

ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

Licenciatura en Sistemas de Información

**“Diseño e Implementación de un Patio de
Comidas Virtual, aplicando tecnología COM y
ASP”**

PROYECTO DE GRADUACION

**Previo a la obtención del TITULO de
LICENCIADO EN SISTEMAS DE INFORMACION**

Presentado por:

**Daniel García Ahmed
Karina Pazmiño Gavilanes
Juan José Peñafiel Guzmán**

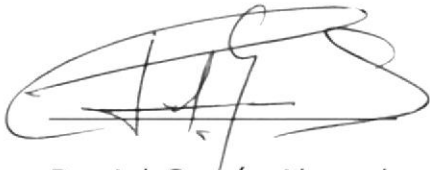
Guayaquil-Ecuador

2002

DECLARACION EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestos en este Proyecto de Grado, nos corresponde exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL"

(Reglamento de Exámenes y Títulos profesionales de la ESPOL)



Daniel García Ahmed



Karina Pazmiño Gavilanes




Juan José Peñafiel Guzmán

TRIBUNAL DE HONOR



Ing. Carlos Martín
Director de Tesis

Ing. Mónica Villavicencio
Coordinadora del L.S.I



Ing. Néstor Arreaga
Miembro Principal

Lic. Jorge Olaya
Miembro Principal

AGRADECIMIENTO

Al Dios Altísimo por habernos bendecido con las facultades para desarrollarnos en esta carrera , por habernos dado la fé y permitirnos estar vivos para disfrutar la satisfacción de haber completado esta etapa tan importante en nuestras vidas. Gracias Dios Soberano!

Cada situación vivida , aprendida y sufrida han servido en beneficio de nuestro progreso profesional.

A nuestras madres Mercedes, Alejandrina y Elena por ser tan excepcionales al apoyarnos con su amor en todo nuestro afán y trabajo para conseguir esta meta.

A nuestros mejores amigos Dayanara ,
Katrina ,Viviana y Andrés por su cariño y
aguante brindado en toda nuestra
trayectoria.

Al Anl. Angel Fiallos, por su invaluable
aporte para la culminación de este
proyecto.

A nuestro gran amigo Johm Macías (+),
nunca te olvidaremos...gracias por seguir
con nosotros....

No olvidamos a nuestros profesores y en
especial a nuestro Director de Tesis, por
estar siempre pendiente del progreso de
este trabajo además de habernos brindado
su amistad y conocimientos, que sirvieron
de soporte para culminar nuestra carrera.

DEDICATORIA

Queremos dedicar todo este proyecto a nuestras familias que nos dieron su apoyo y comprensión en cada momento invertido para nuestra construcción como profesionales.

En honor a ellos y a nuestros seres queridos que no están a nuestro lado pero seguro están felices de que hayamos culminado esta experiencia inolvidable.

Introducción

En el presente documento se proporciona información desde el enfoque administrativo del producto, *PatiodeComidas.com*, es decir, el análisis del mercado, ventajas competitivas y las diferencias con el sistema tradicional, hasta lo que respecta al aspecto tecnológico; se explica el funcionamiento de las tecnologías COM y ASP utilizadas para la implementación del sitio, así como la arquitectura del sistema y el modelo conceptual de datos.

En lo que respecta al análisis del producto, se nombrarán las especificaciones funcionales del mismo, así como las limitaciones y restricciones que tiene el sitio, de igual manera se describirá la industria y el mercado actual, estableciendo el grupo objetivo para nuestro producto y señalando diferencias con los mecanismos tradicionales de comercialización.

En el aspecto técnico, se describirá el desarrollo basado en componentes, el detalle de las tres capas del proyecto: presentación, datos y negocio, y las tecnologías aplicadas en cada una de ellas.

Resumen Ejecutivo

En el presente proyecto de t3pico de Graduaci3n se ha realizado el an3lisis, dise1o e implementaci3n de un sitio Web denominado *Patiodecomidas.com*, que consiste en un "*Patio de Comidas virtual*" el cual permite crear y administrar locales virtuales para los restaurantes que desean comercializar sus productos a trav3s de nuestro sitio en la Internet.

Primeramente se plantea un justificativo para la elaboraci3n del sitio revelando las necesidades que provocaron nuestro inter3s a incursionar en esta l3nea, se especifica la funcionalidad del sitio Web propuesto, considerando las limitaciones que tiene el sistema igual que cualquier otro y tambi3n revisando las restricciones ,el alcance de este tipo de producto tecnol3gico.

En el **cap3tulo 2** , se efectúa un an3lisis de la industria y el producto, donde se mencionan a los clientes directos, es decir los restaurantes , se estudia la reacci3n de los consumidores ante este nuevo tipo de servicio, se revisan nuestros competidores y sus puntos fuertes, el producto sustituto y capacidad del mismo.

Tambi3n se detalla informaci3n referente a la forma de haber seleccionado el grupo de clientes potenciales mediante m3todos para estudio de marketing como las encuestas personalizadas y el an3lisis FODA aplicado a los servicios de entrega a domicilio de comida.

Se consideran las limitaciones y restricciones que tiene el sitio, señalando ventajas, diferencias y desventajas con los mecanismos convencionales de comercialización de comidas.

A continuación en el **capítulo 3**, se detalla el Modelo de Negocios , su funcionalidad; diagramas para revisar el flujo de los procesos relevantes al sitio Web fueron elaborados para una explicación concisa de este tema. Consideramos los sistemas de entregas tradicionales y las definiciones de la manera en como se cancela el valor de los pedidos.

En el **capítulo 4** se detalla la arquitectura del site , las tecnologías utilizadas para la implementación del mismo , se justifica la selección del modelo utilizado, así como proporcionamos la definiciones de tipos de procesos servidores como TPS, Web server,etc. Adicionalmente se describe los requerimientos de hardware y software necesarios para que el site funcione correctamente en la puesta en producción.

Los detalles referidos en el **capítulo 5** especifican la manera en cómo se implementó la capas de datos del site , se adjunta el modelo entidad relación (lógico, físico y relacional) , también se descripción de los objetos diseñados en el modelo físico.

Los temas tratados en el **capítulo 6** conciernen a las especificaciones para implementar la capa de negocios la cual se encargan la reglas operacionales y transacciones dentro del site, además se detalla la estructura de esos "componentes" en conjunto con código de las interfaces ; y también se resume una explicación de la comunicación entre objetos distribuidos COM.

Finalmente, el **capítulo 7** muestra a detalle los objetos creados en la capa de presentación la cual representa la ventana final, al tener en esta implementada la parte visual que utilizan los consumidores finales y los clientes directos. Se revisa el diagrama *top - down* de las páginas diseñadas para el site.

Se revisan los contenidos HTML , las pantallas de mantenimiento elaboradas para el cliente stand-alone desarrolladas en Microsoft Visual Basic 6.0 , se detallan los estilos y las validaciones efectuadas en cada una de las páginas Asp considerada como Cliente Script y las ASP (Server Script); también se considera la comunicación entre los componentes de negocios establecidos.

INDICE GENERAL

Capítulo 1

Especificaciones del Proyecto

1.1 Justificación	1
1.2 Objetivos Generales	3
1.3 Especificaciones Funcionales	4
1.4 Especificaciones No Funcionales	6
1.5 Limitaciones y Restricciones	7
1.6 Alcance	9

Capítulo 2

Análisis de la Industria y Producto

2.1 Definición de la industria y el producto	12
2.2 Clientes	14
2.2.1 Comportamiento del consumidor ante el mercado convencional	17
2.2.3 Mercado virtual vs. Mercado convencional	18
2.2.3.1 Modelo convencional	19
2.2.3.2 Modelo Virtual	20
2.3 Competidores	20
2.3.1 Funcionalidad de los competidores convencionales existentes	21
2.3.2 Desafíos encarados por competidores convencionales	21
2.4 Proveedores	23
2.5 Productos Sustitutos	25
2.6 Análisis FODA del sistema	26
2.6.1 Fortalezas del servicio	26
2.6.2 Oportunidades	26

2.6.3 Debilidades	27
2.6.4 Amenazas del servicio	28

Capitulo 3

Modelo de Negocios

3.1 Definición del Modelo de Negocios	31
3.2 Funcionalidad General	33
3.2.1 Diagrama de Flujo del proceso	34
3.3 Sistema de Entrega	36
3.4 Forma de pago	36
3.5 Seguridad	38

Capitulo 4

Arquitectura del sistema **40**

4.1 Justificación de la Selección del Modelo	43
4.2 Tipos de procesos clientes	44
4.3 Tipos de procesos servidores	45
4.4 Tipos de Middleware	47
4.5 tecnologías y Herramientas escogidas para la implementación	50
4.6 requerimientos de hardware y software para puesta en Producción	51

Capitulo 5

Implementación de la capa de datos **54**

5.1 Modelo Lógico de datos	55
5.2 Modelo Físico de datos	56
5.3 Diagrama Entidad-Relación	60
5.4 Proceso Servidor de base de datos	61

Capitulo 6

Implementación de la Capa de negocios

6.1 Estructura de los componentes de acceso a datos	63
---	----

6.2	Objetos distribuidos de acceso a datos	65
6.3	Comunicación con el proceso servidor de base de datos	70
6.4	Estructura de los componentes de negocios	71
6.5	Objetos distribuido de Negocios	72
6.6	Comunicación entre objetos distribuidos	78
6.7	Proceso Servidor Transaccional	79

Capitulo 7

Implementación de la Capa de Presentación

7.1	Diseño del sitio Web-client basado en browser	84
7.2	Cliente stand-alone	86
7.3	HTML	91
7.4	Client Script -DHTML	94
7.5	Server Script ASP	95
7.6	CSS	101
7.7	Comunicación con componentes de negocios	102
7.8	Proceso Servidor Web	103

Conslusiones y Recomendaciones

Bibliografía

APENDICE A

Arquitectura cliente Sevidor

APENDICE B

Programación en el Web
Tecnologías existentes

APENDICE C

COM /DCOM/COM+

APENDICE E

Comercio Electrónico
Modelo de Negocios
Hosting

CAPITULO 1



- ***ESPECIFICACIONES DEL
PROYECTO***

1.1. Justificación

Los restaurantes de comidas rápidas han tenido un gran crecimiento en el país debido en su mayor parte a la incursión en el mercado de firmas internacionales que han traído consigo mejoras en los estándares de calidad de servicio. Dentro de los mecanismos de venta, el servicio a domicilio constituye un rubro de ingreso importante para este tipo de negocio, siendo la rapidez y efectividad en el producto entregado los factores más influyentes a la hora de evaluar este sistema.

Considerando que la opción del comercio electrónico no ha sido completamente explotada en los países de Latinoamérica así como en Ecuador, hemos visto la necesidad de elaborar un sistema que ayude a los negocios de comidas a promocionar sus productos y receptor pedidos por medio de la Internet, sobre todo pensando en un selectivo grupo de consumidores que confían en la utilización de este medio, hoy por hoy muy usual.

El Internet se ha constituido en la nueva vitrina que tiene el mundo empresarial para exponer sus productos y afianzarse en el mercado, hoy en día, la presencia de una empresa en la Web, dejó de ser una novedad tecnológica para convertirse en un signo de crecimiento e innovación, lo cual brinda más confianza al cliente final.

La mayoría de los mercados de consumo, en la actualidad, se están cubriendo mediante la ayuda del Internet, siempre sujetos a las limitaciones propias del área en donde se

desarrollan, en nuestro país, por ejemplo, las transacciones o negocios “on line” no han tenido mayor presencia, el Internet en la mayoría de los casos, se ha utilizado como una herramienta publicitaria, lo cual deja un campo amplio para el desarrollo de nuevas ideas basadas en la tecnología de la Web.

La tecnologías aplicadas para suplir las necesidades en el mercado consumidor, le ha mostrado a los clientes en nuestro país una nueva forma de obtener servicios, nuevos parámetros de calidad y eficiencia son exigidos, poco a poco las largas colas de espera son menos toleradas , el tiempo y la disponibilidad para acceder a los bienes o servicios de consumo, se convierten en características esenciales a la hora de decidir, por uno u otro proveedor, en otras palabras , los clientes se han vuelto menos tolerantes y las empresas que no satisfagan estas exigencias serán observadores de lujo con el pasar de los años.

A partir de lo expuesto y basados en las nuevas tecnologías de servicio al consumidor que ofrece la Internet, proponemos con la justificaciones ya mencionadas un sitio Web que busca ampliar la cobertura de servicio en el área de entrega de comidas preparadas, permitiendo un mayor poder de decisión para el cliente consumidor y una mayor interactividad entre él y los productos del proveedor.

El potencial del sitio se basa en la facilidad con la que se puede desarrollar una transacción de pedidos en línea ,correspondiendo así con eficacia y agilidad a un creciente mercado virtual.

1.2 Objetivos Generales

Utilizando las nuevas fuentes de información como lo es el Internet, muchas compañías conocen del marco competitivo que esto conlleva, pues la información *on – line* representa una comodidad para el cliente con mayor poder de consumo, que es el usuario de internet.

Esto genera una demanda de mercados para nuestro servicio lo cual buscamos aprovechar y para ello planteamos los objetivos que procuramos alcanzar con el servicio que ofrecemos:

- Obtener una alta rentabilidad que asegure el rendimiento y crecimiento de la empresa en el mercado.
- Incrementar el número de clientes que se anuncia en nuestra página para que de esta manera los usuarios finales encuentre mayor variedad al momento de realizar su pedido.
- Asegurar que los usuario de Internet que realizan el pedido, sean atendidos eficientemente por nuestros clientes.
- Afianzar nuestra presencia en el mercado business to business , mediante el uso de tecnología de punta y calidad en nuestros servicios.
- Conseguir que cuando el usuario navegante del Internet tenga hambre piense en www.patiodecomidas.com .

- Establecer un standard de calidad en lo que se refiere a servicio en línea.
- El visitante al ingresar a nuestro sitio en Internet, www.patiodecomidas.com , podrá escoger entre una serie de lista de comidas por cada restaurante suscrito a nuestro servicio.
- Compromiso corporativo de que el servicio que brindemos sea el mejor y más usado.

1.3 Especificaciones Funcionales

Nuestro producto esta orientado a cubrir las exigencias de un mercado altamente competitivo y cambiante. Consiste básicamente en un sitio WEB al cual los consumidores podrán acceder y de forma virtual visitar el mismo patio de comidas tal como si asistieran a servirse los alimentos al restaurante de su preferencia; al efectuar un pedido a domicilio de lo que se les antoje de entre menús que allí se promocionen.

Entre las características que destacan de nuestro servicio podemos mencionar las siguientes:

- El visitante al ingresar a nuestro sitio en Internet, www.patiodecomidas.com , podrá escoger entre una serie de restaurantes que expenden comidas y distribuyen a domicilio, cada uno de los cuales se encontrará bajo una categoría que describa su especialidad. Por ejemplo las opciones más comunes serían china, italiana etc.

- Una vez seleccionado el restaurante , el visitante entrará a una página diseñada especialmente para ese local, en donde podrá apreciar platos, especialidades, ingredientes, combos, acompañantes ; dejando a consideración del navegante toda una gama de opciones para su selección.
- Todo lo que el usuario elija , será registrado electrónicamente por el sistema, así mismo , tendrá la opción de Confirmar su pedido , en donde podrá elegir tamaños y cantidades de los productos que haya escogido, desechar productos y constatar el valor total de su orden.
- Una vez que se acepte el pedido, se le solicitará una identificación al usuario , de entre las cuales se consideran de vital importancia el número de teléfono, número de cédula o cualquier otra identificación necesaria para la confirmación del pedido.
- El sistema verificará si el cliente ya ha sido registrado , es decir, si ya ha realizado alguna compra en cualquiera de los locales existentes; dicha información reposará en la base de datos destinada para el efecto.
- Si el usuario no está registrado, se le pedirá información relevante , tal como su nombre, dirección, teléfono , cédula de identidad , correo electrónico (e-mail), zona urbana a la que pertenece; dicha información será de uso exclusivo para los procesos de distribución de alimentos de cada restaurante suscrito al patio de comidas.
- El usuario determinará la forma de pago , teniendo como opciones disponibles, el uso de una tarjeta de crédito afiliada para lo cual se pedirán los datos correspondientes, o en su defecto podrá cancelar en efectivo a la hora de recibir la orden.
- El restaurante tendrá la posibilidad de consultar las órdenes entrantes, mediante una opción administrativa en el mismo site, y confirmará la orden mediante una llamada telefónica convencional.

- Tanto el restaurante como el usuario, deberán estar conectados a la Internet, durante todo el tiempo que dure la confirmación del pedido, a través de cualquiera de los proveedores que existen en el mercado local (ISP).
- Una vez que el restaurante confirme la orden, el sistema de despacho y distribución continuará su ejecución tradicional. Para que se asignen los pedidos a los despachadores y se destine el pedido hacia el usuario visitante.
- También son de relevancia el diseño y el funcionamiento de la página Web pues muchas veces son los principales factores que obstaculizan la consumación de las compras online por ende hemos considerado aspectos visuales y de fondo para que al momento de decidir por un proveedor de tiendas virtuales de alimentos , los dueños de los restaurantes piense en nuestro producto.

1.4 Especificaciones no Funcionales

- Se permitirá la inclusión de publicidad que deseen manejar los restaurantes tanto dentro como fuera del site, beneficiando así a nuestros clientes directos que se suscriban al servicio del patio de comidas.
- El mantenimiento del número de pedidos estratificados por sucursal y tipo de producto permitirá a la Gerencia de cada restaurante tomar decisiones al contar con información que servirá tanto para llevar un control estadístico como para efectuar revisiones de margen de venta y despachos realizados.
- La eficiencia en el servicio de pedidos repercutirá en la confianza de los clientes logrando no solo posicionar el producto en el mercado sino a su vez coadyuvar a que

se establezca una cultura de uso del Internet para necesidades tan humanas como la solicitud de alimentos al domicilio.

- Se podrá revisar el margen de visitas por restaurante y/o tipo de comidas más solicitadas en base al número de visitantes que maneja el sitio en general.

1.5 Limitaciones y Restricciones

Hemos elaborado este producto con la finalidad de contribuir a la automatización y agilidad de los procesos compra-venta y despacho con la utilización de la Internet , seremos explícitos al mencionar que no se podrá efectivizar las transacciones de compra-venta en línea , en vista que la misma se confirmará en el momento en que el usuario final recibe al despachador de pedidos.

En vista que en Ecuador no se define aún la aprobación del proyecto de ley sobre comercio electrónico, que regule las transacciones de compra-venta con tarjetas de créditos, hemos optado por no implementar el módulo transaccional de cobro por medio de tarjetas de crédito en línea , y apegarnos al método tradicional para que el cobro se efectivice en el momento que el consumidor final recibe al despachador de pedidos.

Otro aspecto que hemos considerado es el ancho de banda de Internet actualmente no es el ideal como en los países globalizados, una de las limitaciones de este sistema es que los tiempos de respuesta puede tornarse lentos si se congestiona la red.

Los reclamos que se presenten por alguna inconveniencia en los tiempos de entregas o deterioro de los productos solicitados deberán dirigirse a la empresa proveedora del mismo y no significaran responsabilidad de patiodecomidas.com.

La información de los diferentes proveedores que se exhibe, a través de su local dentro de www.patiodecomidas.com, serán de exclusiva responsabilidad de dichas empresas, no asumiremos ninguna responsabilidad sobre el contenido de la información proporcionada por las empresas afiliadas, tampoco garantizaremos la legitimidad de la propiedad de dicha información.

Las ofertas que se exhiban dentro de cada restaurante, son de exclusiva responsabilidad de las empresas que las realicen y comprometen únicamente a estas. Así mismo, la publicidad que se exhibe en la página principal del patio a través de banners o auspicios, es de responsabilidad del restaurante anunciante .

Los visitantes interesados en comprar algunos de los productos ofrecidos en el site, deberán llenar una Orden de Pedido ingresando en esta información relevante que permitirá al proveedor entregar el bien o dar el servicio. Dicha información será considerada confidencial y solo tendrán acceso a ella los respectivos proveedores.

A través del sitio en Internet www.patiodecomidas.com, nos limitaremos únicamente a exhibir y presentar la información relativa a bienes y servicios de las cadenas de restaurantes afiliadas con el fin de poner en contacto a proveedores y visitantes.

1.6 Alcance

Patiodecomidas.com se constituye como un servicio de publicidad y distribución directa que involucra al proveedor de comidas y a los usuarios consumidores, generando así una cultura de negocios por medios electrónicos conocida en términos informáticos como B2B el cual garantiza agilidad de los procesos de compraventa.

Nuestro eslogan : "*Pedimos algo de comer?*", ubicado en las páginas del site asegura un sustancial posicionamiento en las ventas punto a punto desde el lugar de elaboración o distribución hasta el consumidor final; además de fomentar la conquista de un mercado virgen en Latinoamérica que va en aumento día a día.

Además el conocimiento adquirido en la investigación de este tipo de mercados a lo largo del desarrollo e implantación del patio de comidas ,garantiza con seguridad que logremos cubrir necesidades y exigencias que se originen por la creación de este servicio.

Entre los aspectos más importantes del alcance tenemos:

- Los restaurantes que se afilien al servicio puede clasificar sus productos de la mejor manera de acuerdo al tipo de negocio que posea, de la misma manera los nombres de los atributos o características que identifiquen a determinado grupo de producto, el sistema permite realizar una rápida y efectiva búsqueda de los productos ingresados en la base de datos.

- Para la obtención de este servicio será necesario que el proveedor de comidas interesado suscriba un contrato con la empresa representante del patio de comidas a fin de legalizar la utilización del espacio electrónico.
- Las transacciones se podrán efectuar por medio de tarjetas de crédito o en efectivo posterior a una confirmación telefónica; la misma que se efectivizará por el método tradicional de entregas a domicilio.
- Una vez puesta en marcha la construcción de las páginas electrónicas para incluir al negocio del cliente en el patio de comidas , se tendrán 2 semanas para sugerir cualquier cambio en el fondo del diseño de las mismas; tal asunto será tratado con nuestro Web master y diseñadores de planta.
- Cualquier servicio adicional podrá ser incluido de forma gratuita dependiendo de su complejidad, siempre y cuando sea factible su adhesión dentro de las 2 semanas de construcción del *local virtual* dentro del patio de comidas.
- Los costos que se emitan al proveedor del servicio Internet (ISP) por todo el patio de comidas virtual serán cubiertos por la empresa administradora del portal , liberando a su negocio de preocupaciones a este respecto.
- Se contará con un Administrador del servicio automático a fin de que se mantenga la seguridad de los datos que generen las transacciones a partir de los pedidos a domicilio, que solicitan los consumidores finales para cada restaurante afiliado.

CAPITULO 2

- ***ANALISIS DE LA INDUSTRIA Y
EL PRODUCTO***

2.1. Definición de la Industria y el Producto.

La industria del comercio electrónico se ha constituido en la nueva vitrina que tiene el mundo empresarial para exponer, comercializar sus productos y además lograr afianzarse en el mercado.

