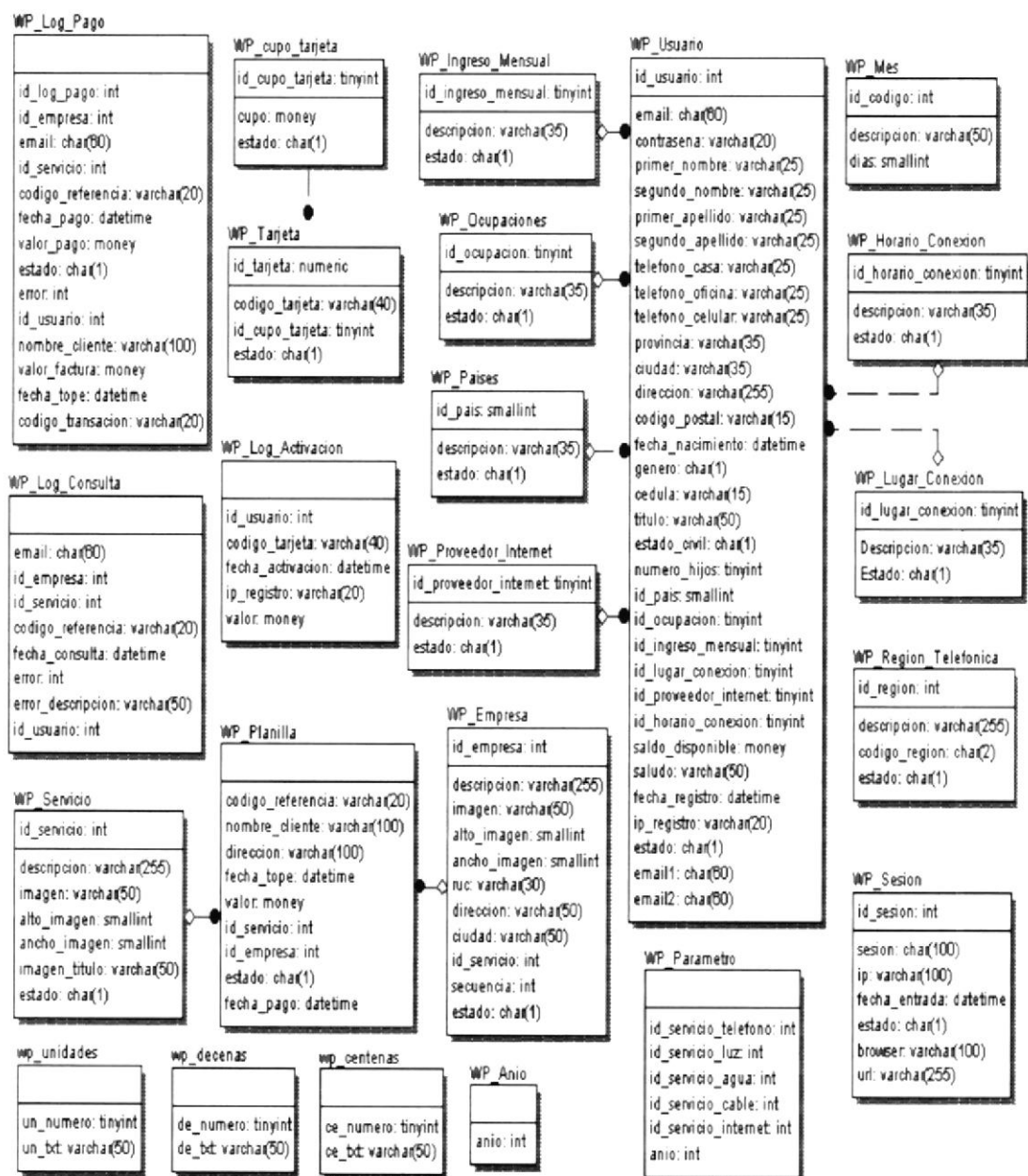
CAPITULO 5

5. IMPLEMENTACIÓN DE LA CAPA DE DATOS.

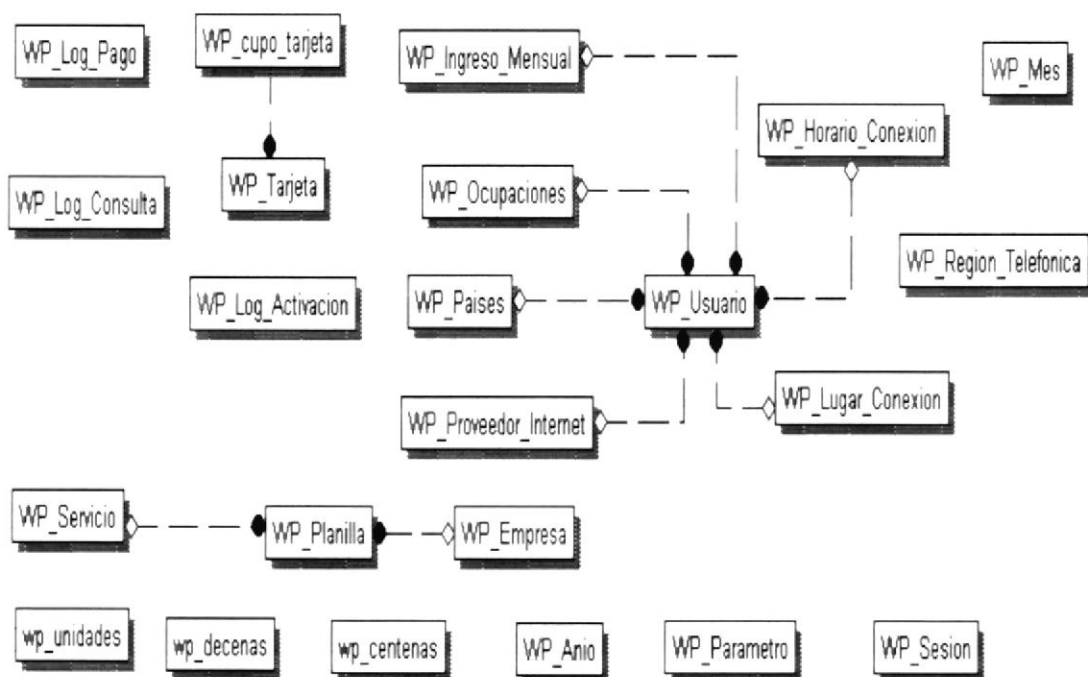
5.1 MODELO LÓGICO DE DATOS.



5.2 MODELO FÍSICO DE DATOS.





5.3 DIAGRAMA DE ENTIDAD – RELACIÓN.





5.4 PROCEDIMIENTOS ALMACENADOS.


A continuación se detalla una lista de Store Procedures que han sido utilizados en el Sitio Web WebPagos.

	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Cupo_Tarjeta	Parámetros: @p_codigo_tarjeta Varcha r(40).		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite conocer el cupo de una determinada tarjeta.			


 Escuela Superior Politécnica del Litoral		Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Aumentar_Saldo_Disponible	Parámetros: @p_Id_Usuario int @p_Cupo_Tarjeta money.		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite aumentar el saldo disponible de un usuario por la activación de una tarjeta.			


 Escuela Superior Politécnica del Litoral		Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_ContrasenaxUsuario	Parámetros: @p_email Varchar(60)		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite conocer solo datos principales de un usuario, como por ejemplo nombres, contraseñas e identificación de usuario.			


 Escuela Superior Politécnica del Litoral		Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_EmailUsuario	Parámetros: @p_email Varchar(60)		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite conocer toda la informacion de un usuario.			


 Escuela Superior Politécnica del Litoral		Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Empresa	Parámetros: @p_id_servicio int		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite conocer las empresas que estas relacionadas con un tipo específico de servicio.			


	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Historico	Parámetros: @p_id_empresa int, @p_mes_desde int, @p_mes_hasta int, @p_id_usuario int		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite conocer los pagos que han sido realizados por un determinado usuario en un rango de meses, ya sea para una empresa específica o para todas las empresas en las que dicho usuario haya efectuado pagos.			

	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Log_Pago	Parámetros: <p style="text-align: center;">@p_id_log_pago int, @p_Numero_Transaccion Varchar(20)</p>		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite obtener informacion de un pago realizado ya que cada vez que se efectúe un pago se guarda un registro en la tabla de Log de pagos.			


	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Numero_Transaccion	Parámetros: @p_id_servicio int, @p_id_empresa int		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite generar un número de transacción que servirá como identificador único de un pago realizado.			


	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Planilla	Parámetros: @p_id_empresa int, @p_id_servicio int, @p_codigo_referencia varchar(20)		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite obtener una planilla determinada ya sea que este pendiente de pago o ya haya sido cancelada, según la empresa, el servicio y el código de referencia.			


 Escuela Superior Politécnica del Litoral		Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Planilla_Fecha_Tope	Parámetros: @p_id_empresa int, @p_id_servicio int, @p_codigo_referencia varchar(20)		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite conocer si una planilla que aun no ha sido cancelada esta todavia en vigencia para ser cancelada en WebPagos.			

 Escuela Superior Politécnica del Litoral		Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Tarjeta_Especificas	Parámetros: @p_Codigo_Tarjeta varchar(40)		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite obtener el cupo de una tarjeta determinada.			

	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Consultar_Usuario_Saldo_Dispo nible	Parámetros: @p_id_usuario int, @p_valor money		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite conocer si un usuario su saldo disponible puede cubrir un determinado valor a pagar.			


	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Ingresar_Log_Activacion	Parámetros: @p_Id_Usuario int, @p_saldo_disponible Money, @p_codigo_tarjeta Varchar(40), @p_ip_registro varchar(20)		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite el ingreso de un registro al log de activaciones cada vez que un usuario active una tarjeta de prepago.			

	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Ingresar_Log_Pago	Parámetros: @p_id_servicio int, @p_id_empresa int, @p_codigo_referencia varchar(20), @p_email varchar(60), @p_valor_pago money, @p_nombre_cliente varchar(100), @p_fecha_tope datetime, @p_id_usuario int, @p_numero_transaccion varchar(20)		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite el ingreso de un registro al log de pagos cada vez que un usuario efectue una cancelación de alguna deuda.			

	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Ingresar_Usuario	Parámetros: <p> @p_email char(60), @p_contrasena varchar(20), @p_primer_nombre varchar(25), @p_segundo_nombre varchar(25), @p_primer_apellido varchar(25), @p_segundo_apellido varchar(25), @p_telefono_casa varchar(25), @p_telefono_celular varchar(25), @p_provincia varchar(35), @p_ciudad varchar(35), @p_direccion varchar(255), @p_codigo_postal varchar(15), @p_fecha_nacimiento varchar(10), @p_genero char(1), @p_cedula varchar(15), @p_titulo varchar(50), @p_estado_civil char(1), @p_id_pais smallint, @p_id_ocupacion tinyint, @p_id_ingreso_mensual tinyint, @p_id_lugar_conexion tinyint, @p_id_proveedor_internet tinyint, @p_id_horario_conexion tinyint, @p_saludo varchar(50), @p_ip_registro varchar(20), @p_estado char(1) </p>		
Descripción de Procedimiento Almacenado: <p>✓ Este procedimiento permite el ingreso de un registro a la tabla de Usuarios cada vez que un nuevo usuario se registre.</p>			

	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Modificar_Empresa_Secuencia	Parámetros: @p_id_servicio int, @p_id_empresa int		
Descripción de Procedimiento Almacenado: <p>✓ Este procedimiento permite la actualización de una secuencia la cual que sirve para tener un control del numero de transaccion de un pago, ya cada vez que se efectúe un pago se debe actualizar con la nueva secuencia disponible para la siguiente transacción. Esta secuencia es manejada siempre por empresa.</p>			

	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Modificar_Planilla_Pagada	Parámetros: @p_codigo_referencia varchar(20), @p_id_servicio int, @p_id_empresa int		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite la actualización del estado de una planilla de Normal a Pagada, en el momento que se efectúe un pago.			

	Escuela Superior Politécnica del Litoral	Descripción de Store Procedures	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: SP_Modificar_Usuario_SaldoDiponi ble	Parámetros: @p_id_usuario int, @p_valor money		
Descripción de Procedimiento Almacenado: ✓ Este procedimiento permite la actualización del saldo disponible que tiene un usuario, ya que cada vez que se efectúe un pago este saldo disponible disminuye.			

5.5 PROCESO SERVIDOR DE BASE DE DATOS.

El más fundamental de todos recibe requerimientos de los clientes (típicamente en lenguaje SQL) a los que devuelve el resultado. Es quien implementa finalmente la inserción, modificación, eliminación y consulta de los datos.

En WebPagos utilizamos Microsoft SQL Server 7.0 como manejador de base de datos.