Ya no se considera novedad tecnológica el que una empresa contrate la elaboración de un site para utilizarlo como signo de crecimiento e innovación, lo cual brinda más confianza al cliente final.

La industria aún deberá sufrir algunas normalizaciones hasta que se logre estándares internacionales tales como el ancho de banda , claras estrategias para las leyes del comercio electrónico.

Con el uso de nuestro servicio los productos de los restaurantes locales podrán ser anunciados en Internet, y los pedidos de los usuarios podrán ser anotadas con exactitud, abaratando costos de personal a la vez que se amplía el mercado de clientes, ya que el número de usuarios de la red aumentan día a día.

Ahora revisemos quiénes forman parte de los consumidores finales del Internet. Estimar la población mundial de Internet en los lineamientos contractuales no es una ciencia exacta. La información mostrada en este mapa fue recogida de numerosas fuentes.



Figura 2.1.1 Población mundial del Internet . Fuente: CommerceNet.

El estudio proyecta para el 2003 que los usuarios de Internet en la región equivaldrían al 5% de los 372 millones usuarios a nivel mundial. Fuente: eMarketer, 2000

Para Latinoamérica, se revisó que sólo 10 a 15% de la población tiene los recursos para comprar On line, según Nevin Cohen, analista de eMarketing, en Latinoamérica hay una "Internet Elite" educada, Cosmopolita y usan apropiadamente la tecnología.

Según otros estudios realizados, los comercios on-line dirigidos al consumidor final deben intentar comprender a esos consumidores que actualmente no son *cyber-compradores*, si quieren tener éxito en el futuro.

El estudio revela que; comprender los hábitos de compra de los consumidores que ya realizan sus compras on-line, es crucial para alcanzar el éxito a corto plazo, pero alcanzar

ese mismo éxito a largo plazo, radicará en la conversión de los NO-compradores, en compradores.

Prosiguiendo con el estudio mencionado esos consumidores NO-compradores probablemente tendrán diferentes perfiles cuando lleguen a ser cibercompradores. Los que no han utilizado todavía Internet, cuando se conecten comprarán, en proporción, mucho más pronto en tiendas virtuales que los que se conectaron a Internet en el pasado.

Por lo anteriormente expuesto, para involucrarnos con esta industria tan movida y cambiante hemos considerado aspectos que al momento de decidir por un proveedor de tiendas virtuales de alimentos, los dueños de los restaurantes piense en nuestro producto.

2.2 Clientes

Basados en el tipo de servicio que ofrecemos, hemos realizado un estudio de los posibles clientes que se suscriban con nosotros de acuerdo al tipo de negocios que estamos localizados identificándolos como un grupo específico.

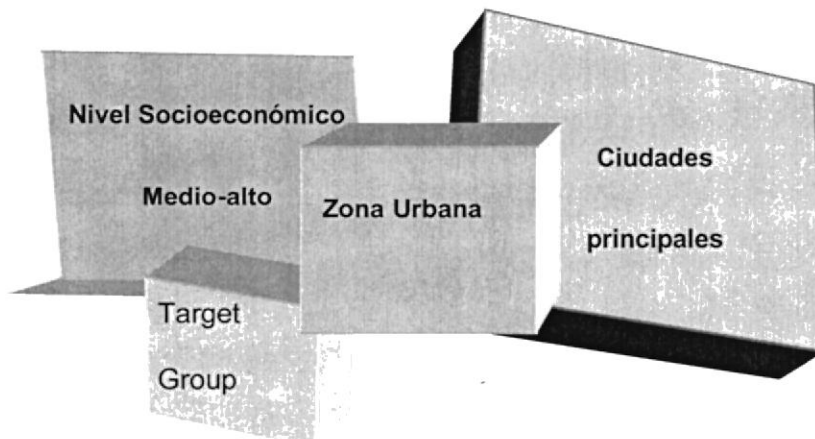


Figura 2.2. Estratificación del grupo objetivo. Fuente: Encuestas realizadas.

GEOGRAFICA	
Región:	A nivel Nacional
Tamaño de Ciudad:	Media – Grande
Densidad:	Urbana
DEMOGRAFICA	
Clase social:	Medio-Alto y Alto
Tamaño de la empresa:	Grande, es decir cadenas de restaurantes que tengan infraestructura para brindar servicio a domicilio.
Ingresos económicos:	Medios y Altos
Ocupación:	Negocios dedicados a la elaboración y venta de alimentos preparados.
PSICOGRAFICA	
Personalidad :	Visionaria, emprendedora, innovadora, ambiciosa, posicionamiento reconocido en el mercado
Valores:	Organizada, responsable, aseada, honesta.

CONDUCTISTA
<p>Tasa de Uso : Clientes reales = 30 % del target group. Clientes potenciales = 70 % del target group.</p> <p>Tipo de Clientes: Cadena de Restaurante de comidas que presten servicio a Domicilio.</p> <p>Beneficios Buscado por el Target group:</p> <ul style="list-style-type: none"> ▪ Obtener mayor venta de los productos por medio de internet. ▪ Calidad en la presentación de sus productos en nuestro sitio. <p>Beneficios Buscado por el Usuario de Internet:</p> <ul style="list-style-type: none"> ▪ Rapidez en el envío de la comida. ▪ Seguridad de los datos del pedido. ▪ Gran variedad de restaurantes y menús.

Tabla 2.2 Clasificación de las variables de Segmentación

Nuestro grupo objetivo son todos los restaurantes y locales de comidas rápidas de tamaño medio-alto con una visión innovadora y organizada, que estén localizadas en zonas urbanas de las principales ciudades del país que prefieran productos de calidad y con ánimos de invertir en un negocio altamente rentable.

2.2.1 Comportamiento del Consumidor ante el mercado convencional.

Actualmente los sistemas convencionales de compra y distribución de alimentos preparados son aceptados por los restaurantes, pero muchas de estas compañías se enfrentan al dilema de la logística de entrega, tener un personal altamente calificado que conozca las zonas de distribución y los tiempos de espera para la recepción de los pedidos lo cual afectará a los clientes directos.

A pesar de todo lo anteriormente mencionado los restaurantes mantienen el esquema convencional , cada uno de ellos haciendo algunos ajustes al mismo con estrategias que proponen aprovecharse al máximo del modo de operación actual.

2.2.2 Reacción del Consumidor frente al mercado virtual.

Se ha demostrado un aumento considerable en la utilización de comercio electrónico entre el 2000 y 2001, pasando de 59,7 mil millones de dólares a 101 millones de dólares, a pesar de las dudas que han existido acerca de la viabilidad de las transacciones negocio al consumidor.

Para el futuro, se predice que el 2002 cerrará con aumento del muy grande 167 mil millones de dólares de gasto mundial en comercio electrónico, seguido por 250 mil millones de dólares en 2003.

La mayoría de las empresas hoy en día usan el correo electrónico para hacer un pedido, para solicitar una cotización, para enviar una lista de precios, etc. ; debemos razonar en base a esta premisa que al utilizar un sistema Internet para acelerar un proceso en la cadena de comercialización de su servicio o producto, ya se lo denomina e-commerce.

Por ende se debe comprender que el uso de e-commerce no comprende únicamente comprar y vender artículos o servicios en la red con la mediación de una tarjeta de crédito, la manera en como se efectivice la transacción no cambia el concepto del tipo de negocios que involucran los procesos mencionados.

Según el análisis revisado consideramos que este tipo de comercialización no deben ser desaprovechadas pues representan una oportunidad excelente de aperturar mercados pocos explotados.

2.2.3.- Mercado virtual vs mercado convencional.

Todo cambio de esquemas considera una etapa de adaptación de la misma hasta que se convierta en parte de las regulares actividades de los negocios. El comportamiento de los mercados varia de manera constante, y hablar de términos virtuales precisa una mentalidad modernista y apegada a las mejoras de las transacciones mediante las nuevas tecnologías que permiten hacer realidad el esquema "virtual" de los negocios mejorando considerablemente la seguridad de los procesos y evitando los típicos errores que se dan en el modelo convencional.

2.2.3.1 Modelo Convencional

Ventajas	Desventajas
<p>Ser un esquema de negocios altamente aceptado en los medios.</p> <p>Contar con buena publicidad tales como páginas amarillas ,propagandas en la Internet</p> <p>Con solo conseguir una llamada telefónica se puede efectuar la solicitud de un pedido.</p>	<p>Predisposición a oponerse a cambios de esquemas.</p> <p>No contar con un sitio Web propio el cual puede convertirse en el centro operacional de los pedidos a domicilio e incrementar la publicidad del local de comidas.</p> <p>Pérdidas de tiempo por parte de los consumidores finales pues muchas veces o no logran contactar el local por congestión de las líneas telefónicas , o no entiende claramente al interlocutor por la rápida manera de explicar las opciones que se ofrezcan en el restaurante contactado.</p>

2.2.3.2 Modelo Virtual

Ventajas	Desventajas
No requerir esperar en ser atendido por una operadora telefónica, el usuario final visita la página de patio de comidas desde su casa.	Se presentarían en caso que este fallando la conexión de Internet o algún problema con el proveedor del servicio.
Facilidad en la toma de decisiones para el consumidor final al no tener que sentirse presionado por el interlocutor para efectuar el pedido y contar visualmente con las posibles opciones que tenga cada restaurante.	Ninguna
Contar con una variedad de locales a su disposición tanto como las opciones que expenden en cada uno de estos. Todo en solo sitio Web que se visita.	Ninguna

2.3. Competidores

Una competencia considerable corresponden al servicio tradicional de entrega de comidas a domicilio los cuales efectúan el inicio de una transacción comercial vía telefónica, el mismo posee algunas limitaciones en cuanto al proceso de selección de productos y agilidad en la atención.

También compite el grupo de productos que se estén orientando en el estilo de negociaciones vía Internet.

Empresas de software que desarrollen sistemas de comercio electrónico, que elaboren una página de servicio con similares características que la nuestra.

Según nuestra investigación, en el Ecuador no existe una página que brinde tales beneficios, por ende no existe actualmente una competencia declarada en este sentido que merme nuestro interés de llevar adelante este proyecto de servicios y alcanzar éxito en los objetivos planteados.

2.3.1 Funcionalidad de los competidores convencionales existentes.

Los competidores convencionales solo cumplen con la función de recepción de pedidos vía telefónica y luego de una breve confirmación se preocupan en la entrega de los pedidos en el domicilio del cliente final cerrando el negocio en sitio pero sin contar con los beneficios que plantea el patio de comidas como el de visitar “virtualmente” el restaurante de la preferencia y escoger los alimentos a solicitar; dando espacio a la imaginación y afectando los

2.3.2. Desafíos encarados por competidores convencionales

A pesar de la acostumbrada utilización del sistema convencional en las solicitudes de alimentos a domicilio, muchas compañías se enfrentan a algunos desafíos que deben enfrentarse constantemente a saber :

El dilema logístico de entregas , no es fácil contar con personal calificado para este tipo de trabajos ya que se requiere que conozcan las zonas de destino de los despachos y que cuenten con un medio de transporte si es que el restaurante no tiene uno disponible.

Las regulares quejas que presentan los usuarios finales sobre las demoras en los tiempos de despachos y a veces hasta deterioro de los productos solicitados es una de las razones que suelen poner en desventajas esta forma de comercialización tradicional.

Para contactarse con un local de comidas el usuario debe efectuar varias llamadas telefónicas y a veces no se consigue contactar con el local , lo cual implica que el mismo deberá desatender sus tareas para preocuparse en conseguir la contactar al operador(a) del local deseado.

Tener operadores(as) que mantengan excelente vocalización durante cada atención de pedidos telefónicos ya que en las horas pico y sobre todo en los feriados es difícil entender los planteamientos de menús dados por el operador(a) en línea; ya que regularmente esta tratando de atender lo más rápido posible a la mayor cantidad de usuarios; y su ritmo de explicación se acelera ante una demanda continua. La situación se torna más difícil si se manejan varias líneas telefónicas.

2.4. Proveedores

Para el funcionamiento completo de patio de comidas será necesario contar con elementos indispensables a que se mencionan a continuación:

- Contar con un proveedor de Servicio de Internet a fin de tener un alquiler virtual de espacio en la red y de que necesita. (Ver apéndice E Tabla E.1.-Lista de Precios de Hosting).
- Gente especializada para el mantenimiento de la página principal así como de los nuevos locales que se adhieran al patio de comidas , hemos planificado un valor de entre \$500 -\$700 mensuales.
- Los valores que el cliente deberá cubrir para tener una local virtual en nuestro sitio o banner publicitario, así como los servicios a los cuales tendrá derecho se detallan a continuación :

Concepto	Descripción	Costos (mensual)
Hosting (alquiler de espacio en la Internet).	<ul style="list-style-type: none"> ▪ 100 MB de espacio para almacenar información referente a su local (*Ver Apéndice E) ▪ 1 cuenta electrónica (email) para recepción de pedidos (*) 	\$95.88
Mantenimiento y actualización de la página	<ul style="list-style-type: none"> ▪ Adición de información ▪ Cambios en la estructura de la página ▪ Control estadístico del sitio: número de órdenes atendidas, número de visitas a su local. 	\$180
Anuncios publicitarios	<ul style="list-style-type: none"> ▪ Diseño de banner y logotipo ▪ Exhibición del banner en patiodecomidas.com 	\$150

2.5. Productos Sustitutos

- Los sistemas tradicionales de servicio de comidas a domicilio que ofrecen las mismas cadenas de restaurantes que son nuestros clientes, son los que satisfacen la demanda actual pues es un medio fácil y accesible en esta sociedad de servicios.

2.6. Análisis FODA del sistema

Patio de comidas como herramienta para el manejo electrónico del servicio a domicilio de las cadenas de comidas , surge como un concepto innovador en el mercado online, por lo cual sus oportunidades en el mismo, dependerán mucho de las campañas que se emprendan para su difusión; de igual manera ,sus amenazas la constituyen principalmente la competencia abierta y muchas veces sin un marco legal claramente establecido y funcional , que conlleva el trabajar en la Internet.

2.6.1 Fortalezas del servicio

Con las innovadoras tecnologías ponemos al alcance de los locales de comidas un mundo de oportunidades para concretar negocios, convirtiéndonos en un eficaz instrumento de publicidad y de apoyo para la distribución de los pedidos a domicilios , colocando los productos en la mejor valla que puede haber, la Internet.

2.6.2 Oportunidades del Servicio

Atiende la demanda de los usuarios de Internet, los cuales ven en ella una herramienta mas cómoda y atractiva para la búsqueda de servicios y/o productos.

Poder convertirnos en líderes del mercado de e-commerce al arriesgarle todo nuestro potencial a este negocio, ya que muchas empresas aún se resisten a invertir con fé y poner lo mejor de sí para salir adelante.

Servicio innovador en la Internet, en lo que al mercado local se refiere , ya que es el único que permite escoger restaurantes y armar pedidos online; todo en un mismo lugar.

El restaurante puede mostrar todos los platos e ingredientes disponibles, los cuales muchas veces no son solicitados en los servicios a domicilio, debido a la falta de tiempo para promocionarlos.

Al proveer un medio visual (gráficos y menús) , el usuario puede tener una mejor idea de lo que esta ordenando, para de esa manera evitar disconformidades a la hora de recibir su orden.

2.6.3 Debilidades del Servicio

Muchos de los empresarios dueños de estos locales no demuestran una gran apertura hacia los negocios en Internet, creando una resistencia a invertir en este tipo de mercado, tal vez considerando como débil la posibilidad de una recuperación de costos en que se incurran para implementar la tienda correspondiente en el patio de comidas.

La tecnología aplicada, es relativamente nueva en el país, lo que se ve reflejado en la poca mano de obra calificada a la hora de evaluar un potencial empleado para el desarrollo de componentes COM/DCOM para aplicaciones distribuidas.

Poca experiencia en el trato real con este tipo de mercados que cada vez se entorna más complicado y exigente.

2.6.4 Amenazas del Servicio

Entre las principales amenazas definimos las siguientes:

- El sitio puede tornarse lento debido a factores externos como las líneas telefónicas o problemas en los servidores locales de Internet, asunto que es muy típico en nuestro medio por la poca infraestructura de comunicación existente en Ecuador.
- Negocio propenso a riesgo debido a no existir el marco legal definitivo para el entorno de las transacciones en que se incurren.
- La información en la página puede encontrarse desactualizada, debido a cambios repentinos de los restaurantes en sus menús o promociones que no hayan sido comunicados con debida anticipación, lo cual trae consigo pedidos que no puedan ser satisfechos.
- Resistencia de los usuarios a usar el Internet para realizar pedidos a domicilio, prefiriendo la manera tradicional, considerándola más confiable y conocida.
- El restaurante afiliado puede hacer caso omiso de los pedidos por Internet , lo cual significaría clientes insatisfechos y repercute no solo en el restaurante sino también en el grado de utilización del site por otros consumidores al generarse una fama de ineficacia.
- Sitios que presten un servicio similar que de alguna manera quieran superar nuestro servicio, cualquier descuido del marketing u otro aspecto del servicio podría mermar el posicionamiento que logremos.
- La libertad que existe en el Internet lo cual expone el sitio a daños provocados por terceros como lo son los virus informáticos, y otros factores incontrolables.

CAPITULO 3

- ***MODELO DE NEGOCIOS***

3.- Modelo de Negocios

Todo negocio implica la comercialización de un servicio o producto a un cliente final por parte de un proveedor, satisfaciendo requerimientos ya definidos por ambas partes. En internet, al igual que el tipo de negocios tradicional, el marco de desarrollo de la transacción comercial es similar, guardando marcadas diferencias en cuanto a seguridad, forma de pago y sistemas de entrega.

Los modelos de negocio en internet se definen dependiendo del tipo de transacción que se realice o la orientación del grupo objetivo al que se quiere atender, entre los modelos de negocio ya establecidos en la actualidad podemos mencionar:

Bussines to Bussines (B2B). El modelo B2B consiste en una solución que se ofrece a un proveedor para que este ofrezca sus servicios o productos al consumidor final (B2C), en este modelo de negocios pueden "agruparse" proveedores de productos o servicios similares bajo un mismo sitio.

Bussines to Customer (B2C). El negocio se establece entre el proveedor del sitio y el consumidor final, no existen intermediarios, es el modelo mas comun en la internet.

Customer to Customer (C2C). El negocio se establece entre los usuarios que visitan la pagina, los proveedores de este tipo de modelo en su mayoría ofrecen servicios y actúan como intermediarios en la negociación entre los clientes, un ejemplo de este tipo de negocios se puede apreciar en los llamados sitios de subasta, en donde el proveedor del servicio cobra una comisión por cada transacción realizada, pero el negocio se establece entre los afiliados al mismo.

3.1 Definición del modelo de negocios

El modelo de negocios implementado en nuestro sistema, es definido como B2B (Business to Bussines), aunque esto es relativamente cierto a la hora de analizar la aplicación en todo su entorno y considerando su verdadero objetivo en el mercado al cual apunta.

La aplicación, siempre debe ser vista como un sistema de creación y administración de restaurantes virtuales en su primera parte y, como un servicio online de pedidos a domicilio, en su segunda parte, esto implica que el modelo de negocios debe comprender cada una de ellas en las que está dividida la implementación del site.

Por lo antes expuesto, procederemos a explicar el modelo de negocios, atendiendo cada una de las perspectivas del mismo.

La administración y creación de restaurantes virtuales

Es la oferta y objetivo principal del sistema, ya que atiende a sus clientes directos, los restaurantes, en este primer término la aplicación se enfoca en el modelo **B2B (Bussines to Bussines)** .

A continuación explicamos los motivos por los cuales escogimos este modelo:

- Los potenciales clientes del sistema PatiodeComidas.com, lo constituyen los restaurantes de comida rápida, o que cuenta con un esquema de servicio a domicilio.

- Estos a su vez, buscan vender sus productos, mostrándolos en nuestro site www.Patiodecomidas.com, con el objetivo de cubrir otra área del mercado y de incrementar sus ventas.
- El restaurante afiliado desembolsará un valor mensual, por el alquiler del “local virtual” en nuestro sitio, y éste a su vez manejará sus ventas directamente.

Servicio “on line” de pedidos a domicilio

Es el producto final, y la razón del negocio, que es el atender pedidos a domicilio a través de la internet, para éste caso el modelo de negocios implementado es definido como **B2C (Business to Customer)** ya que la orientación gira totalmente hacia el consumidor final, que es el usuario de internet

- El consumidor, entrará a la página, www.patiodecomidas.com y escogerá un restaurante en particular.
- Una vez elaborado el pedido, elegirá una forma de pago que será efectivizada en el momento de recibir su orden.

Para una mejor explicación de los modelos de negocios, puede hacer referencia al apéndice E de este manual.

3.2 Funcionalidad General

Una vez implementado el sitio, cualquier restaurante que maneje un esquema de entrega de comida a domicilio podrá "arrendar" un espacio virtual en Patiodecomidas.com, para lo cual existirán categorías previamente definidas de tipos de comida en donde se agruparán los restaurantes , haciendo mas fácil para el consumidor la orientación dentro del sitio.

Los tipos de comida no serán predefinidos, es decir, se ajustarán a la variedad de restaurantes inscritos en la página, lo cual será manejado directamente por el administrador del sitio.