CAPITULO 6

6. IMPLEMENTACIÓN DE LA CAPA DE NEGOCIOS.

6.1 ESTRUCTURA DE LOS COMPONENTES DE ACCESO A DATOS.

En el sistema de WebPagos utilizamos un componente central de acceso a datos, llamado “CD_webPago.dll”, el cual es el contenedor de todas las clases que se encargan de acceder directamente a la base de datos.

6.2 OBJETOS DISTRIBUIDOS DE ACCESO A DATOS.

Las clases por las que está compuesto el componente “CD_webPago.dll” son:

- ✓ CLSDConversionLetras
- ✓ CISDEmpresa
- ✓ CLSDHorarioConexion
- ✓ CLSDIngresosMensuales
- ✓ CLSDLogActivacion
- ✓ CLSDLogPago
- ✓ CLSDLogSesion
- ✓ CLSDLugaresConexion
- ✓ CLSDMeses
- ✓ CISDOcupaciones

- ✓ CLSDPaíses
- ✓ CLSDParametro
- ✓ CLSDPlanilla
- ✓ CLSDProveedorInternet
- ✓ CLSDRegionTelefonica
- ✓ CLSDTarjeta
- ✓ CLSDUsuario

Cada una de las clases mencionadas se definen los siguientes métodos:

- ✓ Ingresar.
- ✓ Modificar.
- ✓ Eliminar.
- ✓ ConsultarId.
- ✓ Consultar.

6.3 COMUNICACIÓN CON EL PROCESO SERVIDOR DE BASE DE DATOS.

Cabe recalcar que las clases que intervienen en el componente de negocio “CN_webPago” tienen el mismo nombre tanto lógico y físico que las clases del componente de datos “CD_webPago”.


RELACIÓN	
COMPONENTE DE NEGOCIO – COMPONENTE DE DATOS	
CLSNConversionLetras	CLSDConversionLetras
CLSNEmpresa	CLSDEmpresa
CLSNHorarioConexion	CLSDHorarioConexion
CLSNIngresosMensuales	CLSDIngresosMensuales
CLSNLogPago	CLSDLogPago
CLSNLogSesion	CLSDLogSesion
CLSNLugaresConexion	CLSDLugaresConexion
CLSNMeses	CLSDMeses
CLSNOcupaciones	CLSDOcupaciones
CLSNPaises	CLSDPaises
CLSNParametro	CLSDParametro
CLSNPlanilla	CLSDPlanilla CLSDUsuario CLSDEmpresa CLSDLogPago
CLSNProveedoresInternet	CLSDProveedoresInternet
CLSNRegionTelefonica	CLSDRegionTelefonica
CLSNTarjeta	CLSDTarjeta CLSDUsuario CLSDLogActivacion
CLSNUsuario	CLSDUsuario


6.4 ESTRUCTURA DE LOS COMPONENTES DE NEGOCIOS.


En WebPagos se realizó la agrupación de las clases de negocios en un solo componente llamado “CN_webPago”.


6.5 OBJETOS DISTRIBUIDOS DE NEGOCIOS.


El componente “CN_webPago” esta conformado por las siguientes clases:


	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNConversionLetras.			
Descripción del Objeto de Negocio: ✓ Permite realizar la conversión de dígitos numéricos a letras			


	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: <p style="text-align: center;">CLSNEmpresa</p>			
Descripción del Objeto de Negocio: <p>✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Empresa</p>			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: <p style="text-align: center;">CLSNHorarioConexion</p>			
Descripción del Objeto de Negocio: <p>✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Horario_Conexión</p>			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNIngresosMensuales			
Descripción del Objeto de Negocio: ✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Ingreso_Mensual			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNLogPago			
Descripción del Objeto de Negocio: ✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Log_Pago			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNLogSesion			
Descripción del Objeto de Negocio: ✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Log_Sesion			


	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNLugaresConexion			
Descripción del Objeto de Negocio: ✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Lugar_Conexión			


	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNMeses			
Descripción del Objeto de Negocio: <input checked="" type="checkbox"/> Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Mes			


	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNOcupaciones			
Descripción del Objeto de Negocio: <input checked="" type="checkbox"/> Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Ocupaciones			


	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNPaises			
Descripción del Objeto de Negocio: ✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Paises			


	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina Parrales	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNParametro			
Descripción del Objeto de Negocio: ✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Parametro			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: <p style="text-align: center;">CLSNPlanilla</p>			
Descripción del Objeto de Negocio: <ul style="list-style-type: none"> ✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, de los datos de la tabla WP_Planilla ✓ Realiza el proceso de pago de planilla de servicios y actualiza las siguientes tablas WP_Planilla, WP_Usuario, WP_Empresa, WP_LogPago 			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNProveedoresInternet			
Descripción del Objeto de Negocio: <input checked="" type="checkbox"/> Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Proveedor_Internet			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: CLSNRegionTelefonica			
Descripción del Objeto de Negocio: <input checked="" type="checkbox"/> Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_Region_Telefonica			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: <p style="text-align: center;">CLSNTarjeta</p>			
Descripción del Objeto de Negocio: <p>✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_ Tarjeta</p>			

	Escuela Superior Politécnica del Litoral	Descripción de Componentes de Negocios	Fecha de Elaboración: 5/Julio/2002
Autor: LSI-ESPOL Integrantes: Dayanara Altamirano Nancy Castro Katrina PARRALES	Proyecto de Sitio Web para Licenciatura de Sistemas de Información	Versión 1.0	
ID: <p style="text-align: center;">CLSNUusuario</p>			
Descripción del Objeto de Negocio: <p>✓ Accesa al objeto de datos del mismo nombre para realizar el ingreso, consulta, modificación y eliminación de los datos de la tabla WP_ Usuario</p>			

IDL Generado

```

[
    odl,
    uuid(66C46F5F-720A-4131-9601-946B8CA6BB06),
    version(1.0),
    hidden,
    dual,
    nonextensible,
    oleautomation
]

interface _CLSNusuario : IDispatch {
    [id(0x6003000f)]
    HRESULT Verificar_Contrasena(
        [in] BSTR p_email,
        [in] BSTR p_contrasena,
        [out, retval] VARIANT_BOOL* );
    [id(0x60030010)]
    HRESULT Encriptador(
        [in, out] BSTR* Contrasena,
        [out, retval] BSTR* );
    [id(0x60030011)]
    HRESULT Ingresar_Usuario(
        [in] BSTR p_email,
        [in] BSTR p_contrasena,

```


[in] BSTR p_primer_nombre,
[in] BSTR p_segundo_nombre,
[in] BSTR p_primer_apellido,
[in] BSTR p_segundo_apellido,
[in] BSTR p_telefono_casa,
[in] BSTR p_telefono_oficina,
[in] BSTR p_telefono_celular,
[in] BSTR p_provincia,
[in] BSTR p_ciudad,
[in] BSTR p_direccion,
[in] BSTR p_codigo_postal,
[in] BSTR p_fecha_nacimiento,
[in] BSTR p_genero,
[in] BSTR p_cedula,
[in] BSTR p_titulo,
[in] BSTR p_estado_civil,
[in] short p_numero_hijos,
[in] short p_id_pais,
[in] short p_id_ocupacion,
[in] short p_id_ingreso_mensual,
[in] short p_id_lugar_conexion,
[in] short p_id_proveedor_internet,
[in] short p_id_horario_conexion,
[in] BSTR p_saludo,

[in] BSTR p_ip_registro,
[in] BSTR p_estado,
[out, retval] VARIANT_BOOL*);

[id(0x60030012)]

HRESULT ConsultarIdUsuario(
[in] long p_Id_Usuario,
[out, retval] _Recordset**);

[id(0x60030013)]

HRESULT Consultar_Usuario(
[in] BSTR p_email,
[out, retval] _Recordset**);

[id(0x60030014)]

HRESULT Modificar_Usuario(
[in] BSTR p_Id_Usuario,
[in] BSTR p_email,
[in] BSTR p_contrasena,
[in] BSTR p_primer_nombre,
[in] BSTR p_segundo_nombre,
[in] BSTR p_primer_apellido,
[in] BSTR p_segundo_apellido,
[in] BSTR p_telefono_casa,
[in] BSTR p_telefono_oficina,
[in] BSTR p_telefono_celular,
[in] BSTR p_provincia,

[in] BSTR p_ciudad,
 [in] BSTR p_direccion,
 [in] BSTR p_codigo_postal,
 [in] BSTR p_fecha_nacimiento,
 [in] BSTR p_genero,
 [in] BSTR p_cedula,
 [in] BSTR p_titulo,
 [in] BSTR p_estado_civil,
 [in] short p_numero_hijos,
 [in] short p_id_pais,
 [in] short p_id_ocupacion,
 [in] short p_id_ingreso_mensual,
 [in] short p_id_lugar_conexion,
 [in] short p_id_proveedor_internet,
 [in] short p_id_horario_conexion,
 [in] BSTR p_saludo,
 [in] BSTR p_ip_registro,
 [in] BSTR p_estado,
 [out, retval] VARIANT_BOOL*);

[id(0x60030015)]

HRESULT Modificar_Usuario_Saldo_Disponible(

[in] short p_Id_Usuario,
 [in] CURRENCY p_valor,
 [out, retval] VARIANT_BOOL*);

```
};
```

```
[
```

```
odl,
```

```
uuid(21E1DB42-DF76-4528-BF45-7BFAD5317B84),
```

```
version(1.0),
```

```
hidden,
```

```
dual,
```

```
nonextensible,
```

```
oleautomation
```

```
]
```

```
interface _CLSNOcupaciones : IDispatch {
```

```
    [id(0x60030002)]
```

```
    HRESULT Consultar_Ocupaciones([out, retval] _Recordset** );
```

```
};
```

```
[
```

```
odl,
```

```
uuid(07AD8DA0-5246-458A-9F2F-2AAF710B80E3),
```

```
version(1.0),
```

```
hidden,
```

```
dual,
```

```
nonextensible,
```

```
oleautomation
```

```

]

interface _CLSNProvedoresInternet : IDispatch {

    [id(0x60030002)]

    HRESULT Consultar_ProvedoresInternet([out, retval] _Recordset** );

};

[

    odl,

    uuid(F00007E8-C0E9-4B2A-93BE-94282124D336),

    version(1.1),

    hidden,

    dual,

    nonextensible,

    oleautomation

]

interface _CLSNPaises : IDispatch {

    [id(0x60030003)]

    HRESULT Ingresar_Pais(

        [in] BSTR p_descripcion,

        [in] BSTR p_estado,

        [out, retval] VARIANT_BOOL* );

    [id(0x60030002)]

    HRESULT Consultar_Paises([out, retval] _Recordset** );

};

```

```

[
    odl,
    uuid(5FCB14C1-36C1-4A59-B706-F63517924809),
    version(1.0),
    hidden,
    dual,
    nonextensible,
    oleautomation
]

interface _CLSNHorarioConexion : IDispatch {
    [id(0x60030002)]
    HRESULT Consultar_HorarioConexion([out, retval] _Recordset** );
};

[
    odl,
    uuid(FEA09DE0-ABCF-4F36-A3FC-6EA1A4C3A2B6),
    version(1.0),
    hidden,
    dual,
    nonextensible,
    oleautomation
]

interface _CLSNIngresosMensuales : IDispatch {

```

```

[id(0x60030002)]
HRESULT Consultar_IngresosMensuales([out, retval] _Recordset** );
};

[
odl,
uuid(1D58BA11-5C64-46B9-843B-75B6A8CC60EB),
version(1.0),
hidden,
dual,
nonextensible,
oleautomation
]

interface _CLSNLugaresConexion : IDispatch {
[id(0x60030002)]
HRESULT Consultar_LugaresConexion([out, retval] _Recordset** );
};

[
odl,
uuid(37F1C18B-7B59-464F-8D08-2B89DE5B3189),
version(1.0),
hidden,
dual,

```

```

nonextensible,
oleautomation
]
interface _CLSNPlanilla : IDispatch {
    [id(0x60030004)]
    HRESULT Consultar_Planilla(
        [in] short p_id_empresa,
        [in] short p_id_servicio,
        [in] BSTR p_codigo_referencia,
        [out, retval] _Recordset** );
    [id(0x60030005)]
    HRESULT Pagar_Planilla(
        [in] short p_id_servicio,
        [in] short p_id_empresa,
        [in] BSTR p_codigo_referencia,
        [in] BSTR p_email,
        [out, retval] long* );
};

[
    odl,
    uuid(F9E1ED9A-2657-489B-87AA-C35F20950AB6),
    version(1.0),