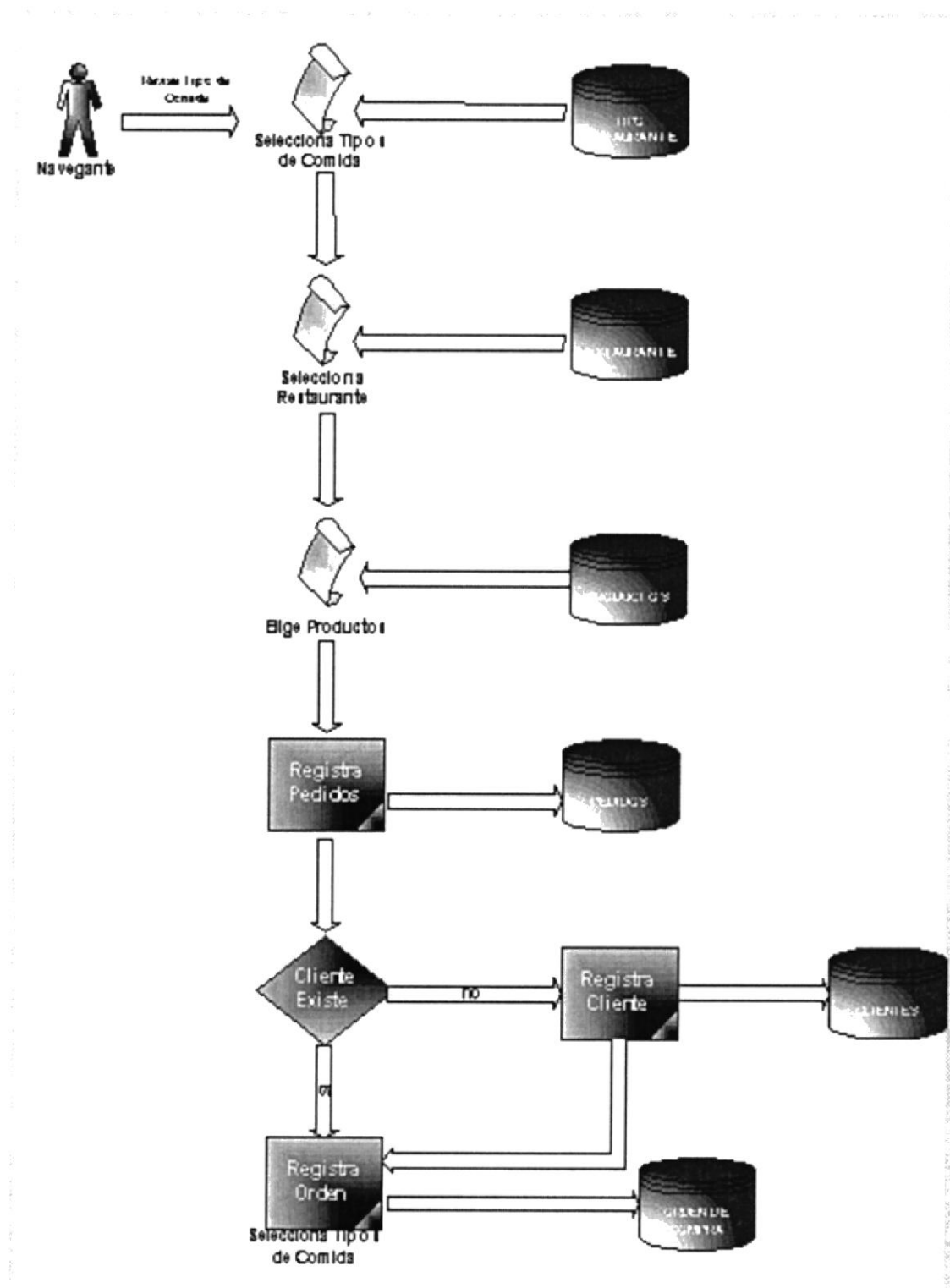
Cada restaurante tendrá una pagina individual en donde exponer sus productos y registrar sus pedidos, así como los clientes que efectúen transacciones en su local virtual.

Una vez que el cliente registre su orden, recibirá una confirmación del restaurante vía correo electrónico luego, el pago y la entrega de los productos, se realizarán de manera personalizada como en el sistema tradicional.

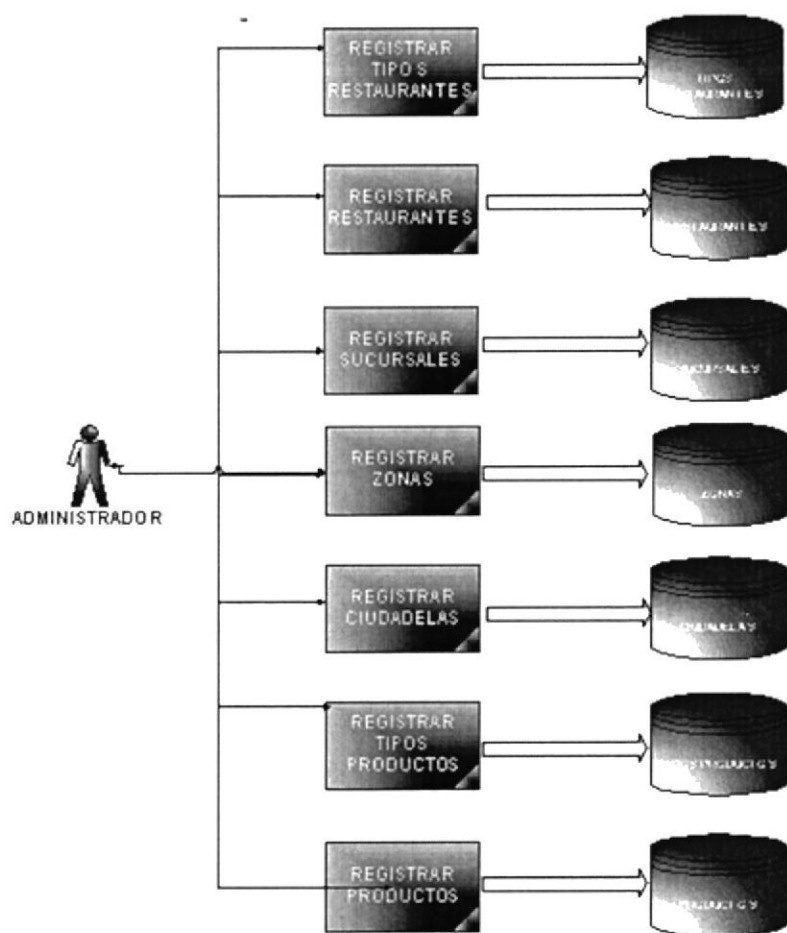
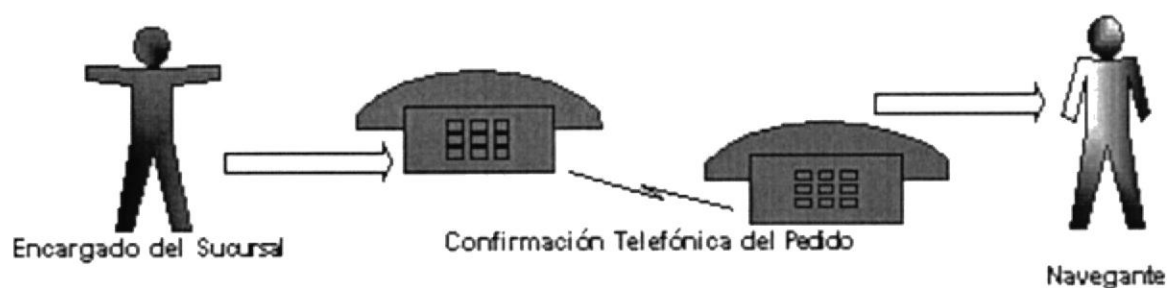
El sitio además ofrecerá reportes de control para los restaurantes: ventas, clientes atendidos, etc , así como un paquete publicitario para el posicionamiento de la pagina en el mercado.

3.2.1.- Diagrama de Flujo del Proceso

Flujo de Proceso del Site www.PatiodeComidas.com



Confirmación del pedido



Flujo de Proceso del sistema administrador (Stand – alone)

3.3.- Sistema de Entrega

- El sitio manejará un sistema de entrega personalizado, es decir, el restaurante afiliado deberá contar con un esquema de entregas a domicilio, lo cual incluye personal y transporte.
- El area de cobertura será establecido por el restaurante dependiendo de sus políticas y la distribución física de sus sucursales; vale acotar que el sitio maneja el concepto de sucursales por restaurante permitiendo definir mas de una sucursal para un restaurante.
- Los consumidores dentro del sistema son agrupados por zonas y luego, en un nivel mas interno, por ciudadelas, lo cual ayudará a que la orden sea atendida por la sucursal mas cercana al lugar donde reside el consumidor.
- Previo a la entrega del producto, el consumidor recibirá una confirmación de su pedido indicando la sucursal que lo atendió.

3.4.- Forma de Pago

- La forma de pago es establecida por Patiodecomidas.com, de tal manera que el restaurante deberá manejarse dentro de esos parámetros.

- La transacción de pago no se realizará de manera electrónica, sino que se mantendrá de la forma como la lleva cada restaurante como cuando se trata de un pedido telefónico.

- El pago de los productos se podrá realizar de dos maneras: en efectivo y por tarjeta de crédito; cuando sea en efectivo, el despachador recibirá el dinero al momento de la entrega física del producto, y el consumidor a su vez recibirá la factura física que certifique la transacción; en el caso de que se realice por tarjeta de crédito, el consumidor deberá proporcionar el número, nombre del emisor, fecha de vencimiento y código de la misma, los datos serán verificados inmediatamente por el restaurante y de aprobarse el crédito, se procederá a la emisión del voucher, el cual será firmado por el consumidor en el momento de la entrega física del producto.

- Los emisores de tarjetas de crédito con los que trabajarán los restaurantes serán establecidos por Patiodecomidas.com, es decir, los restaurantes sólo podrán aceptar dichas tarjetas.

- Los montos mínimos y máximos aceptados para realizar una transacción en el sitio, serán establecidos por el restaurante, dependiendo de sus políticas.

3.5.- Seguridad

La implementación de la seguridad se la debe enfocar desde dos puntos de vista: la del consumidor final y la del restaurante.

Los riesgos en que se incurren son tanto para la transacción como para la entrega del producto.

Riesgos para el Restaurante	Medida a Tomar
Al momento de establecer la forma de pago, los datos suministrados sean incorrectos	Para el caso de tarjeta de crédito, se pedirá la autorización en el momento al emisor, en el caso de pago en efectivo se asume el riesgo
En la entrega de productos, los datos del cliente sean falsos	El mismo riesgo se corre bajo el sistema actual
Riesgos para el cliente	
Los productos anunciados no están disponibles	La orden será confirmada al consumidor, previo a revisar que todo el pedido pueda ser atendido, de no ser así se le indicará al consumidor que revea su orden
Los productos entregados no estén de acuerdo a lo ordenado	El restaurante asumirá la responsabilidad y entregará lo acordado con el consumidor.

CAPITULO 4

- ***ARQUITECTURA DEL SISTEMA***

4.- Arquitectura del Sistema

Nuestro sistema permitirá que los restaurantes manejen su catálogo de productos en la Internet, para lo cual nos pagarán un alquiler o "hosting" en donde estarán almacenados sus productos.

El consumidor final accederá por medio de un browser a la internet, y consultará los productos, elaborará pedidos, los datos del catálogo deberán ser actualizados por parte del cliente (restaurante).

A continuación presentamos un esquema general de este tipo de arquitecturas:

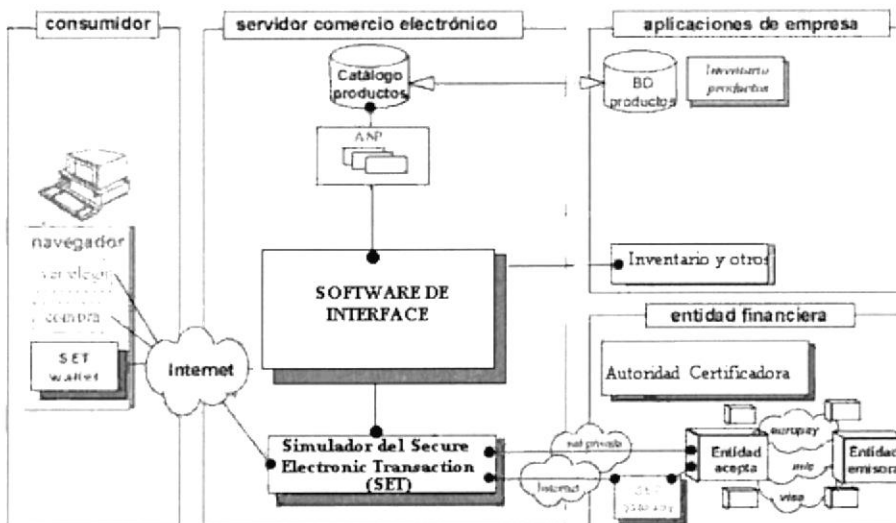


Diagrama 4.1 Estructura de tiendas virtuales

Para implementar la arquitectura de nuestro sistema Patiodecomidas.com, se utilizó el concepto desarrollado por Microsoft, denominado Windows DNA (Distributed Network Application).

- La estrategia Windows DNA de Microsoft descansa en la visión de Microsoft de los componentes cooperativos que son construidos basándose en el estándar binario llamado el modelo de componente objeto (COM).
- La arquitectura Windows DNA, describe como construir aplicaciones de tres niveles para la plataforma Windows.
- Una aplicación de tres niveles es una aplicación cuya funcionalidad puede ser segmentada en tres niveles lógicos de funcionalidad: los servicios de presentación, los servicios de negocios y de datos.

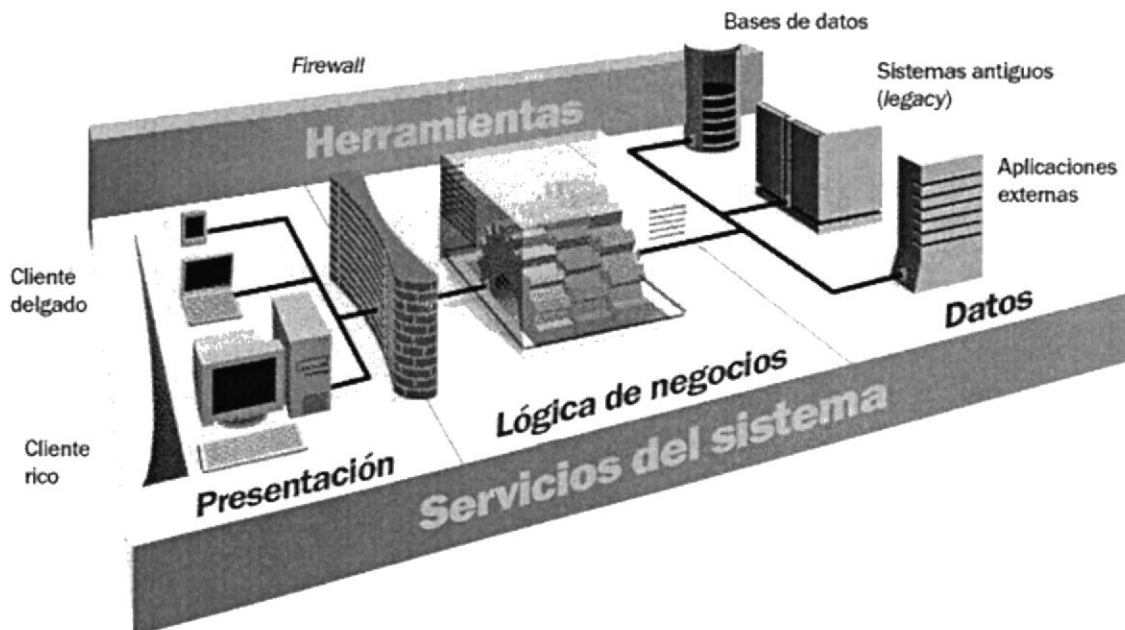


Figura 4.1.2 Arquitectura tres capas

La capa de servicios de presentación es responsable de:

- Obtener información del usuario
- Enviarla información del usuario a los servicios de negocios para su procesamiento.
- Recibir los resultados del procesamiento de los servicios de negocio
- Presentar éstos datos al usuario

El nivel de servicios de negocios es responsable de:

- Recibir la entrada del nivel de presentación.
- Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar.
- Enviar el resultado procesado a nivel de presentación

La capa de servicios de datos es responsable de :

- Almacenar de datos
- Recuperar los datos
- Mantener los datos
- La integridad de los datos

4.1 Justificación de la Selección del Modelo

Con el modelo Microsoft Windows DNA, los componentes eliminan la complejidad de construir aplicaciones multinivel. Las aplicaciones basadas en componentes y el modelo Windows DNA se basan en un conjunto común de servicios de infraestructura y de red proporcionados por la plataforma de aplicación Windows. El servicios de seguridad de Microsoft Windows NT®, por ejemplo, proporciona control de acceso al Servidor de Información Internet, así como servicios de transacciones y mensajes cola de mensajes. Otros servicios comunes incluyen administración de sistemas, servicios de directorio, red y soporte de hardware.

Dentro de las ventajas de implementar un modelo distribuido podemos mencionar:

- **Lista para Internet.** Desarrollar soluciones que exploten al máximo la flexibilidad de la plataforma de aplicación y el alcance global de Internet y las capacidades de comunicación por demanda.
- **Tiempo de comercialización más rápido.** Desarrollar e implementar soluciones rápidamente sin requerir una reeducación del desarrollador o un cambio de paradigma en la manera de construir software. Exponer los servicios y funcionalidad a través de la "plomaría" subyacente para reducir la cantidad de código que los desarrolladores deben escribir.
- **Verdadera interoperabilidad.** Construir la interoperabilidad en todos los niveles de tal forma que se pueda añadir funcionalidad a los sistemas existentes. Adherencia a los protocolos abiertos y estándares de forma que se puedan integrar soluciones de otras compañías.

- **Complejidad reducida.** Integrar servicios clave directamente en el sistema operativo y exponerlos de manera unificada a través de los componentes. Reducir la necesidad de que los profesionales de tecnología de información (TI) funcionen como integradores de sistemas, de forma que se puedan enfocar en la solución de los problemas del negocio.
- **Independencia de lenguaje, de herramienta y de hardware.** Proporcionar un modelo de componente neutral respecto al lenguaje de forma que los desarrolladores puedan utilizar herramientas adecuadas para cada tarea. Construir en el modelo de cómputo de la PC, donde los clientes pueden implementar soluciones en el hardware que está ampliamente disponible.
- **Más bajo costo total de propiedad.** Desarrollar aplicaciones que sean fáciles de implementar, administrar y cambiar a lo través del tiempo.

4.2 Tipos de Procesos Clientes

Cliente stand-alone (Independiente)

Implementado en Visual Basic 6.0 y corre bajo sistema operativo Windows 2000

Cliente Browser (Dependiente)

El tipo de proceso que se ejecuta del lado del browser, Internet Explorer , es el *client script*, el mismo es implementado en javascript y entre sus funciones están:

- Permite validar los datos ingresados en formularios antes de ser enviados al “servidor”
- Evita el tener que enviar datos no validados al “servidor”.
- Mejora el rendimiento ya que no se requiere que los datos viajen al “servidor” para ser validados. La validación ocurre del lado del “cliente”.

Del lado del cliente se recibe código html y script que es interpretado en el web server.

Dependiente del Browser ya que el modelo de objetos del documentos (DOM por sus siglas en inglés) puede variar dependiendo del browser.

El browser se encarga de detectar errores , interpretar y ejecutar el *client script*.

4.3 Tipos de Procesos Servidores

Los procesos servidores que implementamos en nuestro sistema , son transaccionales y de acceso a datos

Database Server

El proceso servidor de la base de datos en nuestro sistema, es del tipo *no transaccional*, ya que no implementamos *store procedures* para el acceso a los datos.

Para el proceso servidor de base de datos se utilizó el motor SQL Server 7.0 de Microsoft

Transaccion Server (TP monitor)

Utilizamos el transaction processing server (Tp monitor). La principal función del tp monitor :

- La administración de procesos, lo cual incluye poner en marcha los procesos del servidor, canalizar trabajo en dirección a ellos, vigilar su ejecución
- La administración de transacciones, lo que significa que garantiza que una transacción sea realizada.

TP ligero vs. TP pesado

TP ligero es la integración de las funciones del monitor TP a los mecanismos de base de datos, mientras que TP pesado soporta toda la arquitectura cliente / servidor de la aplicación, como se describió anteriormente.

Para nuestro caso, se implementó TP pesado, ya que la administración de las transacciones de la base de datos la realiza el TP monitor.

Server Script

Es el script que se ejecuta del lado del web server, para nuestro sistema hemos utilizado la tecnología ASP.

Cuando las páginas html son invocadas por el web server se genera páginas ASP

El server script procesa la información de los formularios.

4.4 Tipos de Middleware

El Middleware soporta la comunicación entre cliente y el servidor.

A continuación daremos una definición de los tipos de middleware utilizados en nuestro sistema *PatiodeComidas.com*

ADO

- Interface a nivel de aplicación (API) que facilita el acceso a cualquier OLE DB data provider.
- Mejora los tiempos de respuesta
- Independiente de la programación.

HTTP

- Protocolo de comunicación entre Web Browser y Web Server
- Usado para acceder y obtener recursos
- Es un protocolo sin estado (stateless). Por cada requerimiento:
- El cliente establece una conexión con el servidor
- El cliente envía el requerimiento.
- El servidor entonces procesa el requerimiento, retorna una respuesta y cierra la conexión.
- El cliente hace un requerimiento a través de una página html

- El cliente debe esperar por una respuesta antes de enviar un nuevo requerimiento.
- HTTP no permite múltiples requerimientos se una misma sesión

COM

La arquitectura de nuestro sistema, como se mencionó anteriormente, se basa en la implementación de tres capas, la comunicación entre los procesos se realiza a través de la infraestructura de COM.

Nuestro sistema ha sido desarrollado bajo la plataforma Windows 2000, por lo que no utilizamos MTS (Microsoft Transaction Server) para la comunicación de procesos sino la tecnología COM+ para poder trabajar con clientes remotos. (*Ver anexo C*)

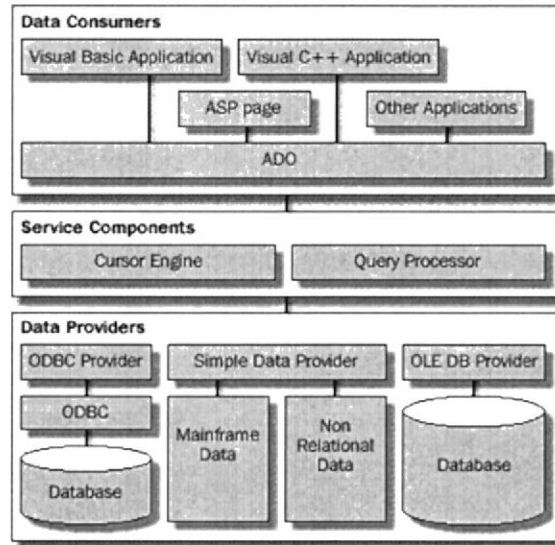
OLE DB

Mientras la conectividad abierta de base de datos (ODBC por sus siglas en inglés) está diseñada para permitir el acceso a los datos relacionales, OLE DB es un diseño estándar para permitir el acceso a toda clase de datos. Conceptualmente, OLE DB tiene tres tipos de componentes: *data providers*, *data consumers*, y *service components*.

Data Providers

Data providers son aplicaciones, como Microsoft SQL 7.0o Microsoft Exchange, o componentes de sistemas operativos, como un archivo de sistema, que tienen datos que otras aplicaciones necesitan acceder. Estos data providers exponen su OLE DB interfaces de tal manera que los *data consumers* puedan acceder directamente.

Figura 4.4.1 Arquitectura OLEDB



Data Consumers

Data consumers son aplicaciones que utilizan los datos expuestos por los *data providers*.

Toda aplicación que utilice ADO es un OLE DB *data consumer*.

Service Components

Service components son componentes de OLE DB que procesan y transportan datos.

Estos componentes incluyen procesos de consultas. Desde el punto de vista de

arquitectura, OLE DB, esta separado en componentes de tal manera que los *data providers* no necesitan poseer datos extras para que ADO pueda entender.

4.5 Tecnologías y Herramientas escogidas para la Implementación

El proyecto se ha basado en la tecnología ASP y COM para distribución de aplicación; se ha utilizado como lenguaje de programación el conjunto de herramientas de Visual Studio, Macromedia (Flash , Dreamweaver) y nos hemos soportado en el material explicativo que se nos impartió durante el dictado de la materias del Tópico de Graduación.