```



```
hidden,  
dual,  
nonextensible,  
oleautomation  
]  
  
interface _CLSNRegionTelefonica : IDispatch {  
    [id(0x60030002)]  
    HRESULT Consultar_RegionTelefonica([out, retval] _Recordset** );  
};  
  
[  
    odl,  
    uuid(E5644C22-87BA-4950-8C2D-2CC6D3FD6F7F),  
    version(1.0),  
    hidden,  
    dual,  
    nonextensible,  
    oleautomation  
]  
  
interface _CLSNLogSesion : IDispatch {  
    [id(0x60030002)]  
    HRESULT Ingresar_LogSesion(
```

```

[in, out] BSTR* p_sesion,
[in, out] BSTR* p_ip,
[in, out] BSTR* p_browser,
[in, out] BSTR* p_url,
[out, retval] VARIANT_BOOL* );
};

[
odl,
uuid(1AA11FFC-47CA-4C96-B79C-B3BCDF56A51F),
version(1.0),
hidden,
dual,
nonextensible,
oleautomation
]
interface _CLSNParametro : IDispatch {
[id(0x60030002)]
HRESULT Consultar_Parametro([out, retval] _Recordset** );
};

[
odl,

```

```
uuid(74D6AB34-3DF5-4060-BE1C-F1F2B3C7EA8F),  
version(1.0),  
hidden,  
dual,  
nonextensible,  
oleautomation  
]  
  
interface _CLSNMeses : IDispatch {  
    [id(0x60030002)]  
    HRESULT Consultar_Meses([out, retval] _Recordset** );  
};  
  
[  
    odl,  
    uuid(C8240CD4-B0EF-4A5D-BA59-AB6A095B2337),  
    version(1.0),  
    hidden,  
    dual,  
    nonextensible,  
    oleautomation  
]  
  
interface _CLSNTarjeta : IDispatch {  
    [id(0x60030003)]
```

```

HRESULT Consultar_Tarjeta(
    [in] BSTR p_Codigo_Tarjeta,
    [out, retval] _Recordset** );
[id(0x60030004)]

HRESULT Activar_Tarjeta(
    [in] short p_Id_Usuario,
    [in] BSTR p_Codigo_Tarjeta,
    [in] CURRENCY p_Cupo_Tarjeta,
    [in] BSTR p_ip_registro,
    [out, retval] CURRENCY* );
};

[
    odl,
    uuid(98C92E4E-553C-4E03-9134-03823F329C7F),
    version(1.0),
    hidden,
    dual,
    nonextensible,
    oleautomation
]

interface _CLSNLogPago : IDispatch {
    [id(0x60030001)]

```

```

HRESULT Consultar_Historico(
    [in] short p_id_empresa,
    [in] short p_mes_desde,
    [in] short p_mes_hasta,
    [in] short p_Id_Usuario,
    [out, retval] _Recordset** );

[id(0x60030002)]

HRESULT Consultar_Log_Pago(
    [in] short p_id_Log_Pago,
    [in] BSTR p_Numero_Transaccion,
    [out, retval] _Recordset** );

};

[
    odl,
    uuid(593DF928-01AA-479B-81DF-E6ED92254844),
    version(1.0),
    hidden,
    dual,
    nonextensible,
    oleautomation
]

interface _CLSNEmpresa : IDispatch {

```

```

[id(0x60030001)]
HRESULT Consultar_Empresas(
    [in] short p_id_servicio,
    [out, retval] _Recordset** );
};

[
    odl,
    uuid(5963555D-7B00-45FC-9774-3192A7371087),
    version(1.0),
    hidden,
    dual,
    nonextensible,
    oleautomation
]

interface _CLSNConversionLetras : IDispatch {
    [id(0x60030000)]
    HRESULT Consultar_Numero(
        [in, out] BSTR* t_debug,
        [in, out] CURRENCY* i_dinero,
        [in, out] short* i_moneda,
        [out, retval] BSTR* );
};

```

6.6 COMUNICACIÓN ENTRE OBJETOS DISTRIBUIDOS.

Se ha utilizado la sentencia `CreateInstance` para crear instancias de los objetos COM, el Objeto solo podrá manejar transaccionalidad si el componente esta registrado en el MTS o COM+.

Cuando se crea el objeto utilizando `CreateInstance`, el nuevo objeto se deriva del `ObjectContext`. El nuevo objeto siempre se ejecuta en la misma actividad que el objeto que lo creo. Si el objeto actual tiene una transacción, el atributo de la transacción del nuevo componente de objeto determina si va o no el nuevo objeto a ejecutarse en el scope de esa transacción, el atributo del componente de la transacción debe tener `Requires Transaction or Support Transaction` para soportar transaccionalidad.

6.7 PROCESO SERVIDOR TRANSACCIONAL.

En `WebPagos` se manejan las siguientes transacciones:

- ✓ **Pagar Planilla.**- La transacción se inicia consultando si un usuario posee saldo disponible en su cuenta, para luego proceder a la disminución del valor a pagar de este saldo. Después de ser debitado el valor del Saldo se procede a cambiar el estado de pagado en la tabla de planilla. A continuación se actualiza la secuencia de número de transacción asociado a la empresa para finalmente registrar la cancelación en el log de pagos. Si ocurriera algún error

en el transcurso de ejecución del proceso automáticamente COM+ procede al reverso dejándolo en su estado inicial.

Los objetos involucrados en la transacción son los siguientes:

CLASE DE NEGOCIO	CLASE DE DATOS
CLSNPlanilla.- Pagar_Planilla	CLSDUsuario.- Consultar y Modificar CLSDPlanilla.- Modificar CLSEmpresa.- Consultar y Modificar CLSDLogPago.- Ingresar

- ✓ **Activar Tarjeta.**- La transacción se inicia actualizando el estado de la planilla a estado de activada, para luego actualizar el saldo disponible del usuario y finalmente guardar el registro correspondiente en el log de activación de tarjetas. Si ocurriera algún error en el transcurso de ejecución del proceso automáticamente COM+ procede al reverso dejándolo en su estado inicial.

Los objetos involucrados en la transacción son los siguientes:

CLASE DE NEGOCIO	CLASE DE DATOS
CLSNTarjeta.- Activar_Tarjeta	CLSDTarjeta.- Modificar CLSDUsuario.- Modificar CLSDLogActivacion.- Ingresar

- ✓ **Registro Usuario.**- La Transacción involucra el ingreso de datos del usuario a WebPagos. Si ocurriera algún error en el transcurso de ejecución del proceso automáticamente COM+ procede al reverso dejándolo en su estado inicial.

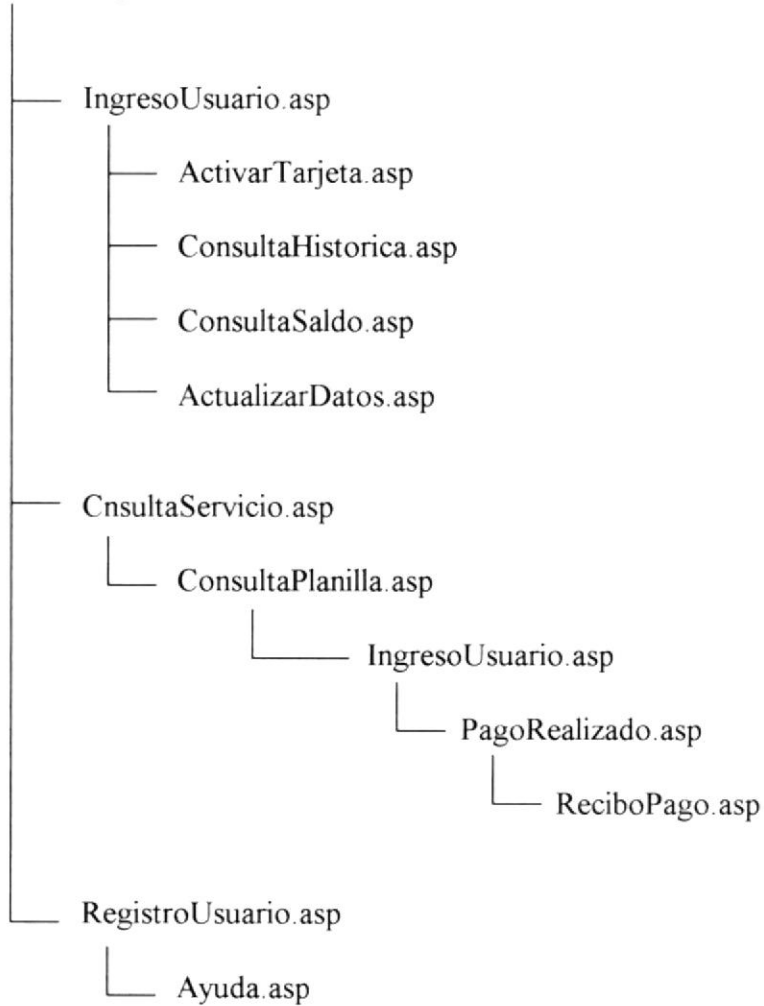
CLASE DE NEGOCIO	CLASE DE DATOS
CLSUsuario.- Ingresar_Usuario	CLSDUsuario – Ingresar

CAPITULO 7

7. IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN.

7.1 DISEÑO DEL SITIO WEB – CLIENTE BASADO EN BROWSER.

Main.asp



7.2 CLIENTE STAND-ALONE.

WebPagos ha implementado un sistema stand-alone utilizando como herramienta de desarrollo a Microsoft Visual Basic 6.0, dentro del cual se maneja la parte administrativa del sitio, como el mantenimiento de las tablas tipo o tablas de parametrización, además de los procesos batch.

Dentro de la sección mantenimiento existen las siguientes opciones:

- ✓ Empresa
- ✓ HorarioConexion
- ✓ IngresosMensuales
- ✓ LugaresConexion
- ✓ Ocupaciones
- ✓ Paises
- ✓ ProveedorInternet
- ✓ RegionTelefonica

Los procesos son los siguientes:

- ✓ Carga de planillas de deudas al sitio por medio de archivos planos.
- ✓ Generación de Tarjeta de Prepago.
- ✓ Generación de archivos de pagos diarios para empresas proveedoras del servicio.

7.3 CLIENTE HTML.

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

7.4 CLIENTE SCRIPT-DHTML.

En WebPagos se implementó JavaScript, para todas las validaciones que se ejecutan del lado del cliente, ya que es el mas recomendado porque no tiene problemas al ser ejecutado desde cualquier tipo de browser.

7.5 SERVER SCRIPT - ASP.

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

Las Active Server Pages son un ambiente de aplicación abierto en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web.

7.6 CSS.

En el Sitio Web se utilizó hojas de estilo para dar formato de presentación en todas las páginas, el css fue aplicado en cabeceras, textos, cajas de texto, etc.

De esta forma al querer cambiar el formato de las páginas solo se modificará el archivo de hojas de estilos.

7.7 COMUNICACIÓN CON COMPONENTES DE NEGOCIO.

En la implementación del sitio web se desarrollaron las siguientes páginas asp con su respectiva relación con componetes de negocio:

RELACIÓN	
PÁGINA ASP – COMPONENTE DE NEGOCIOS	
ActivarTarjeta.asp	CLSNTarjeta
CheckEmail.asp	CLSNUusuario
ConsultaHistorico.asp	CLSLogPago CLSNEmpresa CLSNMeses
ConsultaPlanilla.asp	CLSNPlanilla
ConsultaSaldo.asp	CLSNUusuario CLSNConversionLetras
ConsultaServicio.asp	CLSNRegionTelefonica
IngresoUsuario.asp	CLSNUusuario CLSNPlanilla
ReciboPago.asp	CLSLogPago CLSNConversionLetras
RegistroUsuario.asp	CLSNUusuario CLSNPaises CLSNocupaciones CLSNIngresosMensuales CLSNLugaresConexion CLSNProveedoresInternet CLSNHorarioConexion

7.8 PROCESO SERVIDOR WEB.

El proceso servidor web de WebPagos es el Internet Information Server. Este permite el desarrollo de aplicaciones de negocio basadas en Web que se pueden extender a Internet o distribuir en intranets corporativas. Internet Information Server, introduce un nuevo concepto en Internet; las aplicaciones transaccionales. Las transacciones son los conductos que hacen posible ejecutar aplicaciones reales de negocio con desarrollos rápidos, escalables y fiables.

Lo hemos configurado de la siguiente manera:

Se creó un sitio virtual con el mismo nombre del aplicativo, en donde el documento principal a ejecutarse es main.asp y en segundo lugar es global.asa.

CONCLUSIONES

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

En el desarrollo del sitio WebPagos, nuestro propósito es cumplir con todas las expectativas planteadas a lo largo de este documento.