Tecnologías

La estrategia Windows DNA de Microsoft descansa en la visión de Microsoft de los componentes cooperativos que son construidos basándose en el estándar binario llamado el modelo de componente objeto (COM). COM es modelo de componentes software más ampliamente utilizado en el mundo, disponible en más de 150 millones de máquina de escritorio y servidores en la actualidad. Proporciona el conjunto más rico de servicios integrados, la más amplia elección de herramientas fáciles de utilizar y el más grandes conjunto de aplicaciones disponibles. Además, proporciona el único mercado actualmente viable de componentes cliente y servidor, prefabricados y reutilizables.

Active Server Pages (ASP) es una herramienta para crear las páginas dinámicas e interactivas del web. El ASP es una tecnología de Microsoft que trabaja permitiendo que utilicemos la funcionalidad de un lenguaje de programación que genera el HTML para el webpage dinámicamente.

Herramientas de Programación

- Microsoft: Visual Basic 6.0, Utilitarios de Visual Estudio
- Plataforma : Windows 2000
- Macromedia : Dreamweaver

Herramientas de Diseño

- Macromedia : Dreamweaver, Flash
- Jacsoft : Paint Shop Pro

Herramientas de Base de Datos

- Microsoft : SQL 7.0, Msquery

4.6 Requerimientos de Hardware y Software para la puesta en Producción

Tanto el equipo, como los programas que éste maneje, son factores determinantes a la hora de medir el rendimiento de una aplicación, para nuestro caso en particular, definiremos los requerimientos, no mínimos, sino los que permitan un desempeño adecuado del sistema, tanto a nivel del *cliente stand-alone*, como el *cliente on-line*

El sistema deberá rendir óptimamente en plataformas de tecnología Microsoft, en sistemas operativos a partir de Windows 9X y 2000

Las páginas Web de la tienda virtual, se visualizarán en los principales navegadores Explorer y Netscape , en las últimas versiones.

El sistema administrativo *stand-alone*, se podrá instalar en equipos que tengan una resolución de 640x480, para que así se lo pueda ejecutar en equipos con menor resolución.

Requerimientos de Hardware

Cliente	Memoria RAM	Procesador	Capacidad (HD)	Velocidad Conexión
Stand-alone	128	Pentium III 300 Mhz	Standard	
Online	64	Pentium II 233 Mhz	Standard	28 Kbps

Requerimientos de Software

Cliente	Sistema Operativo	Navegador
Stand-alone	Windows 9x,NT, 2000	Pentium III 300 Mhz
Online	Windows 9x,NT, 2000	Internet Explorer, Netscape

CAPITULO 5

- ***IMPLEMENTACION DE LA
CAPA DE DATOS***

Implementación de capa de datos

En esta parte se explica la manera en como se implementó la "capa de datos " a partir de los diferentes objetos lógicos y físicos , que se han elaborado para el control de los datos que se encuentran almacenados en el la base de datos SQL SERVER.

Queremos enfatizar que la construcción de la "capa de datos" logra a su vez tener alejada a cualquier otra actividad de los métodos relacionados con la apertura y escritura sobre los registros de los objetos de la base de datos y a su vez mantener un criterio de utilización tardía y liberación inmediata es decir cada vez que otra parte del negocio requiera efectuar alguna transacción en la base recupera el recurso y lo libera de inmediato dejando para después la confirmación de la transacción en otro momento que se solicite y así no entorpecer a otra transacción que desee utilizar el mismo recurso.

Para el análisis del modelo lógico definiremos como objetos de las bases de datos a las tablas , ya que para la construcción de este site no se utilizaron procedimientos almacenados (que también son objetos de bases de datos) ; las cuales son estructuras de almacenamiento , las mismas que describiremos a continuación con una breve descripción del tipo de información que almacenarán.

5.1 Modelo Lógico de datos

Aquí se detallan los objetos de la base (tablas) que se diseñaron para implementar el site patio de comidas, las mismas que serán alimentadas por un módulo stand alone para los mantenimientos referenciales y a su vez por el web browser, los datos registrados serán utilizados para el catálogo y transacciones de la tienda virtual.

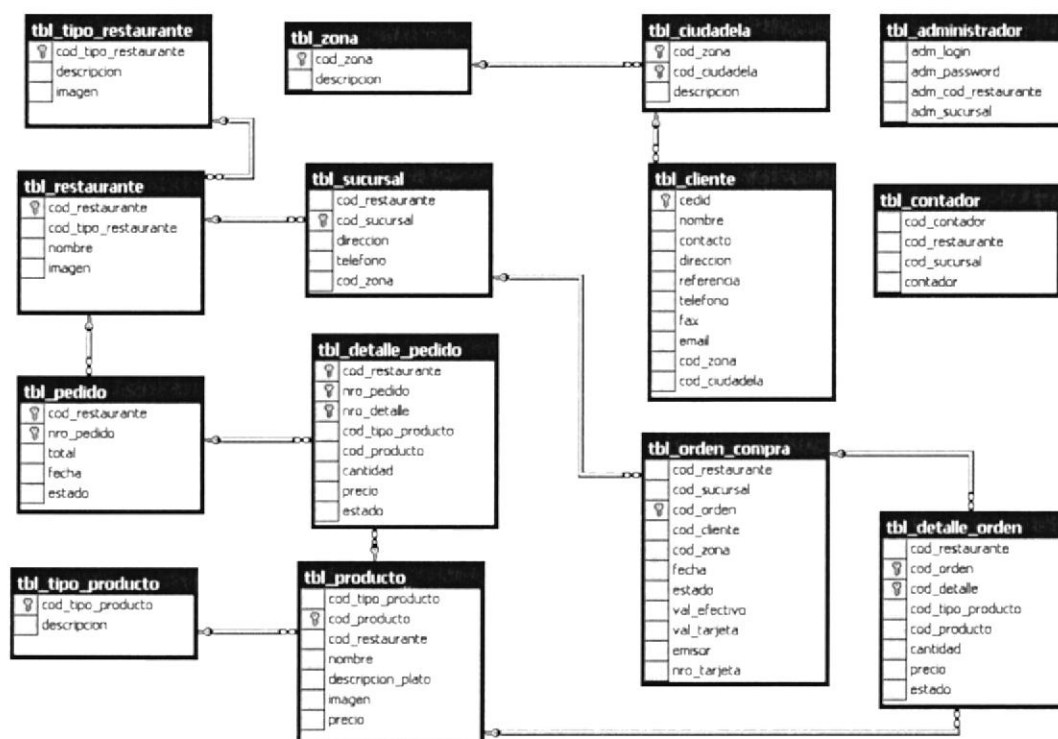


Figura 5.1 Modelo Lógico de datos

La descripción de cada uno de los elementos así como los tipos de datos de las tablas se detalla en la sección 5.2 de este manuscrito.

5.2 Modelo Físico de Datos.

Se detallan el modelo físico de las tablas que se elaboraron para la construcción del site , con la estructura que soporta todo el esquema funcional del patio de comidas:

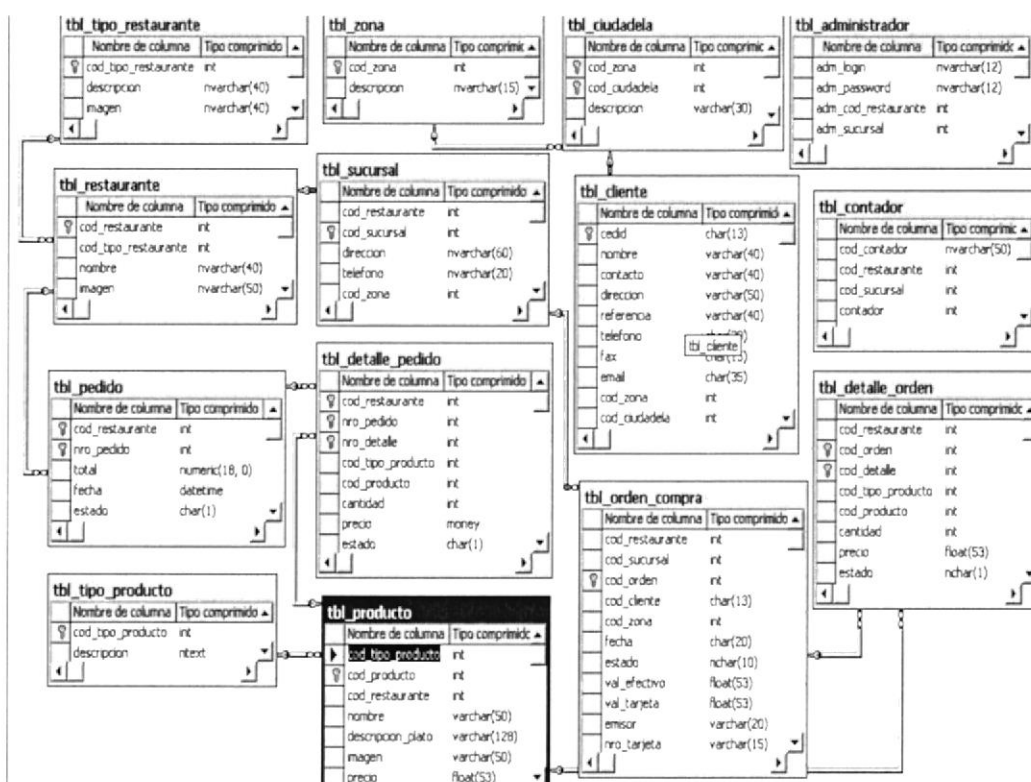


Figura 5.2 Modelo Físico de datos.

A continuación se detalla la funcionalidad de cada tabla y de los elementos que componen las tablas, y los tipos de datos asignados a los mismos:

Tipo Restaurante: (tbl_tipo_restaurante).

En esta tabla se almacena información referente a los tipos de restaurantes, y los gráficos de cada uno; los mismos que aparecerán en el catálogo online, cuando el consumidor ingrese al site.

Zona (tbl_zona):

En esta tabla se almacena información de las zonas en que se configura la ciudad local: Norte, Sur , Centro etc.

Ciudadela (tbl_ciudadela) :

Se registrarán las ciudadelas que conforman la zona en la cual se configure el restaurante afiliado al patio de comidas.

Administrador (tbl_administrador) :

Guarda registro de los administradores de los pedidos que se envían a cada uno de los restaurantes y por sucursal de tal manera que se puedan controlar y despachar prontamente.

Restaurante (tbl_restaurante) :

En esta estructura se almacenará los restaurantes que se afilien al servicio de patio de comidas.

Sucursal (tbl_sucursal) :

Almacenará registro de cada uno de las sucursales que aperturen cada uno de los restaurantes.

Ciente (tbl_cliente) :

Esta tabla servirá para almacenar los consumidores finales que visiten el site y soliciten pedidos en cualquiera de los restaurantes.

Pedido (tbl_pedido) :

Se registra las solicitudes de comidas a domicilio que efectuan los consumidores finales en cualquiera de los restaurantes.

Detalle Pedido (tbl_detalle_pedido) :

Se graban cada uno de los items que compone el pedido solicitado por el consumidor final en cualquiera de los restaurantes del patio de comidas.

Tipo Producto (tbl_tipo_producto) :

Corresponde a almacenar en esta estructura la clasificación de las comidas a saber : plato fuerte , combo, ingrediente , bebida; y así tener de manera dinámica estratificados los alimentos que expenden en cada restaurante.

Producto (tbl_producto) :

Consiste en el catálogo de comidas clasificadas por tipo, las mismas que se configuran dependiendo de cada restaurante.

Orden Compra (tbl_orden_compra):

En esta tabla se registran los pedidos solicitados por los consumidores finales , una vez aceptado el mismo para luego recibir la confirmación del mismo.

Detalle de Orden Compra (tbl_detalle_orden_orden) :

Se registra cada uno de los items que componen la orden de compra solicitada por el consumidor final.

Contador(tbl_contador):

Se graba los contadores , que consisten en las diferentes secuencias que se manejen en el site ; por cada uno de las sucursales y restaurantes configurados en el patio de comidas.

Los campos y tipos de datos se pueden visualizar en el gráfico 5.2 adjunto a esta sección.

5.3 Diagrama Entidad-Relación

En esta parte se visualiza el diagrama relacional de cada una de las tablas creadas para la construcción del sitio :

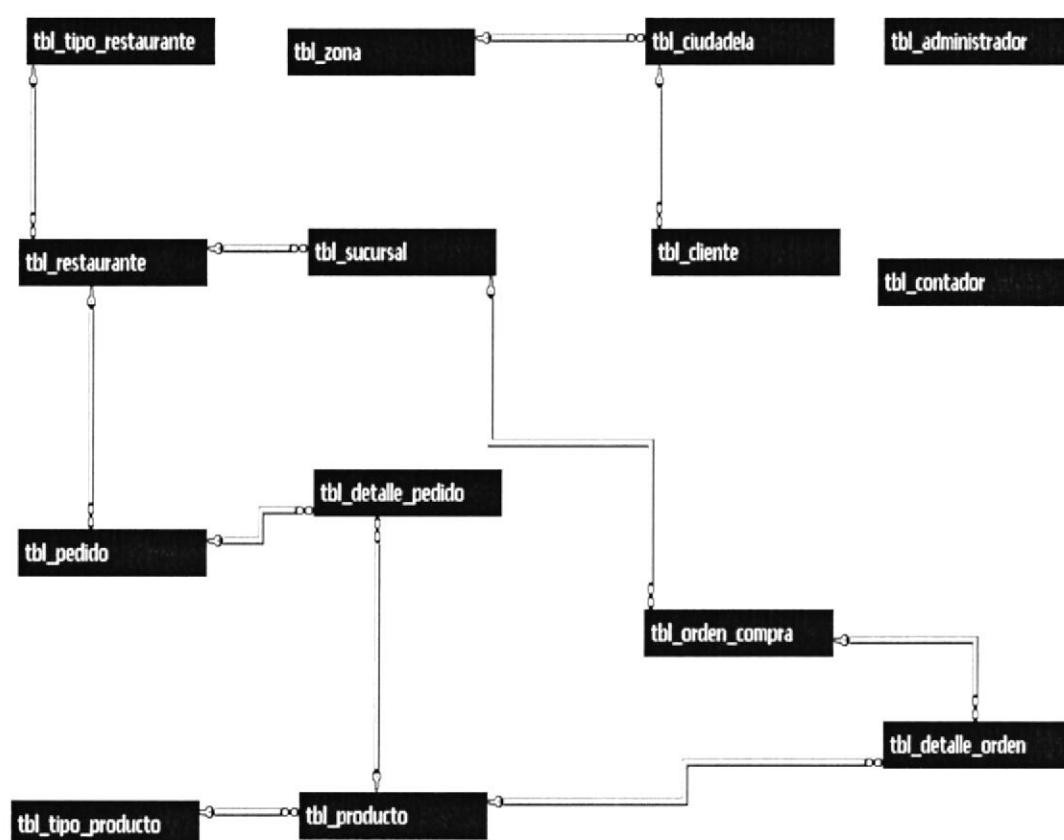


Figura 5.3 Modelo Entidad –Relacions

5.4 Proceso Servidor de base de datos.

Para la construcción del sitio propuesto hemos utilizado el manejador de base de datos Sql Server v.7. la cual consiste en un manejador RDBMS que permite diseñar objetos que se administran en un conjunto de herramientas propias de este producto. Nuestro site no contiene procedimientos almacenados ya que los procesos son manejados por el TPS monitor , revise más información de esto en la sección 4.3. del presente manuscrito.

CAPITULO 6

- ***IMPLEMENTACION DE LA CAPA
DE NEGOCIOS***

Implementación de la Capa de Negocios

El nivel de la lógica del negocio es el corazón de la aplicación, donde son mantenidas las reglas del negocio y del procesamiento específico de aplicación. La lógica de negocio colocada en componentes es el puente entre los niveles del ambiente del cliente y datos.

6.1.- Estructura de los componentes de acceso a datos

Los componentes de acceso a datos interactúan directamente con la base de datos, realizan las operaciones de mantenimiento (select, insert, update, delete).

A continuación detallaremos los componentes de datos utilizados en nuestra aplicación:

Componente	Descripción
Dbtiporestaurante	Permite ingresar y consultar los diferentes tipos de restaurantes que determinarán el tipo de comida. Hace referencia a la tbl_tipo_restaurante
Dbrestaurante	Permite dar mantenimiento a la tabla tbl_restaurante
Dbciudadela	Permite ingresar y consultar restaurantes. Hace referencia a la tabla tbl_restaurante
Dbzona	Permite ingresar y consultar zonas. Hace referencia

	a la tabla tbl_zona
Db sucursales	Mantenimiento de la tabla tbl_sucursales
Componente	Descripción
Db cliente	Manipula la información contenida en la tabla maestro tbl_cliente
Db producto	Mantenimiento de la tabla maestro tbl_producto que constituye el catálogo principal
Db tipo producto	Permite dar mantenimiento a la tabla tbl_tipo producto
Db pedido	Permite el ingreso de datos y consulta a la tabla transaccional tbl_pedido
Db detalle pedido	Permite el ingreso de datos y consulta a la tabla transaccional tbl_detalle pedido
Db orden compra	Permite el ingreso de datos y consulta a la tabla transaccional tbl_orden compra
Db detalle orden	Permite el ingreso de datos y consulta a la tabla transaccional tbl_detalle orden
Db contador	Permite dar mantenimiento a la tabla tbl_contador, que almacena los números secuenciales de las demás tablas.

6.2.- Objetos distribuidos de acceso a datos

Una vez descritos los componentes que se encargan de la manipulación directa de los datos (mantenimientos y consultas) en este punto se enfocarán los módulos de clase de cada componente que son denominados *objetos* una vez que el componente ha sido compilado.

Los objetos incluyen toda la funcionalidad del componente.

Componente : Dbtiporestaurante

Objeto	Métodos	Funcionalidad
tiporestaurante	Add	Agrega un tipo de restaurante
	Upd	Actualiza la información de un tipo de restaurante en particular
	Del	Elimina físicamente un tipo de restaurante.
	List	Lista los tipos de restaurantes registrados en el sistema

componente : Dbrestaurante

Objeto	Métodos	Funcionalidad
restaurante	Add	Agrega un restaurante
	Upd	Actualiza la información de un restaurante
	Del	Elimina físicamente un registro de restaurante.
	List	Lista los restaurantes registrados en el sistema
	Listbytipo	Lista los restaurantes por tipo de comida
	Getbyid	Consulta específica de restaurante por código

componente : Dbciudadela

Objeto	Métodos	Funcionalidad
ciudadela	Add	Agrega un registro ciudadela
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Getbyid	Consulta de ciudadela por código
	Getbyzona	Consulta de ciudadelas por zona

componente : Dbzona

Objeto	Métodos	Funcionalidad
Zona	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	List	Lista las zonas registradas.
	Getbyid	Consulta de zona por código

componente : Dbsucursal

Objeto	Métodos	Funcionalidad
Sucursal	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Getbyid	Consulta de sucursales por código
	Listbyres	Listado de sucursales por restaurantes

componente : Dbcliente

Objeto	Métodos	Funcionalidad
Cliente	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Getbyid	Consulta de clientes por cédula

componente : Dbproducto

Objeto	Métodos	Funcionalidad
Producto	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Getbyid	Consulta de productos por código
	Listbytipores	Lista productos por tipo de producto y restaurante

componente : Dbtipoproducto

Objeto	Métodos	Funcionalidad
Tipoproducto	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	List	Lista los productos por tipo de producto

componente : Dbpedido

Objeto	Métodos	Funcionalidad
Pedido	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Getbyid	Consulta de pedidos por número

componente : Dbdetallepedido

Objeto	Métodos	Funcionalidad
Detallepedido	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Listbypedido	Lista detalle por número de pedido

componente : Dbordencompra

Objeto	Métodos	Funcionalidad
ordencompra	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Getbyid	Lista los tipos de restaurantes registrados en el sistema
	Listbyressuc	Listado de órdenes de compra por restaurante y sucursal
	listbycliente	Listado de órdenes de compra por cliente
	listbyfecha	Listado de órdenes de compra por fecha

componente : Dbdetalleorden

Objeto	Métodos	Funcionalidad
detalleorden	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Listbyord	Lista detalle por número de orden

componente : Dbcontador

Objeto	Métodos	Funcionalidad
Contador	Add	Agrega un registro
	Upd	Actualiza un registro específico.
	Del	Elimina físicamente un registro.
	Getbyid	Consulta de contador por código de restaurante y sucursal

6.3.- Comunicación con el proceso servidor de base de datos

La comunicación con el proceso servidor de base de datos se realiza a través del middleware ADO, que facilita la comunicación con el OLEDB data provider, que en nuestro caso es SQL 7.0.

6.4.- Estructura de los componentes de negocios

Los componentes de negocio permiten la comunicación con los componentes de datos, un componente de negocio puede implementar uno o más componentes de datos.

A continuación detallaremos los componentes de negocios utilizados en nuestra aplicación:

Componente	Descripción
Btiporestaurante	Trabaja con los componentes de datos:dbtiporetaurante y dbcontador
Brestaurante	Trabaja con los componentes de datos:dbrestaurante y dbcontador
Bciudadela	Trabaja con los componentes de datos:dbciudadela y dbcontador
Bzona	Trabaja con los componentes de datos:dbzona y dbcontador
Bsucursal	Trabaja con los componentes de datos:dbsucursal y dbcontador
Bcliente	Trabaja con los componentes de datos:dbciudadela y dbzona y dbcliente
Bproducto	Trabaja con los componentes de datos:dbproducto y dbcontador
Btipoproducto	Trabaja con los componentes de datos:dbtipoproducto y dbcontador

Bpedido	Trabaja con los componentes de datos: dbpedido y dbcontador
Bdetallepedido	Trabaja con los componentes de datos: dbdetallepedido y dbcontador
Bordencompra	Trabaja con los componentes de datos:dbordecompra y dbcontador
Bdetalleorden	componentes:dbdetalleorden y dbcontador

6.5.- Objetos distribuidos de Negocios

A continuación detallamos las interfaces implementadas en el proyecto:

_CNTIPORESTAURANTE

```
interface _CNTipoRestaurante : IDispatch {
    [id(0x60030000)]
    HRESULT Listar([out, retval] _Recordset** );

    HRESULT Insertar(
        [in, out] BSTR* pv_descripcion,
        [in, out] BSTR* pv_imagen,
        [out, retval] unsigned char* );

    HRESULT Actualizar(
        [in, out] short* pn_cod_tipo_restaurante,
```

```
[in, out] BSTR* pv_descripcion,  
[in, out] BSTR* pv_imagen,  
[out, retval] unsigned char* );
```

```
HRESULT Eliminar(  
    [in, out] short* pn_cod_tipo_restaurante,  
    [out, retval] unsigned char* );
```

```
};
```

_CNRESTAURANTE

```
interface _CNRestaurante : IDispatch {
```

```
HRESULT ListarXTipo(  
    [in, out] VARIANT* pn_cod_tipo_restaurante,  
    [out, retval] _Recordset** );
```

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

```
HRESULT Insertar(  
    [in, out] short* pn_cod_tipo_restaurante,  
    [in, out] BSTR* pv_nombre,  
    [in, out] BSTR* pv_imagen,  
    [out, retval] unsigned char* );
```

```
HRESULT Consultar(  
    [in, out] short* pn_cod_restaurante,
```

```
[out, retval] _Recordset** );
```

```
HRESULT Actualizar(
```

```
    [in, out] short* pn_cod_restaurante,
    [in, out] short* pn_cod_tipo_restaurante,
    [in, out] BSTR* pv_descripcion,
    [in, out] BSTR* pv_imagen,
    [out, retval] unsigned char* );
```

```
HRESULT Eliminar(
```

```
    [in, out] short* pn_cod_restaurante,
    [out, retval] unsigned char* );
```

```
};
```

_CNSUCURSAL

```
interface _CNSucursal : IDispatch {
```

```
    HRESULT Consultar(
```

```
        [in, out] short* pn_cod_sucursal,
        [out, retval] _Recordset** );
```

```
    HRESULT ConsultarXRest(
```

```
        [in, out] short* pn_cod_restaurante,
        [out, retval] _Recordset** );
```

```
    HRESULT ConsultarXRestZona(
```

```
        [in, out] VARIANT* pn_cod_restaurante,
```



```
[in, out] VARIANT* pn_zona,  
[out, retval] _Recordset** );
```

```
HRESULT Insertar(
```

```
[in, out] short* pn_cod_restaurante,  
[in, out] BSTR* pv_direccion,  
[in, out] BSTR* pv_telefono,  
[in, out] short* pn_cod_zona,  
[out, retval] unsigned char* );
```

```
HRESULT Actualizar(
```

```
[in, out] short* pn_cod_sucursal,  
[in, out] short* pn_cod_restaurante,  
[in, out] BSTR* pv_direccion,  
[in, out] BSTR* pv_telefono,  
[in, out] short* pn_cod_zona,  
[out, retval] unsigned char* );
```

```
HRESULT Eliminar(
```

```
[in, out] short* pn_cod_res,  
[in, out] short* pn_cod_sucursal,  
[out, retval] unsigned char* );
```

```
};
```

_CNTIPOPDUCTO

```
interface _CNtipoproducto : IDispatch {

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

    HRESULT Consultar(
        [in, out] VARIANT* P_COD_TIPO,
        [out, retval] _Recordset** );

    HRESULT Actualizar(
        [in, out] short* pn_cod_tipo,
        [in, out] BSTR* pv_descripcion,
        [out, retval] unsigned char* );

    HRESULT Eliminar(
        [in, out] short* pn_cod_tipo,
        [out, retval] unsigned char* );

    HRESULT Insertar(
        [in, out] BSTR* pv_descripcion,
        [out, retval] unsigned char* );
};
```

_CNPRODUCTO

```
interface _CNProducto : IDispatch {

    HRESULT ConsultarXTipoyRes(
```

```
[in, out] VARIANT* PN_COD_TIPO,  
[in, out] VARIANT* PN_COD_RES,  
[out, retval] _Recordset** );
```

HRESULT Insertar(

```
[in, out] VARIANT* PN_COD_TIPO,  
[in, out] short* PN_COD_RES,  
[in, out] VARIANT* PV_NOMBRE,  
[in, out] VARIANT* PV_DESC,  
[in, out] BSTR* PV_IMAGEN,  
[in, out] double* PN_PRECIO,  
[out, retval] unsigned char* );
```

HRESULT Consultar(

```
[in, out] VARIANT* pn_cod_producto,  
[out, retval] _Recordset** );
```

6.6.- Comunicación entre Obletos Distribuidos

Windows recogerá todos los parámetros de la función, y utilizando las bondades inter-proceso del sistema operativo, copiará los valores de parámetro al proceso que contenga los códigos y los datos de implementación del objeto.

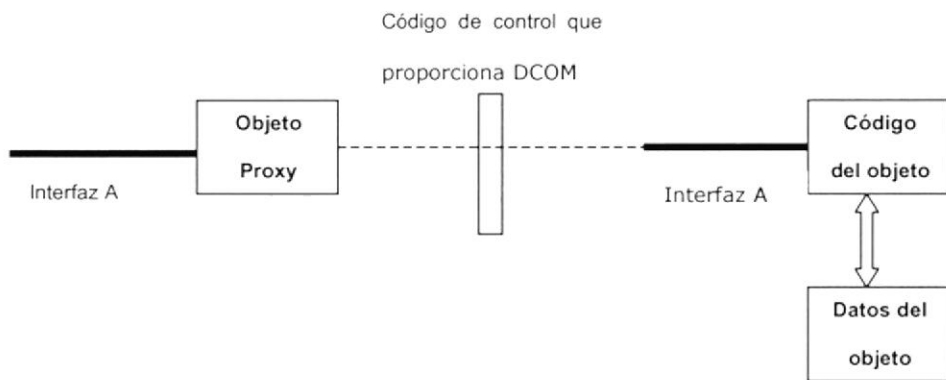


Fig.6.6.1 Cómo los objetos recorren el espacio de proceso

La ventaja principal de la automatización remota se da, cuando el sistema remoto ya contiene los datos necesarios para funcionar. La estructura de control de pocas llamadas de función puede ser insignificante con la alternativa de transferir megabytes a través de la red.

6.7.- Proceso Servidor Transaccional

Como mencionamos en el capítulo 4, utilizamos el tp monitor (Transaccion processing server) para administrar y controlar las transacciones.

Las transacciones que se procesan en nuestra aplicación son :

Transacción (Cliente Stand-alone)	Componente de negocio	Componente de datos
Ingreso de restaurantes	Brestaurante	Dbrestaurante
Ingreso de tipo de restaurantes	Btiporestaurante	dBtiporestaurante
Ingreso productos	Bproducto	dBproducto
Ingreso tipo_productos	Btipoproducto	dBtipoproducto
Ingreso sucursales	Bsucursales	dBsucursales
Ingreso zona	Bzona	dBzona
Ingreso de ciudadelas	bciudadela	dbciudadela

Transacción (Cliente on-line)	Componente de negocio	Componente de datos
Presentar tipo comida	Brestaurante	Dbrestaurante
Presentar tipo restaurante	Btiporestaurante	dBtiporestaurante

Presentar productos	Bproducto	dBproducto
Elaborar pedido	Btipoproducto	dBtipoproducto
Realizar orden	Bsucursales	dBsucursales

Transacción (Administración on-line)	Componente de negocio	Componente de datos
Autenticación del usuario	Brestaurante	Dbrestaurante
Consulta de pedidos	Btiporestaurante	dBtiporestaurante
Consulta de ventas	Bproducto	dBproducto
Consulta de clientes por sucursal	Btipoproducto	dBtipoproducto

CAPITULO 7

- ***IMPLEMENTACION DE LA CAPA
DE PRESENTACION***

Implementación de la Capa de Presentación.

La capa de presentación constituye el “front end” del proyecto, es decir es la capa que interactúa directamente con el visitante, mostrándole las diferentes alternativas que se pueden manejar. Para su implementación se han aplicado tecnologías comunes en el ambiente de internet, como lo son : DreamWeaver , Flash (ambos de Macromedia) y lenguaje HTML para la definición de estilos y formatos. Esta capa es de gran importancia porque constituye la imagen visual de nuestro servicio.



7.1. Diseño del Sitio Web-Cliente Basado en Browser.

En esta sección se demostrará cómo el navegante puede acceder a las ventanas del patio de comidas. A continuación una explicación gráfica

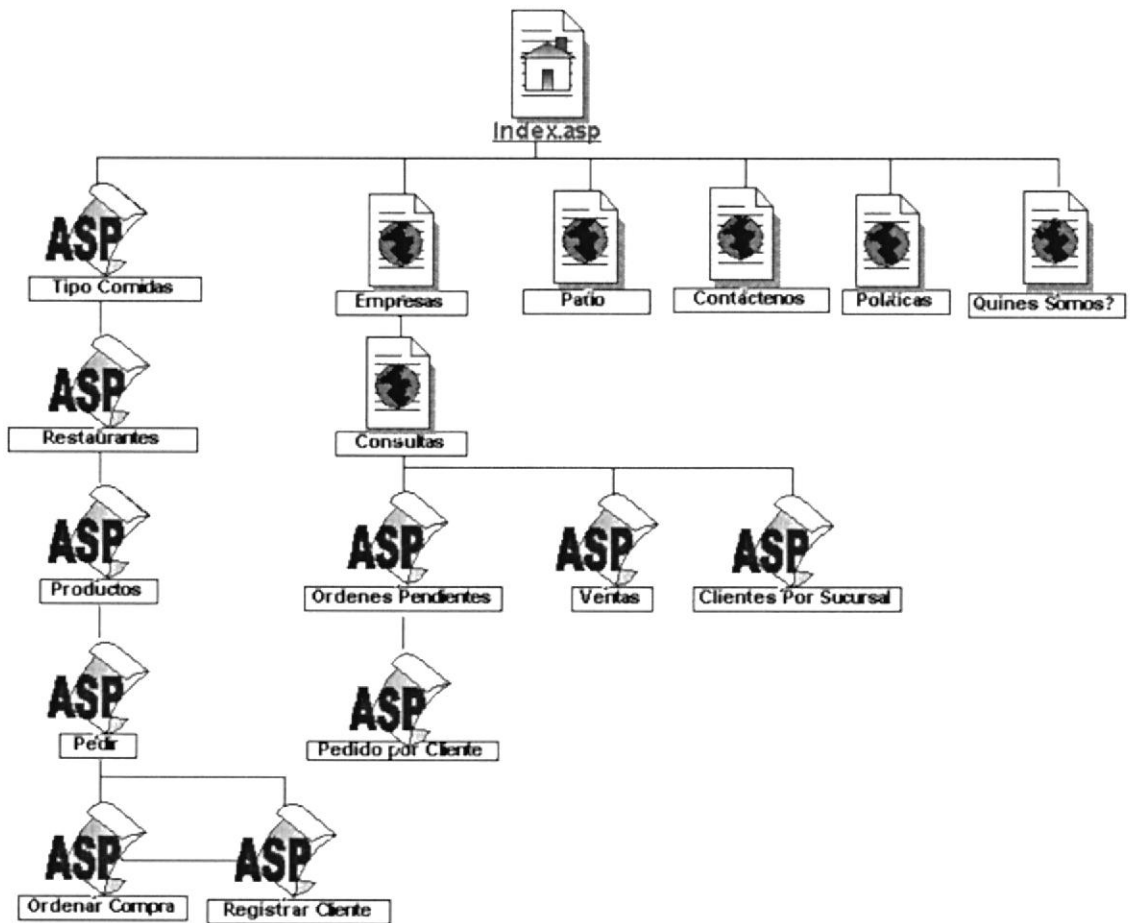
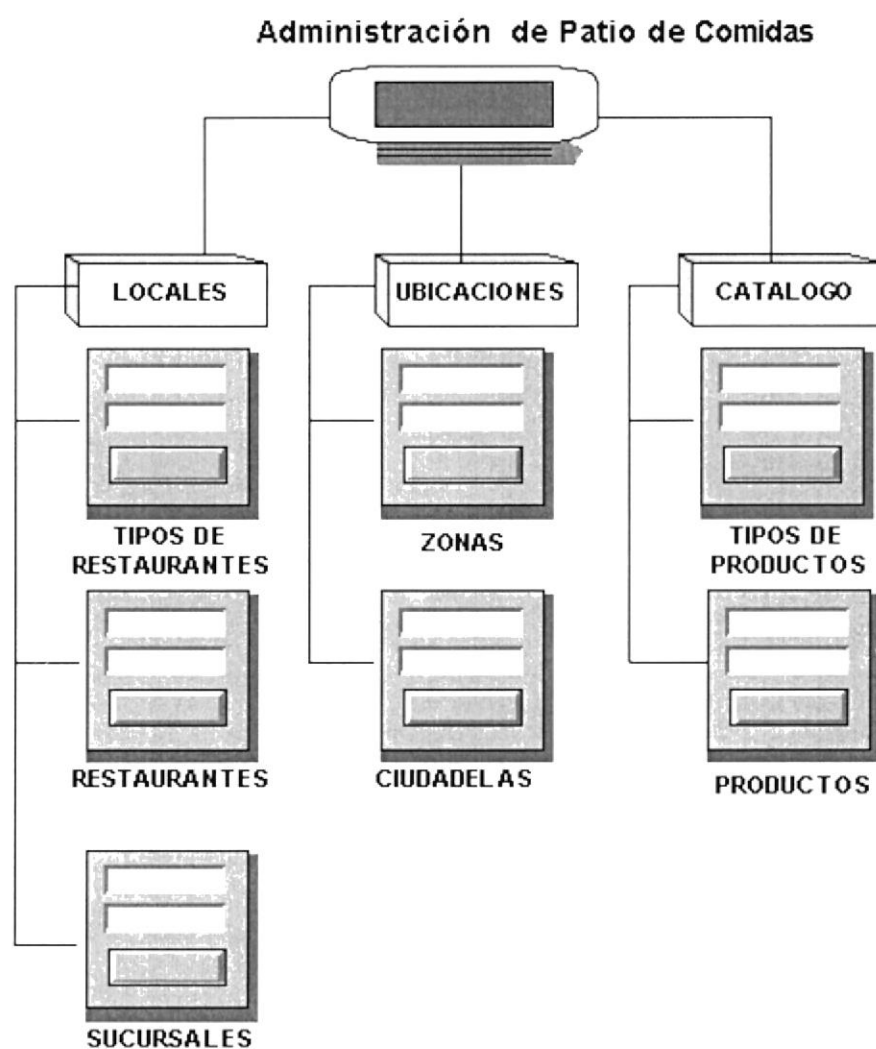


FIGURA. 7.1. Diagrama Topdown de Sistema Stand-alone.

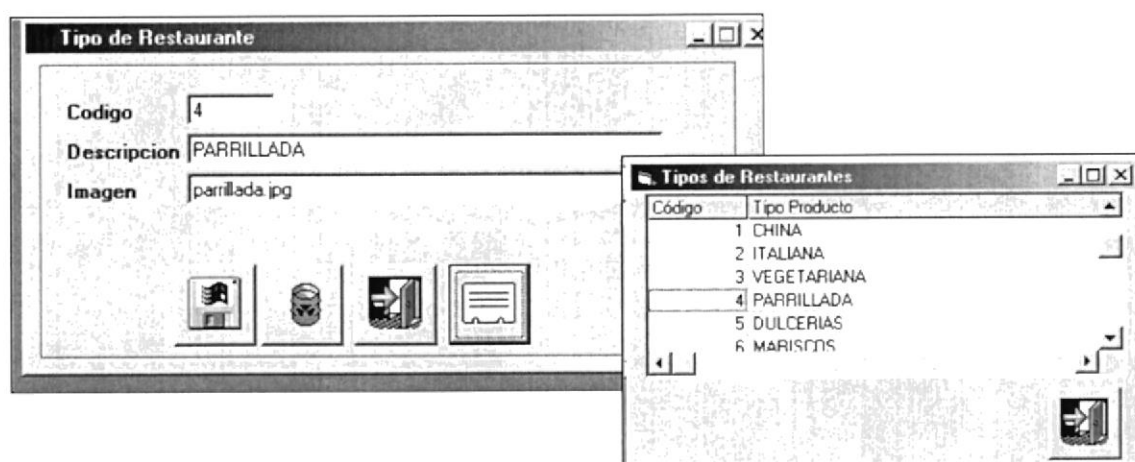
7.2.- El Cliente Stand-alone.

La aplicación de Administración del Patio de Comidas, permite configurar las localidades afiliadas a nuestro sitio, ubicaciones geográficas, tanto de restaurantes como de los usuarios finales que compran. A continuación una explicación gráfica de la distribución del mismo.



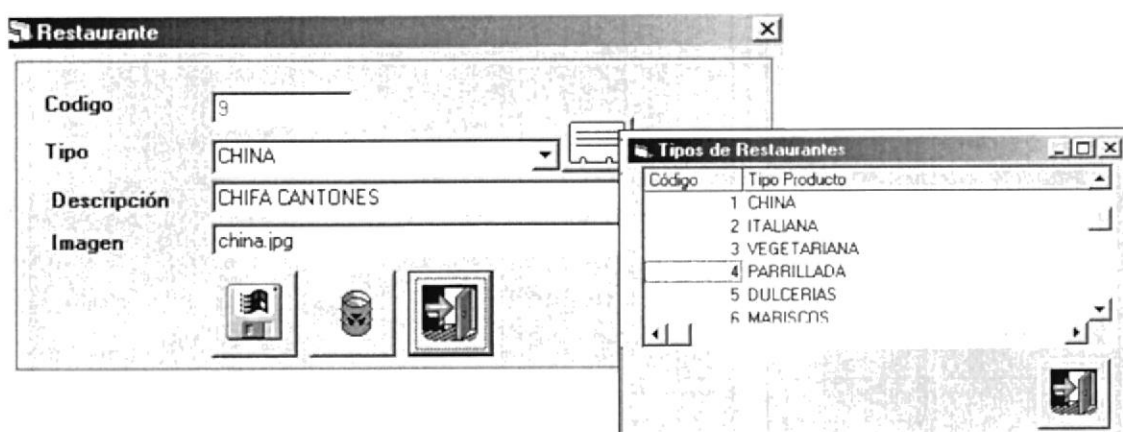
Mantenimiento de Tipos de Restaurantes.-

El mantenimiento de Tipos de Restaurante permite ingresar un configurar el sitio para cualquier grupo de restaurantes que se identifiquen con un tipo de comida específica.



Mantenimiento de Restaurantes.-

Los Restaurantes se ingresan según el tipo de comida que expendan, pues es más fácil identificarlos de esta manera.



Mantenimiento de Sucursales.-

Las sucursales se registran por cada restaurante afiliado a nuestro patio de comidas, de tal manera se forma una cadena de restaurantes los cuales ofrecerán sus servicios a la una mayor cantidad de zonas de la ciudad, ya que la configuración de las mismas permite identificar a que zona geográfica pertenece.

Mantenimiento de Sucursales

Sucursal: 2

Restaurante: CHIFA CANTONES Zona: NORTE

Dirección: ALBORADA

Teléfono: 2345443

DEL

Mantenimiento de Zonas.-

En esta pantalla se configuran las zonas geográficas de la ciudad, las cuales nos sirven para clasificar las sucursales y los clientes. Pues si un cliente vive en el norte, la sucursal que recibe su orden será del norte.

Zona

Cod Zona: 1

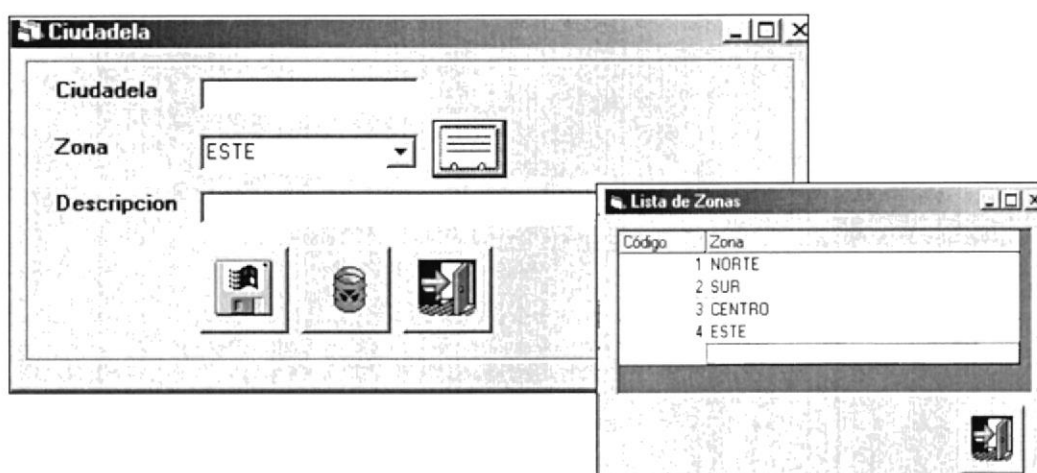
Descripción: NORTE

Lista de Zonas

Código	Zona
1	NORTE
2	SUP
3	CENTRO
4	ESTE

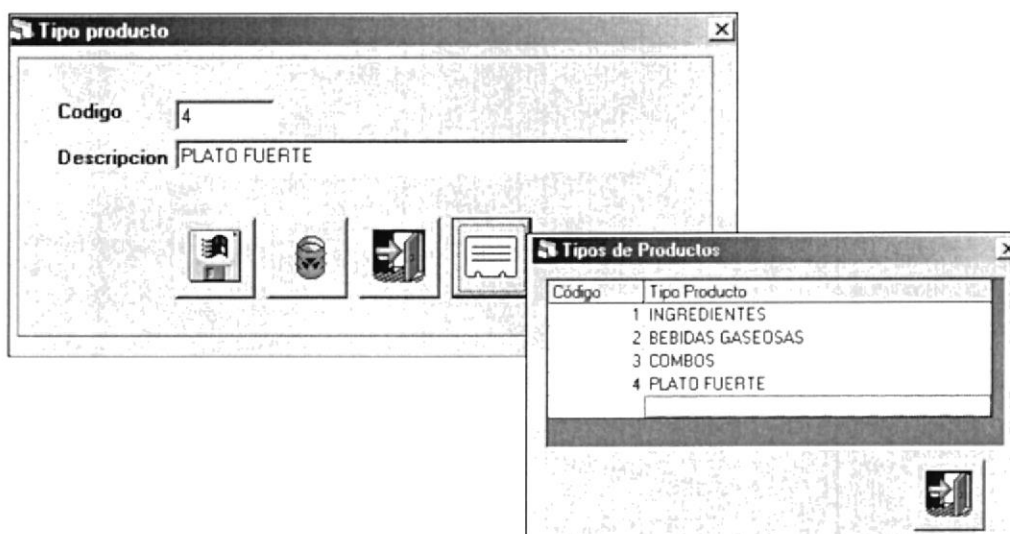
Mantenimiento de Ciudades.-

El mantenimiento de las ciudades permite clasificar las mismas por zona geográfica, de esta manera se cubre una mayor cantidad de lugares a los cuales se dará servicio por parte de los restaurantes que tengan sucursales en las zonas donde pertenezcan dichas ciudades.



Mantenimiento de Tipo de Productos.-

Los tipos de productos son configurados para poder clasificar el menú de un determinado restaurante.



Mantenimiento de Productos.-

En esta pantalla se realiza el mantenimiento de los productos, configurándolos por Tipo de Producto y Restaurante, ya que de esta manera se pueden clasificar los menús de los restaurantes.



The screenshot shows a window titled "Producto" with the following fields and controls:

- Producto:** Text input field containing "15".
- Tipo Producto:** Dropdown menu with "PLATO FUERTE" selected.
- Restaurante:** Dropdown menu with "CANTONES" selected.
- Nombre:** Text input field containing "TALLARIN CANTONES".
- Descripcion:** Text input field containing "TALLARIN CON PEDACIDOS DE POLLO, CAMARON, CERDO,".
- Imagen:** Text input field containing "tallacanto.jpg" and a folder icon for file selection.
- Precio:** Text input field containing "3.80".