RECOMENDACIONES

Entre las recomendaciones más importantes podemos enumerar las siguientes:

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

BIBLIOGRAFÍA

1. <http://www.mis-pagos.com>
2. <http://www.mispagosaldia.com>
4. <http://www.monografias.com>
5. <http://www.microsoft.com>
6. <http://www.arsys.com.es>
7. <http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/cap0307.htm>
8. <http://www.ii.uam.es/~amartine/si2/ejercic/si2temas.htm>
9. <http://www.dcti.uninet.edu/estudios.html>
10. <http://ti.cicei.ulpgc.es/master/proacmaster.html>

APENDICES

APENDICE A.- ARQUITECTURA C/S

CONCEPTOS

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"), auditoría 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 pueden destacar las siguientes:

- ✓ El servidor presenta a todos sus clientes una interface única y bien definida.
- ✓ El cliente no necesita conocer la lógica del servidor, sólo su interface 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.

Procesos Servidores

Varios tipos de servidores especializados:

File Servers

- ✓ El más elemental de todos
- ✓ Sólo provee el servicio de dar acceso a archivos
 - Datos comunes a todos los clientes, pero la aplicación no está subdividida
- ✓ Si bien el concepto de "computador dedicado" a la tarea de file server (NetWare) ya no es aplicable, el servicio file server sigue y seguirá siendo implementado en los sistemas operativos

DataBase Servers

- ✓ El más fundamental de todos
- ✓ Recibe requerimientos de los clientes(típicamente en lenguaje SQL) a los que devuelve el resultado
- ✓ Es quien implementa finalmente la inserción, modificación, eliminación y consulta de los datos
- ✓ En ciertos casos participa dentro de la arquitectura junto con un proceso intermediario
 - Pasaría de ser 2-Tier a 3-Tier o incluso n-Tier

Transaction Processing Server

- ✓ Concepto heredado del mundo de los Mainframe
- ✓ Mediador entre procesos clientes y servidores
- ✓ Administrador de Transacciones
- ✓ Proceso que permite ejecutar otros procesos

Consta de 2 Categorías:

- ✓ TP Lite
 - DataBase Server con stored procedures que implementen transacciones
 - 2-and-a-half-Tier
- ✓ TP Heavy
 - Basados en TP Monitors
 - 3-Tier, n-Tier

GroupWare Server

- ✓ Orientado a mejorar la comunicación entre grupos de personas
 - Mail, grupos de discusión, workflow, etc
- ✓ Permiten el manejo de información no estructurada
 - Web Server
- ✓ Recibe requerimientos HTTP del proceso cliente (Web Browser)
- ✓ Retorna al cliente la respuesta en código HTML
 - HTML puede residir de manera estática
 - HTML puede ser generado dinámicamente

- ✓ Han llevado la tecnología C/S a otra dimensión
 - Son la más clara prueba de que los sistemas distribuidos son posibles y necesarios
 - Object Server
- ✓ Es la nueva tendencia en el desarrollo de aplicaciones C/S
 - Facilitan el desarrollo de Sistemas Distribuidos con esquemas 3-Tier y n-Tier
- ✓ Basado en el concepto de ORB (Object Request Broker)

Están sujetos a 2 arquitecturas básicas:

- ✓ CORBA (Common ORB Architecture)
- ✓ DCOM (Distributed Component Object Model)

Procesos Clientes

- ✓ Clientes Independientes (standalone)
 - No necesitan de un contenedor para ser ejecutados
 - Emplea API propio del sistema operativo
- ✓ Clientes Dependientes
 - Sólo pueden ser ejecutados "dentro" de un contenedor
 - La dependencia genera interoperabilidad

APENDICE B.- PROGRAMACIÓN EN EL WEB.

HTML

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

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

Las marcas del lenguaje HTML especifican:

- ✓ La estructura lógica del documento:

- Cabeceras, tipos y tamaños de las fuentes.
 - Párrafos de texto.
 - Centrado.
 - Enumeraciones o listas.
 - Formularios.
 - Tablas.
- ✓ Distintos estilos que definen el texto:
- Negrita.
 - Cursiva.
 - Diferentes efectos: (direcciones de correo, citas textuales, etc...).
- ✓ Inclusión de hipertextos para acceder a otros documentos relacionados
- ✓ Inclusión de imágenes y ficheros multimedia.

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

DHTML

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

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

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

CSS

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

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

JAVA SCRIPT

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

Tanto en el Netscape 4.0 como en el Explorer 4.0, se introdujeron nuevas extensiones que ofrecían a los creadores de páginas web una forma de trabajar con el DHTML. De cualquier manera, estas extensiones no se estandarizaron antes de

la aparición de estas versiones de los dos navegadores. Por este motivo, ahora tenemos dos versiones tan incompatibles de JavaScript

APENDICE C.- COM/DCOM/COM+ COM

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

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

Con el modelo Microsoft Windows DNA, los componentes simplifican la complejidad de la construcción de aplicaciones multi-capa. Las aplicaciones

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

DCOM

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

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

El DCOM también puede funcionar dentro de una empresa o en redes distintas de la Internet pública. Usa protocolos TCP/IP y HTTP. El DCOM viene como parte de NT 4.0 y es una actualización gratuita para Windows 95. El DCOM está ya, o pronto lo estará, disponible en todas las principales plataformas UNIX y en los productos de servidores grandes de IBM. El DCOM reemplaza la Automatización Remota del OLE (OLE Remote Automation).

El DCOM es en lo general equivalente a una Arquitectura de Intermediación de Solicitud de Objetos Comunes (Common Object Request Broker Architecture, CORBA) por cuanto proporciona un juego de servicios distribuidos. El DCOM es la aproximación de Microsoft a un ambiente de programas y objetos de datos para toda una red. La CORBA es patrocinada por el resto de industria de tecnología de la información bajo los auspicios del Grupo de Administración de Objetos (Object Management Group OMG).

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 in-dirección 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+

✓ 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.

✓ Servicio de eventos COM+

Algunas aplicaciones de varios niveles requieren que determinados usuarios o aplicaciones reciban notificaciones de eventos interesantes como, por ejemplo, el contrato de un nuevo empleado, la modificación en el precio de la mercancía o la falta de existencias en el nivel de inventario donde se necesita un reaprovisionamiento. Los eventos COM+ proporcionan un servicio para el envío y entrega de notificaciones de eventos entre las aplicaciones.