At the bottom of the window, there are four icons: a floppy disk (save), a trash can (delete), a folder with an arrow (upload), and a speech bubble (help).

7.3. HTML

Página: Quienes somos?.- QUIENES.HTM

Esta página muestra claramente una definición de la empresa que administra el patio de comidas, cual es su misión y objetivos para con los clientes y usuarios.



Página: Patio.- PATIO.HTM.

Presenta una breve explicación de cómo funciona el sitio, para de esta manera orientar al usuario final de la metodología utilizada y del proceso a seguir al momento de ordenar su pedido.



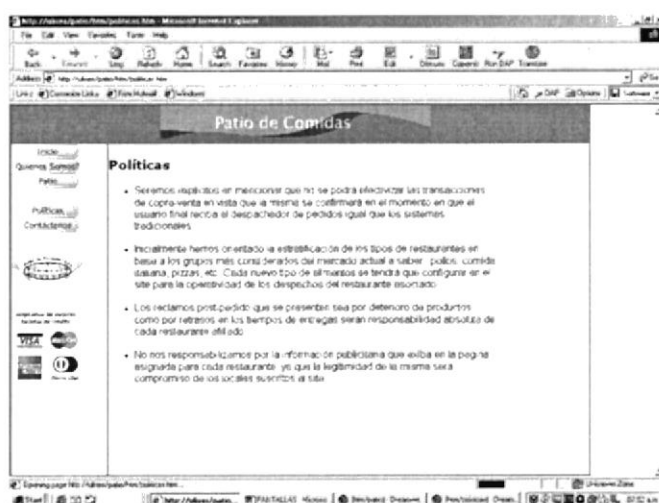
Página: Contáctenos.-CONTACTO.HTM.

Presenta la dirección de nuestras, esta orientada a los distintos restaurantes de la ciudad que les interese registrarse en nuestro patio de comidas virtual.



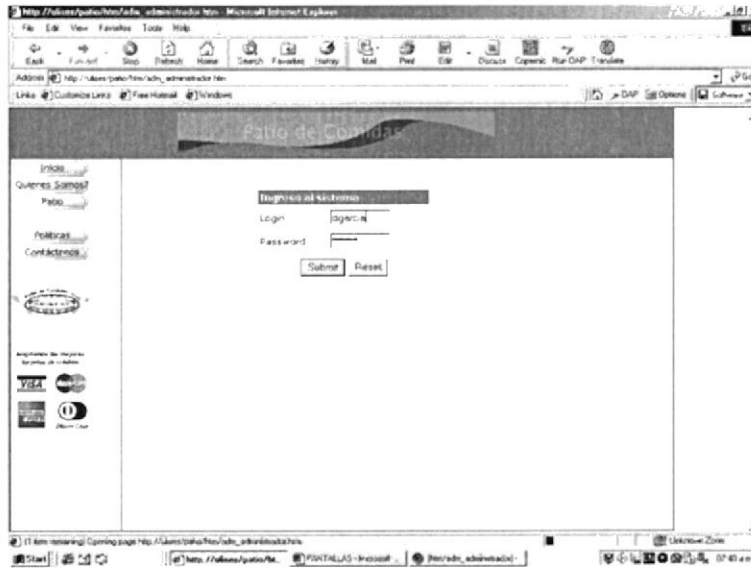
Página: Políticas.-POLITICAS.HTM

Presenta las reglas del negocio de la empresa que administra el patio de comidas. Las restricciones al momento de utilizar nuestros servicios de enlace con los distintos restaurantes de nuestro patio.



Página: Empresas.-ADM_ADMINISTRADOR.HTM

Permite que los representantes de las sucursales de los restaurantes afiliados al patio de comidas ingresen a realizar las diferentes consultas que proporciona el sitio.



7.4. Client Script –DHTML.

CLIENTE_INGRESO

function checkOption,

Valida que los productos en la página anterior estén chequeados, pues solo esos serán pedidos

function CheckAll,

Valida que todos los datos en el ingreso de clientes estén llenos.

RESTAURANTE_COMIDAS.

function ValidaNumeros,

Valida que los datos que se ingresen el cantidad de productos, sean solo números.

ADM_ADMINISTRADOR

function ValidaPassword,

Valida que el password del usuario que intenta realizar las diferentes consultas del sitio sea el correcto.

7.5. Server Script –ASP.

Tipo de Comidas.- PATIODECOMIDAS.ASP.

Muestra un listado de los tipos de Restaurantes o comidas que están registrados en nuestro site. Esta página implementa código ASP para listar los tipos de restaurantes de la base de datos.



Restaurantes.-PATIORESTAURANTE.ASP.

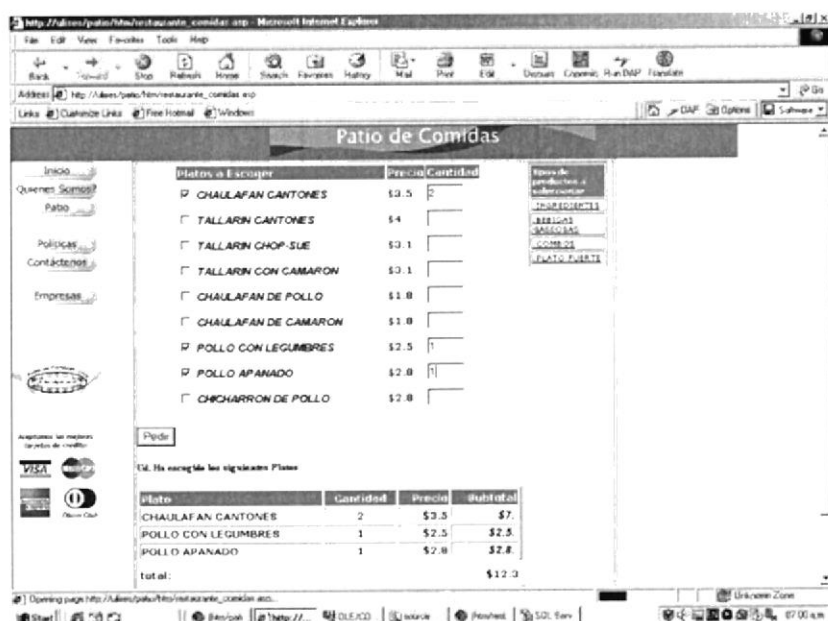
Muestra un listado de los restaurantes clasificados por tipos. Esta página implementa código ASP para consultar los restaurantes según el tipo de restaurante se haya elegido el usuario en la página patiodecomidas.asp.



Productos.-RESTAURANTE_COMIDAS.ASP.

Muestra y permite escoger del catálogo de los productos que ofrece cada restaurante, los cuales también se clasifican por tipos de producto. Esta página implementa código ASP para listar los productos del menú del restaurante escogido en la página patiorestaurant.asp según el tipo de producto que el cliente prefiera, al momento de presentar el primer menú se toma como parámetro para la consulta un tipo de producto cualquiera, si el usuario desea escoger otro, la consulta se vuelve a realizar con los nuevos parámetros.

Cuando el cliente escoge sus productos por cada tipo de producto, debe realizar el pedido, en este momento se implementa código asp, para grabar su pedido con sus respectivos detalles.



El cliente puede pedir los productos que él requiera, y escoger el menú de un tipo de producto cualquiera hasta cuando él tome la decisión de aceptar su pedido. Aquí se llama a otra página asp llamada `cliente_registrado.asp`.

Envio Orden .-CLIENTE_REGISTRADO.ASP.

Si el usuario ya esta registrado en el patio de comidas simplemente debe especificar la forma como va a cancelar su orden y registrarla, en ese momento se ejecuta código asp

para grabar la orden de compra con sus respectivos detalles, tomando en cuenta la ciudadela y zona en la que vive el cliente, para de esta manera rutear la orden de compra a la sucursal mas cercana del restaurante escogido.

Si el usuario aún no es cliente del patio de comidas, tiene opción para Registrarse, en ese momento se llama a otra pagina asp, Cliente_registro.asp .

The screenshot shows a web browser window with the title 'Patio de Comidas'. The main content area contains a registration form with the following sections:

- Datos del Cliente:** A text input field with the value '0317441158'.
- Datos de Pago:** A radio button for 'Efectivo' and a radio button for 'Tarjeta' (which is selected).
- Tarjeta:** A dropdown menu showing 'Mastercard' and a text input field with the value '5387626255367474'.

At the bottom of the form are two buttons: 'Registrar' and 'Limpiar'. The browser's address bar shows the URL: 'http://192.168.1.100/Cliente_registro.asp?ord_pedido=0'.

Registrar Clientes.-CLIENTE_INGRESO.ASP

Si el usuario final es un cliente nuevo de nuestro site, debe ingresar sus datos personales para registrarse. En esta página se implementa código asp para grabar los datos del cliente y además grabar la orden de compra con sus respectivos detalles, tomando en cuenta la ciudadela y zona en la que vive el cliente, para de esta manera rutear la orden de compra a la sucursal mas cercana del restaurante escogido.

Patio de Compras

Inicio
Quiénes Somos
Patio
Políticas
Contactenos

Asociados de negocios
tarjetas de crédito

VISA
MasterCard
American Express

Datos del Cliente

Zona: NORTE
Ciudadela: Alborada
Cadula: 597766417
Nombre: FABRICO RENOSO
Contacto: FABRICO
Dirección: ALBORADA 4 M7 121 VILLA 6
Referencia: FRENTE A MI COMERCATO
Telefono: 34544
Fax:
Email: fabrico_renosof@hotmail.com

Datos de Pago

Efectivo

Tarjeta

Tarjeta abiente:
Tarjeta: Mastercard

Ordenes Pendientes.- ADM_REP_ORDEN.ASP

Muestra un listado de las órdenes pendientes por despachar, esta consulta utiliza código asp para consultar las órdenes pendientes por sucursal.

Restaurante: CHIFA CANTONES
Sucursal: SAMANES
Zona: NORTE
Telefono: 7777

Ordenes de Pedidos

ID	Cliente	Fecha	Valor	Efectivo/Tarjeta
1	KARINA FAZAMINO	26/06/2002	\$100	Efectivo
2	JUAN JOSE PENAFIEL	26/06/2002	\$100	Efectivo
3	KARINA FAZAMINO	26/06/2002	\$100	Efectivo
4	Angel Fuentes	26/06/2002	\$100	Efectivo
5	Daniel Garcia	26/06/2002	\$8	Tarjeta
6	JUAN JOSE PENAFIEL	26/06/2002	\$8	Tarjeta
7	KARINA FAZAMINO	26/06/2002	\$10	Tarjeta
8	JUAN JOSE PENAFIEL	26/06/2002	\$8	Tarjeta
9	Daniel Garcia	26/06/2002	\$8	Tarjeta
10	JUAN JOSE PENAFIEL	26/06/2002	\$8	Tarjeta

Página 1 de 3

Pedio Por Client.-

Muestra el detalle de lo que se ha pedido en una determinada orden. Esta página implementa código asp para consultar los detalles de una orden de compra pendiente luego de haber consultado el listado de todas las órdenes de la sucursal.



The screenshot shows a web browser window with the URL `http://alamos/patio/Menu/Info.asp?det_ordem.asp?ord=30`. The page title is "Patio de Comidas". On the left side, there are navigation links for "Inicio", "Quienes Somos", "Patio", "Horarios", and "Contactenos". Below these are logos for "CANTONES", "POLLO", and "AFANADO".

The main content area displays the following order details:

Por: Orden:	30
Cliente:	MADINA FACAMENDO
Dirección:	ALBORADA ORL. M2 913
Ciudad:	VILLA 4
Email / Teléfono:	maribamolina@hotmail.com
Zona:	NORTE
Fecha:	08/07/2002
Valor:	12.9
Estado:	Pendiente

Below the details is a table showing the items ordered:

Plato	Tipo Plato	Cantidad	Precio	Subtotal
CARILLAFAD CANTONES	PLATO FUERTE	2	\$3.5	\$7
POLLO CON LEGUMBRES	PLATO FUERTE	1	\$2.5	\$2.5
POLLO AFANADO	PLATO FUERTE	1	\$2.9	\$2.9
Total:				\$12.9

7.6 .- CSS.

Estilos.css

Productos

Font: Arial,Helvética,sans-serif

Size : 14

Style:Italic

Color: #000000

TipoProducto

Font: Georgia,Times New Roman

Size : 18

Style:Italic

Color: #FF3300

TextBox

Font: Verdana, Arial

Size : 12

Style:None

Color: #000000

7.7.- Comunicación con componentes de negocios.

PAGINA ASP	COMPONENTE
PATIODECOMIDAS.ASP	BTIPORESTAURANTE
PATIORESTAURANTE.ASP	BRESTAURANTES
RESTAURANTECOMIDAS.ASP	BPEDIDO
	BDETALLEPEDIDO
	BTIPOPRODUCTO
	BPRODUCTO
CLIENTE_INGRESO.ASP	BZONA
	BCIUDELA
	BCLIENTE
	BSUCURSAL
	BORDENCOMPRA
	BDETALLEPEDIDO

7.8.- Proceso Servidor Web.

El web server es el encargado de atender los requerimientos del browser (cliente), ya sea para solicitar recursos (imágenes, vídeo, etc) o páginas html generadas dinámicamente (ASP)

El web server implementado en este site, es el Internet Information Server (IIS) desarrollado por Microsoft, he implementado en arquitectura NT.

Dentro de los requerimientos que maneja nuestro web server tenemos:

- Generación dinámica de páginas html, a través de un motor ASP que se encarga de interpretar el server script.
- Mantiene el estado a nivel de usuario y a nivel de aplicación usando objetos, venciendo las limitaciones de http.

Conclusiones

Queremos concluir mencionando que la utilización del servicio que ofrecemos brindará una nueva forma de servicio a los negocios de comidas que despachen a domicilio pedidos particulares, esto de la mano con el apoyo de parte de los restaurantes, al mantener constatemente actualizado su catálogo de productos y manteniendo un estándar de calidad que fortalezca la confianza en el nuevo servicio on – line. El respaldo de firmas de restaurantes reconocidos, dará estabilidad y confianza en los alquileres virtuales de locales lo cual generará a su vez una sana competencia y una cultura estándar de servicio al consumidor final.

Deseamos destacar que las tecnologías utilizadas para la elaboración de este producto aseguran el éxito y rápido posicionamiento en el mercado debido a que pertenecen a un grupo de herramientas de punta probadas en el mercado , las mismas que procuran esquematizar sistemas altamente competitivos.

Sin importar que no exista leyes definidas claramente para la comercialización por internet en Ecuador y las posibles dudas que tengan los navegantes durante los primeros días que se libere la página al Web , llevamos una ventaja sobre los tipos de negocios tradicionales a no efectuar transacciones de pagos en línea , la misma se efectivizará en el momento en que el consumidor final reciba en su localidad al despachador de pedidos.

Invitamos a los restaurantes que tengan implementados servicios de despachos a domicilio a suscribirse a alquilar un local virtual en www.patiodecomidas.com

para ser parte de esta innovadora forma de vender y acercarse a un mercado de alto poder de consumo, como lo es de la internet; por nuestra parte le aseguramos el éxito en esta línea especializada.

Recomendaciones

A los restaurantes que contraten el servicio de alquiler de un local virtual en el patio de comidas , recomendamos considerar los aspectos mencionados a continuación:

- Gran parte del beneficio que se derive del patio de comidas dependerá que los administradores de cada restaurante den la debida atención y gestionen los despachos a domicilio ya que un descuido al respecto podría repercutir no solo en la imagen del local sino de todo el patio de comidas. Por esto sugerimos a la alta gerencia de los restaurantes que asignen al menos a una persona para que se encargue de revisar los pedidos que lleguen por medio de la red de esta manera tener centralizado este servicio que bien manejado representará un rubro relevante en las ventas.
- Los restaurantes deben analizar la importancia de no sólo hacer conocer sus especialidades en la red , sino además estar siempre cambiando los menús y combos de temporadas de tal manera que genere comodidad a los cyber-compradores ; y también revisar siempre que los productos sean de fácil salida por el precio asignados a los mismos.

- Promover el servicio, buscando auspiciantes de casa reconocidas para asignar la publicidad de los mismos en cada local , lo cual generará el acrecentamiento de una imagen de servicio que repercutirá en las ganancias y posicionamiento en el mercado.

Bibliografía

- Desarrollo de Componentes COM/ActiveX con V B 6.0
Dan Appleman (Prentice Hall)
- Programación de aplicaciones distribuidas utilizando V B 6.0
Microsoft Press
- Marketing Enfoque Latinoamérica
Rolan Arellano C. (Mc Graw Gill)
- Encuestas personalizadas realizadas a los restaurantes de comida rápida, con el fin de captar su opinión acerca de la innovación de los negocios aplicando la internet y su valor en los procesos de ventas.

Además hemos consultado las siguientes páginas web

- URL,http://www.osmosislatina.com/aplicaciones/transaccion_financiera.htm
- URL,<http://www.xml.apache.org>
- URL,<http://www.corpece.net>
- URL,<http://www.verisgn.com>

APENDICE A



- ***ARQUITECTURA CLIENTE /
SERVIDOR***

1. Antecedentes.

Las computadoras personales y los paquetes de software de aplicaciones proliferan comercialmente. Estas computadoras, también conocidas como estaciones de trabajo programables, están conectadas a las Redes de Área Local (LAN), mediante las cuales, los grupos de usuarios y profesionales comparten aplicaciones y datos. Las nuevas tecnologías de distribución de funciones y datos en una red, permiten desarrollar aplicaciones distribuidas de una manera transparente, de forma que múltiples procesadores de diferentes tipos (computadoras personales de gama baja, media y alta, estaciones de trabajo, minicomputadoras o incluso mainframes), puedan ejecutar partes distintas de una aplicación. Si las funciones de la aplicación están diseñadas adecuadamente, se pueden mover de un procesador a otro sin modificaciones, y sin necesidad de retocar los programas que las invocan. Si se elige una adecuada infraestructura de sistemas distribuidos y de herramientas de desarrollo, las aplicaciones resultantes podrán trasladarse entre plataformas de distintos proveedores.

A mediados de los 90's, se decía que el desarrollo de aplicaciones Cliente / servidor era inevitable por varias razones: era más eficiente que el procesamiento centralizado, sobre todo cuando aumentaba mucho la cantidad de usuarios; ya existían en ese momento servidores razonablemente eficientes y confiables, de los cuales la mayoría había adoptado el estándar para una interfaz Cliente / servidor (el ODBC SQL). Además, se consideraba imprescindible para apoyar con información a la creciente cantidad de ejecutivos de nivel medio para la toma de decisiones.

Sin embargo, existía tecnología para esta arquitectura desde hacía ya mucho tiempo atrás, sin que nada ocurriera. Los primeros trabajos conocidos para la arquitectura Cliente / servidor los hizo Sybase, que se fundó en 1984 pensando en lanzar al mercado únicamente productos para esta arquitectura. A fines de la década de los 80's el producto fue lanzado para el voluminoso segmento "low-end" del mercado, en conjunción con Microsoft, teniendo como soporte de la base de datos un servidor OS/2, y como herramienta "front end" básica el Dbase IV de Ashton Tate. El Dbase IV no se mostró como una herramienta adecuada, y los desencuentros comerciales entre Sybase, Microsoft e IBM (en aquel momento socia de Microsoft para el OS/2) hicieron el resto.

La situación era muy diferente en 1994, cuando los principales fabricantes tradicionales (Informix, Oracle, Sybase) habían lanzado al mercado poderosos servidores y, a ellos, se agregaba IBM que estaba lanzando su producto DB2 para, prácticamente, todos los sistemas operativos importantes (además de sus clásicos MVS y VM, ahora anunciaba AIX, OS/2, Windows NT, Hewlett Packard's UNIX, Sun's UNIX, Siemens' UNIX, etc.) y Microsoft que, luego de finalizar su acuerdo con Sybase, partió para su propio SQL Server para Windows NT. Existía un conjunto de lenguajes "front end" como, por ejemplo, Delphi, Foxpro, Powerbuilder, SQL Windows, Visual Basic, etc. Se decía en aquel momento que Visual Basic, más allá de sus méritos intrínsecos como lenguaje, era el favorito para dominar el mercado, cosa que está ocurriendo. Por otra parte, en la comunidad informática existían muchas dudas sobre la calidad de los optimizadores de los sistemas de gerencia de base de datos, cuyas fallas del pasado habían sido causantes de verdaderas historias de horror.

2.- La Arquitectura Cliente / servidor.

La arquitectura Cliente / servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, inicia un proceso de diálogo: produce una demanda de información o solicita recursos. La computadora que responde a la demanda del cliente, se conoce como servidor. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. Cliente / servidor 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. Se puede decir que la arquitectura Cliente / servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información; estableciendo así un enlace de comunicación transparente entre los elementos que conforman la estructura. No existe una definición específica adoptada universalmente de la Arquitectura Cliente / servidor, las empresas de cómputo

enfocan el concepto basándose en la funcionalidad que representa según los servicios que ellas mismas ofrecen.

2.1.- Características.

Entre las principales características de la arquitectura Cliente / servidor, se pueden destacar las siguientes:

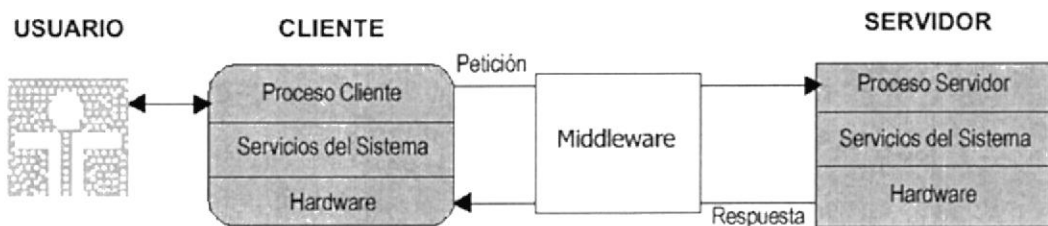
- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.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

Todos los sistemas desarrollados en arquitectura Cliente / servidor poseen las siguientes características distintivas de otras formas de software distribuido:

- **Servicio.** El servidor es un proveedor de servicios; el cliente es un consumidor de servicios.
- **Recursos compartidos.** Un servidor puede atender a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.
- **Protocolos Asimétricos.** La relación entre cliente y servidor es de muchos a uno; los clientes solicitan servicios, mientras los servidores esperan las solicitudes pasivamente.
- **Transparencia de ubicación.** El software Cliente / servidor siempre oculta a los clientes la ubicación del servidor.

- **Mezcla e igualdad.** El software es independiente del hardware o de las plataformas de software del sistema operativo; se puede tener las mismas o diferentes plataformas de cliente y servidor.
- **Intercambio basados en mensajes.** Los sistemas interactúan a través de un mecanismo de transmisión de mensajes: la entrega de solicitudes y respuestas del servicio.
- **Encapsulamiento de servicios.** Los servidores pueden ser sustituidos sin afectar a los clientes, siempre y cuando la interfaz para recibir peticiones y ofrecer servicios no cambie.
- **Facilidad de escalabilidad.** Los sistemas Cliente / servidor pueden escalarse horizontal o verticalmente. Es decir, se pueden adicionar o eliminar clientes (con apenas un ligero impacto en el desempeño del sistema); o bien, se puede cambiar a un servidor más grande o a servidores múltiples.
- **Integridad.** El código y los datos del servidor se conservan centralmente; esto implica menor costo de mantenimiento y protección de la integridad de los datos compartidos. Además, los clientes mantienen su individualidad e independencia.

La arquitectura Cliente / servidor es una infraestructura versátil modular y basada en mensajes que pretende mejorar la portabilidad, la interoperabilidad y la escalabilidad del cómputo; además es una apertura del ramo que invita a participar a una variedad de plataformas, hardware y software del sistema. Conceptualmente, los componentes de la arquitectura Cliente / servidor son el cliente, el servidor y la infraestructura de comunicaciones (middleware), como a continuación se explica.



2.2.- Cliente.

El cliente es la entidad por medio de la cual un usuario solicita un servicio, realiza una petición o demanda el uso de recursos. Este elemento se encarga, básicamente, de la presentación de los datos y/o información al usuario en un ambiente gráfico.

Se comunica 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; además, requiere el uso de los recursos de la computadora para cualquier actividad y puede interactuar con uno o varios servidores. Los clientes se suelen situar en PC's o en estaciones de trabajo se encargan de realizar el FRONT END, que es la parte de la aplicación que interactúa con el usuario, en ellos permanecen las aplicaciones particulares de cada usuario, y realizan funciones como:

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

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.

2.3.- Servidor.

El servidor es la entidad física que provee un servicio y devuelve resultados; ejecuta el procesamiento de datos, aplicaciones y manejo de la información o recursos. En el servidor se realiza el BACK END que es la parte destinada a recibir las solicitudes del cliente y dónde se ejecutan los procesos.

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 PC's 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.

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 computadoras personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

En el servidor permanecen las aplicaciones que deben ser compartidas por varios usuarios. Normalmente, aunque con excepciones, el cliente y el servidor están

ubicados en distintos procesadores; incluso, un servidor puede fungir como cliente de otros servidores.

Existen diversos servidores mismos que se clasifican basándose en su funcionalidad; estos son denominados servidores dedicados ya que administran el uso de algún recurso en particular, por ejemplo:

Servidor de Archivos.

El cliente envía solicitudes de registros de archivos al servidor; es un simple servicio de datos compartidos por medio de la red. Son útiles para almacenar archivos (documentos, imágenes, planos, etc.) y aplicaciones de productividad (procesadores de texto, hojas de cálculo).

Servidor de Bases de datos.

El cliente envía solicitudes de SQL en calidad de mensajes (un mensaje por instrucción); el servidor hace uso de su propia capacidad de procesamiento para encontrar los datos solicitados y devolverlos por medio de la red, sin enviar todos los registros. Este tipo de servidores permiten hacer consultas específicas y obtener reportes flexibles.

Servidor de Transacciones.

El cliente activa procedimientos remotos que residen en el servidor con un mecanismo de bases de datos de SQL. Es decir, el intercambio por la red consiste en un solo mensaje de solicitud / respuesta que ejecuta un grupo de instrucciones SQL (llamadas transacciones) en el servidor; al crear la aplicación Cliente / servidor, se genera código tanto para el cliente como para el servidor. A estas aplicaciones se

les llama procesamiento de transacciones en línea (OLTP: online transaction processing), las cuales requieren de firmes controles sobre la seguridad e integridad de la base de datos.

Servidor de Groupware.

Este tipo de servidor es utilizado para el seguimiento de operaciones dentro de la red. El groupware dirige la administración de información semiestructurada como texto, imagen, correo, tableros de avisos y flujo de trabajo; estableciendo un contacto directo entre personas. El software de groupware, generalmente, son aplicaciones creadas en un lenguaje de scripts e interfaces basadas en formas. Como ejemplo está Lotus Notes.

Servidor de Objetos.

La aplicación Cliente/Servidor se genera como un conjunto de objetos de comunicación; los objetos del cliente se comunican con los objetos del servidor mediante un corredor de solicitudes de objetos (ORB: object request broker). El cliente invoca un método de un objeto remoto. El ORB localiza una instancia de esa clase de servidor de objetos, invoca el método solicitado y envía los resultados al objeto cliente. Los objetos del servidor deben ofrecer soporte de concurrencia y participación; el ORB se encarga de reunir todos los elementos.

Servidor de Web.

Se usan como una forma inteligente para comunicación entre empresas a través de Internet. Este servidor permite transacciones con el acondicionamiento de un browser específico. Este modelo está integrado por clientes compactos y portátiles en comunicación con servidores amplios. Tal comunicación se da mediante un

protocolo denominado HTTP, el cual define un conjunto simple de órdenes, los parámetros se transmiten en cadenas sin estipulaciones de datos tecleados. En el proceso de ampliación de Internet, ya existen combinaciones con objetos distribuidos (Java) para ofrecer modalidades más interactivas.

Servidor de Impresoras.

Es el dispositivo encargado de administrar las colas de impresión direccionando los trabajos a los diferentes dispositivos.

Servidor de Aplicaciones.

En este medio se almacenan y ejecutan las aplicaciones de software utilizadas por los usuarios, evitando así la duplicidad de las mismas, permiten un mejor control para la actualización de versiones y productos.

Servidor de Respaldos.

Este es una innovación propia del ambiente Cliente / servidor; de gran utilidad ya que administra la ejecución de los respaldos en línea, asegurando que estos se realicen aún en caso de error, ya que si un dispositivo falla, automáticamente direccionará el proceso a otro para que se ejecute el respaldo correspondiente.

El proceso del servidor es reactivo, es decir, realiza una acción basándose en una instrucción previa; simplificando, realiza una función posterior a una petición o a la ejecución de una transacción requerida por el cliente, o bien por otro servidor.

2.4.- Middleware.

Para que los clientes y servidores puedan comunicarse se requiere de una infraestructura lógica que proporcione los mecanismos básicos de direccionamiento y transporte. A dicha infraestructura se le denomina Middleware, el cual es un término que abarca a todo el software distribuido necesario para el soporte de interacciones entre clientes y servidores.

El middleware es un módulo intermedio que no pertenece a los dominios del servidor, ni a la interfaz de usuario, ni a la lógica de la aplicación en los dominios del cliente; tampoco debe confundirse con la red física en sí (cableado, señales de radio o infrarrojas), el middleware es una interfaz lógica estándar de los servicios de red.

Sus funciones son:

- Independizar las dos entidades: El cliente y el servidor no necesitan saber comunicarse entre ellos, sino cómo comunicarse con el módulo de middleware.
- Traducir la información de una aplicación y pasarla a la otra: acepta consultas y datos recuperándolos de la aplicación cliente, los transmite y envía la respuesta de regreso. También genera los códigos de error.
- Controlar las comunicaciones: da a la red las características adecuadas de desempeño, confiabilidad, transparencia y administración.

Existen dos tipos de middleware:

1. El **Middleware general** es el sustrato de la mayoría de las interacciones de Cliente / servidor. Incluye las pilas de comunicación, directorios distribuidos, servicios de autenticación, horario de la red, llamadas a procedimientos remotos (RPC's), y servicios en cola. Incluye también las extensiones del sistema operativo de redes, como los servicios distribuidos de archivos e impresión y los

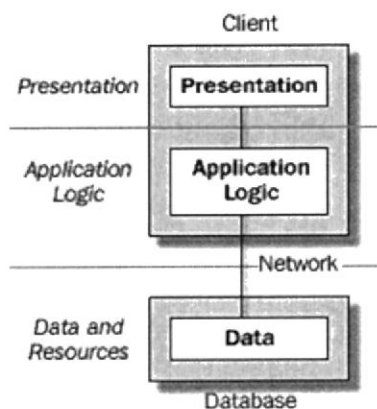
productos de middleware orientado a mensajes (MOM: message-oriented middleware).

2. El **Middleware de servicios** específicos es necesario para cumplir un tipo particular de servicio de Cliente / servidor; así, existe un middleware específico para los servidores dedicados: Middleware para bases de datos, middleware para OLTP, middleware para groupware; middleware para objetos, etc. El middleware es una herramienta adecuada, que no sólo es flexible y segura, sino que también protege la inversión en tecnología y permite manejar diferentes ambientes de computación.

3.- Modelos de la Arquitectura Cliente / servidor

Dado que la percepción sobre la Arquitectura Cliente / servidor es algo abstracto, algunos autores la definen en base a modelos de dos y tres niveles de acuerdo al número de estratos respectivos para representar a sus componentes:

Modelos de Dos Niveles (Two Thier).



Se considera como modelo Cliente / servidor de dos niveles o capas a la estructura más simple, cuyos componentes son:

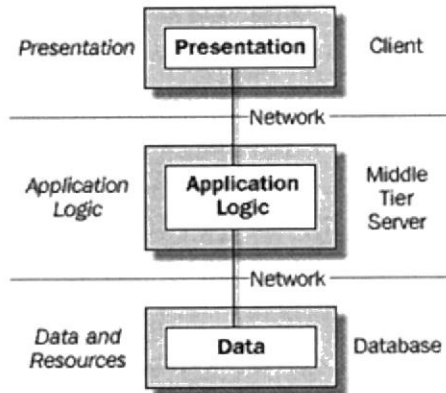
Clientes: Por medio de la interfaz con el usuario vía una petición se solicita un servicio, el uso de un recurso, o bien el acceso a un conjunto de datos.

Servidores: Satisfacen la solicitud del usuario recibiendo la petición, direccionándola y enviando la respuesta al cliente, ya sea la consulta respectiva de datos, ejecutando el proceso requerido o permitiendo el acceso y/o uso del recurso.

En este modelo se acostumbra instalar las bases de datos dentro del servidor, por las ventajas de almacenamiento y velocidad que ofrece en comparación con las del cliente.

Modelo de Tres Niveles (Three Tier).

El objetivo de este modelo es dividir las funciones de una aplicación en tres componentes:



Presentación: Este componente se encarga de la interacción hombre máquina a través del monitor, teclado, ratón, o bien mediante algún otro medio como reconocedor de voz.

Servidores: Compuesto por varios servidores o componentes de Software localizados en una o más plataformas que se encargará de conectar los sistemas existentes.

Información: En este componente se incluye la información en sí, los sistemas y aplicaciones existentes.

APENDICE B



- ***PROGRAMACION EN EL WEB***
- ***TECNOLOGIAS EXISTENTES***

Introducción

Cuando el World Wide Web inició su funcionamiento como lo conocemos, empezando a tomar popularidad aproximadamente en 1993, solo se podía apreciar texto, imágenes y enlaces.

Un documento HTML es estático, lo que significa que existe en un estado constante; es un archivo de texto que no cambia. La introducción de Plugins en los navegadores permitió mayor interactividad entre el usuario y el cliente, aunque estaba limitado por la velocidad y la necesidad de tener que bajar e instalar cada plugin que se necesitara, por lo que estos se desarrollaron mayormente en áreas de video, audio y realidad virtual.

Viendo la necesidad de tener un mayor interacción entre el usuario y los sitios web, se fueron presentando diferentes tecnologías, con características similares para el paso de parámetros y acceso a bases de datos, las cuales dieron un giro a internet; dotando de "inteligencia" e interactividad a un sitio web, esto fue el punto de partida para el comercio electrónico.

A continuación veremos algunas de las herramientas más usadas para el desarrollo de sitios Web.

1.- Cold fusion (Allaire)

ColdFusion es un lenguaje de servidor que fue desarrollado por Allaire (ahora parte de Macromedia, Inc). Los dos componentes fundamentales: el lenguaje de programación ColdFusion Markup Language (CFML) y el servidor ColdFusion. son

el centro de la tecnología ColdFusion. CFML funciona con "tags" (etiquetas) fáciles de aprender que facilitan al usuario el trabajo con objetos funcionales de gran estabilidad. Los "tags", parecidos a HTML y XML, los procesa el servidor de ColdFusion, sin que sea necesario que el código de fuente se encuentre en el ordenador del cliente.

Es una potente herramienta para realizar las funciones de acceso a la información alojada en bases de datos.

Mediante el uso de esta herramienta, se puede distribuir información a nivel de Internet y/o de intranets, ya que permite conectar una base de datos al interior de una red (Intranet) o a nivel de redes más amplias en el Web (Internet).

La conexión con la base de datos es realizada haciendo uso de programación, generando posteriormente las páginas Web de manera dinámica, cuyo contenido será la información que está alojada en la base de datos. Así mismo, permite introducir nueva información dentro de una base de datos, tener acceso a datos actualizados periódicamente, automatizando toda la actividad relacionada con dicha base de datos.

Cold Fusion centra su potencialidad en la confiabilidad y el control del manejo de datos. Reconoce la complejidad del manejo e interacción de escritos CGI, ofreciendo una potente seguridad, veloz carga de datos, procesamiento rápido de escritos CGI que posibilita el cumplimiento de tareas de entrada o devolución de datos.

Entre las funciones de Cold Fusion están:

- Sirve a cualquier requisición de datos una vez que cuente con la instalación y configuración de las fuentes de datos ODBC de 32-bits.
- Detecta errores producidos por la mala configuración o por el registro completo de la bitácora del servidor SQL.
- Funciona correctamente en una máquina remota. Se ejecuta sin problemas en el Microsoft Internet Information Server, aún teniendo gran cantidad de solicitudes. Gracias a ello brinda un correcto funcionamiento tanto en Internet como en Intranets.
- Provee de ayuda para la configuración que permita generar páginas HTML en forma dinámica.
- Crea estructuras condicionales dinámicamente para personalizar la solicitud de datos y el envío de los mismos hacia el cliente. Así mismo, diseña cadenas de datos para crear dinámicamente menús desplegados y para llenar listas de selección y listas de documentos.

¿ Cómo funciona Cold Fusion ?

Se pueden realizar requisiciones a través de un URL, las cuales son enviadas al servidor Web, y éste a su vez la hace a la interfaz de Cold Fusion, la que se conecta a una fuente de datos ODBC, a la cual solicita los datos que requiere extraer de la base de datos. Una vez se ha obtenido la información que se ha solicitado, la interfaz envía los datos hacia el Servidor Web y éste al browser, en donde los mismos son desplegados gráficamente.

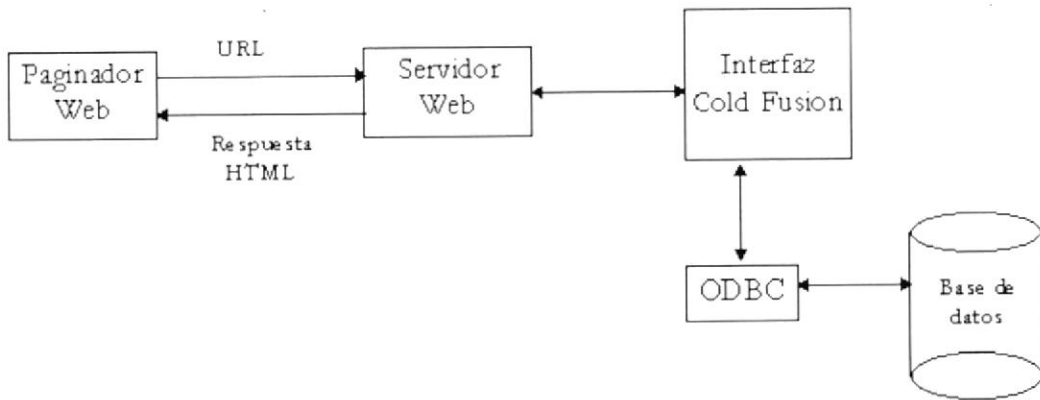


Diagrama B.1 Arquitectura Cold Fusion para acceder bases de datos en el Web.

1.1.- Ventajas y desventajas

- Puede servir cualquier requisición de datos mediante el uso de fuentes de datos ODBC de 32-bits y de sentencias SQL.
- Detecta fácilmente los errores.
- Intuitivo script basado en etiquetas
- Integración con otros lenguajes web, como HTML, XML, WML, Flash u otros
- Librería de funciones
- Integración con protocolos de Internet
- Independencia de clientes
- Etiquetas personalizadas y funciones definidas por el usuario
- Funciona correctamente tanto en Internet como en Intranets.

- Capacidades poderosas de manejo de datos, es muy confiable y ofrece control.
- Integración con bases de datos, e-mail, directorios, Java, XML, y sistemas empresariales permitiendo desarrollar aplicaciones Web complicadas de forma rápida y fácil.
- Ofrece una potente seguridad, veloz carga de datos, procesamiento rápido de escritos CGI para la entrada o devolución de datos.
- Extensibilidad a través de C, C++, Java
- Conectividad con COM, CORBA y EJB

2.- CGI (Por sus siglas en inglés "Common Gateway Interface").

El CGI es un estándar para comunicar aplicaciones externas con los servidores

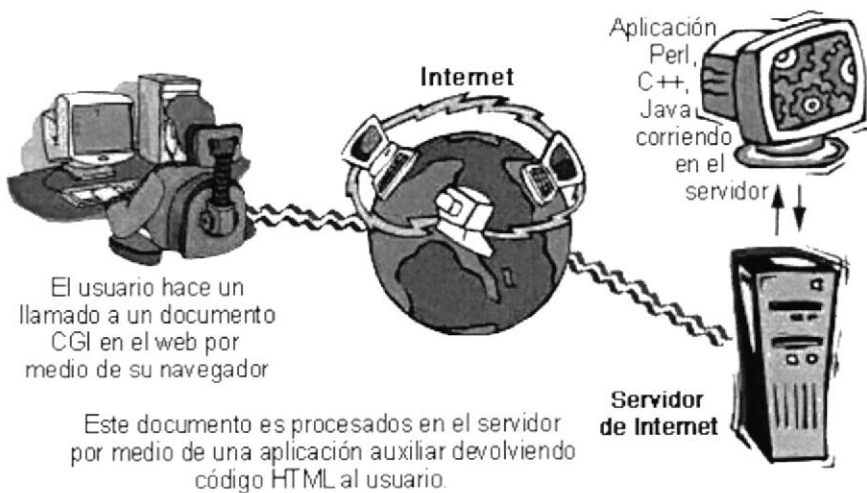


Diagrama B.2 Arquitectura de CGI en la ejecución de CGI's

2.1.- Ventajas y desventajas

- Probablemente lo que afectará al usuario típico del Web, es que hecho de que los programas CGI necesitan residir en un directorio especial, así el servidor sabe que tiene que ejecutarlo.
- Como los CGI son ejecutados en el servidor, permite el acceso a bases de datos.
- Permite ejecutar en el servidor programas previamente compilados.
- Para usar alguno de los lenguajes interpretados, como PERL, TCL, o un shell de Unix, el script necesita residir en el directorio /cgi-bin.
- Como un programa CGI es un ejecutable, es equivalente a dejar a el mundo ejecutar un programa en tu sistema, que no es lo mas seguro a hacer. Por ello existen una serie de precauciones de seguridad que son necesarias de implementar cuando se usan programas CGI.
- Hay servidores cuya configuración particular obliga a que la extensión de los cgis sea .cgi, esto puede variar las extensiones, lenguajes y directorios para ejecutar CGI's.

3.- PHP (Hypertext preprocessor)

Veamos algo de su historia, PHP fue concebido en otoño de 1994 por Rasmus Lerdorf. Las primeras versiones no distribuidas al público fueron usadas en un sus páginas web para mantener un control sobre quien consultaba su

currículum. La primera versión disponible para el público a principios de 1995 fue conocida como "Herramientas para páginas web personales" (Personal Home Page Tools). Consistían en un analizador sintáctico muy simple que solo entendía unas cuantas macros y una serie de utilidades comunes en las páginas web de entonces, un libro de visitas, un contador y otras pequeñas cosas. El analizador sintáctico fue reescrito a mediados de 1995 y fue nombrado PHP/FI versión 2. FI viene de otro programa que Rasmus había escrito y que procesaba los datos de formularios. Así que combinó las "Herramientas para páginas web personales", el "intérprete de formularios", añadió soporte para mSQL y PHP/FI vio la luz. PHP/FI creció a gran velocidad y la gente empezó a contribuir en el código.

A mediados de 1997 el desarrollo del proyecto sufrió un profundo cambio, dejó de ser un proyecto personal de Rasmus, al cual habían ayudado un grupo de usuarios y se convirtió en un proyecto de grupo mucho más organizado. El analizador sintáctico se reescribió desde el principio por Zeev Suraski y Andi Gutmans y este nuevo analizador estableció las bases para PHP versión 3. Gran cantidad de código de PHP/FI fue portado a PHP3 y otra gran cantidad fue escrito completamente de nuevo.

A finales 1999, tanto PHP/FI como PHP3 se distribuyen en un gran número de productos comerciales tales como el servidor web "C2's StrongHold" y Redhat Linux. Una estimación conservativa basada en estadísticas de NetCraft (ver también Estudio de NetCraft sobre servidores web), es que más de 1.000.000 de servidores alrededor del mundo usan PHP. Para hacernos una idea, este número es mayor que el número de servidores que utilizan el "Netscape's Enterprise server" en Internet.

El crecimiento de PHP desde entonces ha sido exponencial, y han surgido versiones nuevas.

Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones.... No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.



Diagrama B.3. Arquitectura de PHP

En el nivel más básico PHP es equiparable a un CGI cualquiera. La mayor fuerza de PHP es que está preparado para soportar accesos a muchos tipos de bases de datos.

Además de esto, PHP soporta la utilización de otros protocolos como IMAP, SNMP, NNTP, POP3 o HTTP a nivel de socket.

3.1.- Ventajas y desventajas

- PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos.
- Para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.
- En el nivel más básico PHP es equiparable a un CGI cualquiera.
- PHP soporta la utilización de otros protocolos como IMAP, SNMP, NNTP, POP3 o HTTP a nivel de socket.

4.- XML

Antes de nada conviene repasar su historia y precedentes. La versión 1.0 del lenguaje XML es una recomendación del W3C desde Febrero de 1998, pero se ha trabajado en ella desde un par de años antes. Está basado en el anterior estándar SGML (Standard Generalized Markup Language, ISO 8879), que data de 1986, pero que empezó a gestarse desde principios de los años 70, y a su vez basado en el GML creado por IBM en 1969. Esto significa que aunque XML pueda parecer moderno, sus conceptos están más que asentados y aceptados de forma amplia. Está además asociado a la recomendación del W3C DOM (Document Object Model), aprobado también en 1998. Éste no es más que un modelo de objetos (en forma de API) que permite acceder a las diferentes partes que pueden componer un documento XML o HTML.