En el modelo de eventos COM+, aquellas aplicaciones que envían notificaciones de eventos se denominan editores, mientras que las aplicaciones que reciben las notificaciones de evento se llaman suscriptores. Normalmente a los eventos COM+ se les conoce como LCE (eventos poco asociados) debido a que los editores y suscriptores no se conocen entre sí. En cambio, un conjunto de eventos se define dentro de una clase evento, que se encuentra instalada en una aplicación COM+. Los editores y suscriptores conocen las clases de eventos, sin embargo, éstos no se conocen entre sí.

La separación de los editores respecto de los suscriptores ofrece algunas ventajas. No es preciso controlar quién recibe los mensajes cuando se crea una aplicación

de editor, por lo que se escribe menos código. No es preciso modificar la aplicación del editor si se desea agregar o eliminar suscriptores una vez incluidos en la producción. Es más, tampoco se tienen que modificar los suscriptores al agregar o eliminar los editores asociados a una clase evento.

A continuación, se analiza brevemente el funcionamiento de este modelo de eventos. Primero se crea una clase evento que implemente una o varias interfaces. Cada interfaz define un grupo de métodos que representa un conjunto de eventos. Para implementar estas interfaces y responder a los eventos cuando se activan se escriben componentes de suscriptores. Finalmente, se escribe una aplicación del editor para crear objetos a partir de la clase evento y llamar a varios métodos cuando ésta desea activar eventos. El servicio de eventos se ocupa del procesamiento del método y de la entrega de eventos a cada suscriptor.

Existen dos tipos de suscripciones: suscripciones persistentes y suscripciones temporales. Cuando se entrega un evento a una suscripción persistente, el servicio de eventos COM+ crea un nuevo objeto a partir de un componente suscriptor configurado y solicita el método asociado al evento. El servicio de eventos activa la aplicación suscriptor si aún no se está ejecutando. Para una suscripción persistente, se crea y se destruye un objeto cada vez que se activa un evento.

El segundo tipo especificado es una suscripción temporal. Una vez que se ejecuta una aplicación, ésta puede registrarse a sí misma con una clase evento específica para recibir los eventos cuando se activan. En este caso, el servicio de eventos no

crea ni destruye un objeto para cada evento. La principal diferencia que existe entre los dos tipos de suscripciones radica en que el suscriptor temporal debe crear una nueva suscripción mediante la programación y registrar una devolución de llamada para recibir los eventos.

La creación de un suscriptor temporal requiere más código que la de un suscriptor persistente. La creación de este último resulta más sencilla porque se pueden utilizar las herramientas de administración de COM+. Si se reinicia el servidor que contiene la clase evento, los suscriptores persistentes continuarán en funcionamiento sin ningún problema. No obstante, todos los suscriptores temporales deberán crear y registrar una nueva suscripción.

Una arquitectura gradual de notificación de eventos requiere una comunicación asíncrona. Si el editor espera que cada suscriptor procese su notificación de eventos, el sistema podría resultar inadecuado. Asimismo, algunas aplicaciones puede que necesiten que se envíen notificaciones de eventos sin una conexión consistente con MSMQ. Los eventos COM+ se han integrado totalmente con los componentes en cola para proporcionar las ventajas de una comunicación asíncrona y sin conexión. Tanto los editores como los suscriptores se pueden configurar para utilizar componentes en cola en lugar de RPC. Para muchas aplicaciones, los eventos COM+ sólo resultan útiles cuando se utilizan con el servicio de componentes en cola.

APENDICE D.- IIS Y ASP

INTRODUCCIÓN DE IIS Y ASP

Estos servicios, integrados en la plataforma de aplicaciones Microsoft, permiten un enlace de alto rendimiento con la capa de presentación. El producto Microsoft Internet Information Server (IIS) permite el desarrollo de aplicaciones de negocio basadas en Web que se pueden extender a Internet o distribuir en intranets corporativas. Con Internet Information Server, Microsoft introduce un nuevo concepto en Internet; las aplicaciones transaccionales. Las transacciones son los conductos que hacen posible ejecutar aplicaciones reales de negocio con desarrollos rápidos, escalables y fiables.

Active Server Pages (ASP), un componente de Internet Information Server, es el entorno de scripting de servidor, sin necesidad de compilar e independiente del lenguaje, que se usa para crear y ejecutar aplicaciones Web dinámicas e interactivas. Mediante la combinación de DHTML, scripts y componentes, ASP permite que los desarrolladores de aplicaciones generen contenido Web interactivo y potentes aplicaciones Web.

Con la actual tendencia a distribuir los procesos en los entornos empresariales, es importante tener comunicaciones flexibles y fiables entre las aplicaciones. Los negocios requieren a menudo que aplicaciones independientes en sistemas distintos se comuniquen entre sí e intercambien mensajes incluso aunque las

aplicaciones no funcionen al mismo tiempo. Las aplicaciones construidas combinando scripts ASP y en comunicación con los componentes cooperativos pueden interactuar con los sistemas existentes, aplicaciones y datos.

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 sirven 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.

ASP

ASP responden al nombre Active Server Pages, que en castellano significa Páginas de Servidor Activas.

Gracias a esta tecnología creada por Microsoft, el usuario de Internet puede recibir páginas generadas dinámicamente en el servidor.

Agrupadas en la categoría de lenguajes de script (guión) las páginas ASP contienen además de los tags de HTML habituales en las páginas Web, fragmentos de código que el servidor resolverá antes de enviarlo al navegador.

Aplicaciones de las páginas ASP

La facilidad para conectar con una Base de datos y extraer datos de la misma dinámicamente visualizándolos en el navegador es la utilidad más practicada de las páginas ASP.

Puede conectarse a gestores de Base de datos SQL, Access, Oracle, o cualquier otro motor que disponga de driver ODBC.

Comercio electrónico, portales, sedes personalizables y todas aquellas aplicaciones en las que el protagonista es la información dinámica.

Requisitos

Para procesar una página ASP no existe ninguna restricción especial en el lado del cliente, por lo que es indiferente la utilización del navegador Internet Explorer o Netscape Communicator sin embargo, en el lado del servidor, es necesario un servidor Web de Microsoft. Se utiliza el archivo ASP.DLL para interpretar el código, siendo el servidor más extendido Internet Information Server (más conocido como IIS).

Estos son los servidores de contenidos ASP posibles para plataformas Microsoft

- ✓ Internet Information Server 3.0 o superior (para sistema operativo NT)

- ✓ Personal Web Server (para Windows 95 y Windows 98)

Para plataformas Unix es necesario añadir un software que actúe de intérprete siendo algunos de los más conocidos:

- ✓ Chilisoft
- ✓ Instant ASP