Como HTML (Hypertext Markup Language) es una lengua de marcas (markup), es decir, una lengua que presenta información encuadrada por marcas; que presenta un juego limitado de etiquetas orientadas presentación (título, apartado, imagen, vínculo hipertexto, etc),

XML es un metalenguaje, que va a permitir inventar a discreción nuevas marcas para aislar todas las informaciones elementales (título de obra, precio de artículo, número de seguridad social, referencia de parte...), o agregados de informaciones elementales, que puede contener una página Internet.

El XML es un subconjunto de SGML y tanto XML como HTML crecieron al amparo de este.

SGML (Standard Generalized Markup Language) es un meta-lenguaje, es decir, un lenguaje que ofrece la posibilidad de definir otros lenguajes. Sin embargo, mientras que HTML es una instancia concreta de SGML, un conjunto de etiquetas definidas especialmente para visualizar páginas web

XML es otro metalenguaje, como SGML, pero con la particularidad de que ha surgido por la necesidad de disponer de algo más simple que SGML a la hora de describir documentos.

Además, aunque SGML ha sido utilizado durante más de una década para mantener repositorios de información estructurada, no es del todo apropiado para su distribución por la red por multitud de razones, la principal, es su dificultad para ser soportado por los navegadores.

La gran tecnología que capacita a Internet tal y como la conocemos hoy es HTML (lenguaje utilizado para definir la presentación de las páginas que encontramos en la Web). Gracias a HTML (Hypertext Markup Language) hemos podido combinar texto y gráficos en una página y crear sistemas complejos con hiperenlaces entre páginas. Pero HTML no es útil en lo que se refiere a la descripción de la información. Por ejemplo, se puede usar HTML para dar formato a una tabla, pero no para describir los elementos de datos que componen la tabla. Aquí es donde XML promete ser más útil.

La atención que ha generado XML es debido al lugar donde se facilita el intercambio de información, ya que es posible utilizar este lenguaje tanto para generar como distribuir información que sea utilizada por bases de datos ,aplicaciones de servidor ,aparatos inalámbricos, impresoras, etc. La principal ventaja que presenta este lenguaje es su independencia del sistema operativo y de la aplicación que será capaz de utilizarlo, esto es, se puede tener un documento escrito en XML y este puede ser manipulado en los sistemas operativos: Sun Solaris, Windows, AIX, Linux o en un ambiente Java,VBScript,PL/SQL.

El proceso que lleva a cabo el documento es el siguiente:

- 1 Definir el documento y su DTD (Definición de TAGS).
- 2 Se procesa ("parse") el documento vía "DOM" (Document Object Model) o "SAX" (Simple API for XML).
- 3 Y se aplica el "XSL" (Extensible Stylesheet Language) correspondiente

Estos tres pasos son los que garantizan que el mismo documento escrito en XML pueda ser llevado de un punto a otro sin ningún conflicto. Las posibilidades que pueden surgir a partir de esto son interminables.

Hoy en día ya han sido definidos varios vocabularios que definen este tipo de TAGS en base a industrias, de manera que de la misma forma en que ya existe un estándar para TAGS de presentación (HTML), varias industrias han empezado a definir sus propios TAGS : Industria Química QML "Chemical Markup Language" , Industria Legal XFDL "Extensible Forms Description Language" y otra gran gamma de Industrias, una lista se encuentra en: <http://www.oasis-open.org/cover/>

En dado caso de no existir algún TAG en el vocabulario o simplemente no existir el vocabulario como tal, es posible definir sus propios TAGS mediante un DTD "Data Type Definiton". Para poder intercambiar cualquier documento XML es necesario que el receptor conozca la definición de estos DTD.

¿Y qué es un DTD? Pues es una definición de los elementos que puede haber en el documento XML, y su relación entre ellos, sus atributos, posibles valores, etc. De hecho DTD está por Document Type Definition, o Definición de Tipo de Documento. Es una especie de definición de la gramática del documento.

4.1 Ventajas y desventajas

- XML es un metalenguaje, que permite inventar a discreción nuevas marcas.
- Es posible utilizar este lenguaje tanto para generar como distribuir información que sea utilizada por bases de datos ,aplicaciones de servidor, aparatos inalámbricos, impresoras, etc.
- Lo que presenta este lenguaje es su independencia de sistema operativo y de la aplicación que será capaz de utilizarlo, esto es, se puede tener un documento escrito en XML y este puede ser manipulado en varios sistemas operativos.

- Los documentos XML son sensibles a mayúsculas, esto es, en ellos se diferencia las mayúsculas de las minúsculas.
- Debido a que existe un SOLO documento que define los datos es posible concentrar los esfuerzos de administración y programación en este solo documento maestro (XML).
- Soporta lenguajes como javascripts, Java,VBScript,PL/SQL.

4.- Java Server Pages

Java Server Pages TM (JSP) es un conjunto de tecnologías que permiten la generación dinámica de páginas web combinando código Java (scriptlets) con un lenguaje de marcas como HTML ó XML, para generar el contenido de la página.

Como parte de la familia de la tecnología Java, con JSP podemos desarrollar aplicaciones web independientes de la plataforma. Una característica importante es que permite separar la interfaz del usuario de la generación del contenido dinámico, dando lugar a procesos de desarrollo más rápidos y eficientes.

Adicionalmente, pueden acceder directamente a componentes Java Beans ó Enterprise Java Beans (EJB), instanciandolos y estableciendo sus propiedades e invocando sus métodos directamente desde la página JSP. Esto permite desarrollar aplicaciones n-capas donde se separan en lo posible los datos, la lógica del negocio y la lógica de presentación, encapsulando, generalmente, en Beans el acceso a los datos.

La tecnología JSP es una extensión de la tecnología Servlets, los cuales son aplicaciones 100% Java que corren en el servidor: Un Servlet es creado e inicializado, se procesan las peticiones recibidas y por ultimo se destruye. Este diseño explica por que un Servlet reemplaza perfectamente a un CGI, ya que el servlet es cargado una sola vez y esta residente en memoria mientras se procesan las peticiones recibidas y se generan las respuestas a los usuarios.

Cada vez que un cliente solicita al servidor web una página JSP, este pasa la petición al motor de JSP el cual verifica si la página no se ha ejecutado antes ó fue modificada después de la última compilación, tras lo cual la compila, convirtiéndola en Servlet, la ejecuta y devuelve los resultados al cliente en formato HTML.

La especificación JSP es el producto de una amplia colaboración de varias de las industrias líderes en el desarrollo de software, liderados por Sun Microsystems.

JSP tiene gran variedad de formas para comunicarse con las clases de Java, servlets, applets y el servidor web; por esto se puede aplicar una funcionalidad a nuestra web a base de componentes.

Es posible ver el código del servlet generado, este código debe estar en el directorio que se informa en la estructura de directorios del servidor.

Servlets

La herramienta más importante que se usa a la hora de desarrollar web con Jsp son los Servlets; los servlets son la primera línea de batalla del desarrollo de las aplicaciones web. Estos aportan una manera fácil para que nuestro servidor se comunique con el lado cliente.

Los servlets dan un modelo general de clases para ejecutar servicios. Al más básico nivel, esta es la definición de un servidor.

Son programas que se ejecutan en un servidor Web y construyen páginas Web. Construir páginas Web al vuelo es útil (y comúnmente usado) por un número de razones:

La página Web está basada en datos enviados por el usuario. Por ejemplo, las páginas de resultados de los motores de búsqueda se generan de esta forma, y los programas que procesan pedidos desde sites de comercio electrónico también.

Los datos cambian frecuentemente. Por ejemplo, un informe sobre el tiempo o páginas de cabeceras de noticias podrían construir la página dinámicamente, quizás devolviendo una página previamente construida y luego actualizándola.

Las páginas Web que usan información desde bases de datos corporativas u otras fuentes. Por ejemplo, usaríamos esto para hacer una página Web en una tienda online que liste los precios actuales y el número de artículos en stock.

Contenedor de Servlet

El contenedor de servlets tiene que encargarse, entre otras cosas, de pasar las peticiones del cliente al servlet y este último de devolver la respuesta solicitada al cliente. La implementación actual del servlet difiere de un programa a otro, pero la interface entre contenedor del servlet y el servlet se especifica en la API del servlet.

5.1.- Ventajas y desventajas

- Independencia de la plataforma
- Rendimiento mejorado
- Separación de la lógica de la aplicación de la presentación de los datos
- Uso de componentes (Java Beans)
- Facilidad de administración y uso
- El importante respaldo de la sólida tecnología Java TM.
- Permite acceso a bases de datos
- Se tiene que ubicar en un lugar determinado los servlets, para poder ejecutarlos
- Los servlets permiten ejecutar escritos de otros archivos

Análisis Comparativos de las tecnologías anteriormente señaladas

Característica	Cold Fusion	CGI	PHP	XML	JSP	ASP
Ejecuta los scripts en el servidor	Si	Si	Si	Si	Si	Si
Permite definir nuevas etiquetas	No	No	No	Si	Si	No
Solo muestra HTML en el browser, no muestran los script.	Si	No	Si	Si	Si	Si
Puede llamar a determinados archivos para que sean ejecutados en el servidor.	Si	Si	Si	Si	Si	Si
Facilidad para crear plantillas en una pagina.	Si	No	No	No	Si	Si
Necesita servicios de otros productos para funcionar en otras plataformas se servidores.	Si	Si	Si	No	No	Si
Extensas herramientas de soporte	+/-	Si	Si	+/-	Si	Si
Habilitada para interactuar con Base de Datos	Si	Si	Si	Si	Si	Si

APENDICE C

- ***COM / DCOM / COM+***

Introducción

COM permite a los desarrolladores de software construir aplicaciones a partir de componentes de software binario que pueden ser implementados en cualquier nivel del modelo de la aplicación. Estos componentes proporcionan soporte para empaquetado, partición y funcionalidad distribuida de aplicación. COM permite que las aplicaciones sean desarrolladas con componentes que encapsulan cualquier tipo de código o funcionalidad de aplicación, tal como un control de interfase de usuario o una línea de objeto de negocio. Un componente puede tener una o mas interfases; cada una expone un conjunto de métodos y propiedades que pueden ser consultadas y establecidas por otros componentes o aplicaciones. Por ejemplo un componente cliente puede exponer varias propiedades tales como nombre, dirección y teléfono.

Con el modelo Microsoft Windows DNA (Aplicaciones de Red Distribuidas), los componentes eliminan la complejidad de construir aplicaciones multinivel. Las aplicaciones basadas en componentes y el modelo Windows DNA se basan en un conjunto común de servicios de infraestructura y de red proporcionados por la plataforma de aplicación Windows. El servicios de seguridad de Microsoft Windows NT®, por ejemplo, proporciona control de acceso al Servidor de Información Internet, así como servicios de transacciones y mensajes cola de mensajes. Otros servicios comunes incluyen administración de sistemas, servicios de directorio, red y soporte de hardware.

1.- COM

El término COM significa "Component Object Model" y es una especificación de código binario reusable que puede correr en un sistema basado en componentes.

COM está basado en la programación orientada a objetos. El cliente COM y la clase COM viven en archivos binarios separados como muestra la figura 1.

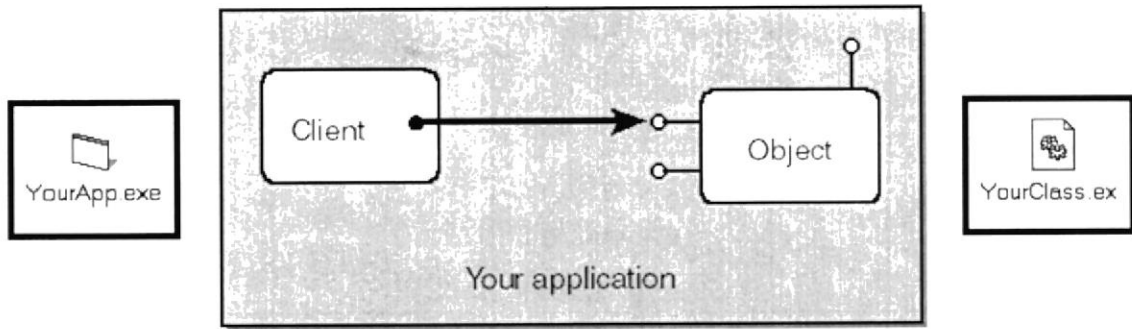


Figura C.1

COM es un modelo basado en la reusabilidad binaria. Esto significa que los componentes pueden ser usados sin ninguna dependencia con el código de origen.

La reusabilidad del código en una forma binaria elimina los problemas de compilación que ocurren cuando los sistemas son ensamblados de acuerdo a un estilo de desarrollo basado en reuso de código. El reuso binario hace más fácil incorporar pequeños cambios en un sistema.

La DLL puede ser recompilada y reemplazada sin necesidad de afectar todas las aplicaciones del cliente.

Los sistemas basados en código de origen usualmente deben ser recompilados en todas sus líneas, esto hace que el mantenimiento y la expansión del software sea más costosa.

COM permite a los desarrolladores de software construir aplicaciones a partir de componentes de software binario que pueden ser implementados en cualquier nivel del modelo de la aplicación. Estos componentes proporcionan soporte para empaquetado, partición y funcionalidad distribuida de aplicación. COM permite que las aplicaciones sean desarrolladas con componentes que encapsulan cualquier tipo de código o funcionalidad de aplicación, tal como un control de interfase de usuario o una línea de objeto de negocio. Un componente puede tener una o más interfases; cada una expone un conjunto de métodos y propiedades que pueden ser consultadas y establecidas por otros componentes o aplicaciones. Por ejemplo un componente cliente puede exponer varias propiedades tales como nombre, dirección y teléfono.

Dentro de las características de COM podemos mencionar:

- Un objeto COM puede contener datos
- Posee uno o más conjuntos de funciones llamadas interfaces, a las que se puede llamar para actuar en los datos o llevar a cabo otras operaciones, como la muestra y la persistencia de datos.
- Las funciones del objeto están contenidas en una aplicación o biblioteca de vínculos dinámicos. Normalmente son las únicas funciones que pueden acceder directamente a los datos del objeto.
- Los objetos COM pueden identificarse por GUID, que identifica un determinado tipo de objeto en el tiempo y en el espacio.
- El GUID lo puede usar el sistema para asociar una determinada instancia de un objeto con la aplicación , o .DLL, que contiene las funciones de ese objeto.

2.- DCOM

Desde el principio, COM fue diseñado a extender los procesos. Las primeras versiones de COM hacían esto posible, sólo cuando el proceso cliente y el proceso servidor se estén ejecutado en una sola computadora. Con la aparición de Microsoft Windows NT 4, COM permitía la intercomunicación de procesos para extenderlos a otras computadoras.

Este nuevo protocolo hizo posible el desarrollo de objetos distribuidos en una ambiente de red como lo indica la figura C.1. Esto fue un punto significativo para Microsoft en su estrategia de computación empresarial.

Primero, Microsoft buscaba que se identificara el término COM con la finalidad de ser usado en un sistema distribuido. DCOM hizo su primera aparición con Windows NT 4 en Agosto de 1995 .COM distribuido (DCOM) es el término usado cuando se habla de las facilidades remotas de COM, sin embargo algunos

desarrolladores de COM piensan en COM como una tecnología distribuida, y por lo tanto, la palabra "distribuido" delante de COM, está de mas.

Las capacidades de DCOM están basadas en el mecanismo de inter procesos conocido como *Llamada a Procedimiento Remoto (RPC por sus siglas en inglés)*. RPC es una tecnología estándar que ha sido madurada en diferentes plataformas. La implementación de Microsoft es conocida como MS-RPC. Sin embargo COM y RPC tienen una relación simbiótica. COM ofrece a RPC un campo orientado a objetos y RPC brinda a COM la habilidad de ser un objeto distribuido.

DCOM introduce nuevos requerimientos que no pueden ser usados cuando el cliente y el proceso corren en la misma máquina. Primero, DCOM ofrece a infraestructura para la creación y activación de un objeto remoto. Un cliente tiene la posibilidad de crear y usar objetos que corran en computadoras remotas-. Además COM y Windows NT, proveen una capa de seguridad de tal manera que todo proceso remoto sea autenticado y autorizado.

3.- COM +

A principios de los 90s, el concepto subyacente que facilitaba la interoperabilidad era la componentización; la tecnología subyacente que permitía la interoperabilidad fue COM. Resultó que la componentización no solo es una gran forma de lograr la interoperabilidad, sino una gran forma de diseñar y desarrollar software en general. Así, a mediados de los 90s Microsoft amplió la aplicabilidad de COM más allá de las aplicaciones de escritorio para incluir también aplicaciones distribuidas al introducir el Microsoft Transaction Server (MTS). MTS fue una extensión al modelo de programación COM que proporcionó servicios para el desarrollo, implementación y administración de aplicaciones distribuidas basadas en componente. MTS fue la base de los servicios de plataforma de aplicación que facilitaron el desarrollo de aplicaciones distribuidas para la plataforma Windows de una manera mucho más sencilla y barata que otras alternativas.

COM+ es el siguiente paso evolutivo de COM y MTS. La unificación de los modelos de programación inherentes en los servicios COM y MTS hace más fácil el desarrollo de aplicaciones distribuidas al eliminar las molestas y tediosas tareas asociadas con el desarrollo, depuración, implementación y administración de una aplicación que depende de COM para ciertos servicios y de MTS para otros. Los beneficios para el desarrollador de aplicaciones es hacer más rápido, fácil y, en última instancia más barato el desarrollo de aplicaciones distribuidas reduciendo la cantidad de código requerido para aprovechar los servicios subyacentes del sistema.

Para continuar ampliando COM y los servicios ofrecidos actualmente por el MTS 2.0, COM+ consiste de mejoras a los servicios existentes así como nuevos servicios para la plataforma de aplicación. Estos incluyen:

- **Traiga su propia transacción.** Los componentes COM son capaces de participar en transacciones administradas por ambientes de procesamiento de transacciones que no son de COM+ que den soporte al Protocolo de Transacciones de Internet (*Transaction Internet Protocol*, TIP).
- **Seguridad expandida.** Soporte para seguridad ya sea basada en roles o en permisos de acceso de proceso. En el modelo de seguridad basado en roles, el acceso a varias partes de una aplicación es otorgado o negado basado en el grupo o rol lógico que se ha asignado al que hace la petición (por ejemplo, administrador, empleado de tiempo completo o empleado de tiempo parcial). COM+ expande la actual implementación de la seguridad basada en rol al incluir seguridad a nivel de método para interfases basadas en *IDispatch(Ex)* y las hechas a la medida.
- **Administración centralizada.** El Explorador de Servicios de Componente (*Component Services Explorer*), un reemplazo para el actual Explorador MTS y DCOMCNFG, presenta un modelo administrativo unificado haciendo más fácil instalar, administrar y monitorear aplicaciones de n niveles al eliminar el problema de utilizar numerosas herramientas de administración individuales.

- **Base de datos en memoria.** La base de datos en memoria mantiene la información de estado, duradera y transitoria, de una manera consistente. Es un sistema de base de datos completamente transaccional, en memoria, diseñada para proporcionar acceso extremadamente rápido a datos que estén en la máquina en la cual reside.
- **Cola de componentes.** Para ejecución asíncrona diferida cuando los componentes cooperativos están desconectados, esto es adicional al modelo de programación cliente / servidor sincrónico, basado en sesiones, donde los clientes mantienen una conexión lógica hacia el servidor.
- **Notificación de eventos.** Para las ocasiones en que es deseable un mecanismo de notificación poco acoplado, los Eventos COM+ son un mecanismo de eventos unicast/multicast, de publicar / subscribir, que permite a múltiples clientes "subscribirse" a eventos que son "publicados" por varios servidores. Esto es adicional al marco de trabajo de notificación de eventos existente que se proporciona con los puntos de conexión.
- **Balance de carga.** El balance de carga permite a las aplicaciones basadas en componentes distribuir su carga de trabajo entre una agrupación (*cluster*) de servidores de aplicaciones de una manera transparente al cliente.

APENDICE D



- ***INTERNET INFORMATION SERVER (IIS)***
- ***ACTIVE SERVE PAGE (ASP)***

Internet Information Server

Integrado con la plataforma de aplicación de Microsoft está una compuerta de alto desempeño hacia el nivel de presentación. El Internet Information Server de Microsoft permite el desarrollo de aplicaciones de negocios basadas en Web que pueden ser extendidas sobre Internet o implementadas en intranets corporativas. Con el IIS, Microsoft introduce un nuevo paradigma en las aplicaciones transaccionales de Internet. Las transacciones son la plomería que hace posible el ejecutar aplicaciones de negocio reales con desarrollo rápido, fácil escalabilidad y confiabilidad.

Las páginas de servidor activas (*Active Server Pages*, ASP), un componente del IIS, son el ambiente de scripts del lado del servidor, sin compilaciones independientes del lenguaje que se utiliza para crear y ejecutar aplicaciones de servidor Web interactivas. Combinando DHTML, scripts, y componentes , las ASP permiten a los desarrolladores de aplicación crear contenido Web interactivo, dinámico y poderosas aplicaciones basadas en Web.

Con la tendencia hacia el cómputo distribuido en ambientes empresariales, es importante tener comunicación flexible y confiable entre aplicaciones. Los negocios regularmente requieren que aplicaciones independientes que se ejecutan en sistemas diferentes se comuniquen entre ellas mismas, e intercambien mensajes aún y cuando las aplicaciones puedan no estar ejecutándose al mismo tiempo. Las aplicaciones construidas utilizando una combinación de scripts ASP comunicándose con componentes cooperativos puede inter operar con sistemas, aplicaciones y datos existentes.

Active Server Page (ASP)

1. Introducción.

El ASP fue anunciado oficialmente al mundo por Microsoft el 16 de julio de 1996, codenamed Denali.

Ganó renombre cuando fue liado con IIS v3.0 una habitación del Web server en March1997.

Active server Pages(ASP) es una herramienta para crear las páginas dinámicas e interactivas del web. El ASP es una tecnología de microsoft que trabaja permitiendo que utilicemos la funcionalidad de un lenguaje de programación que genera el HTML para el webpage dinámicamente.

Cuando un archivo ASP se carga en un browser, es porque el servidor envió una página HTML. Puede parecer extraño, pero el servidor analiza y ejecuta todas las secuencias de comandos ASP del servidor antes de enviar el archivo, es decir, el servidor ejecuta todas las sentencias ASP y el resultado lo retorna en formato HTML, por lo que el usuario siempre recibe código HTML estándar en el browser.

Un archivo de páginas Active Server (ASP) es un archivo de texto con la extensión .asp que contiene cualquier combinación de lo siguiente:

- Texto
- Etiquetas HTML
- Secuencias de comandos del servidor

2.- Arquitectura ASP

En la arquitectura de asp el debe servidor interprete los scripts ASP, veamos como es el proceso de esta tecnología.

Necesitamos un usuario navegando, el browser en el que esta navegando llega a una página Asp, para que sea vea en el browser, necesita que suceda lo siguiente:

El browser hace una solicitud para ver la página al servidor web, este interpreta que en esa página hay scripts de Asp, entonces ejecuta dichos scripts, construye una página con la estructura HTML y la envía como respuesta al Browser.

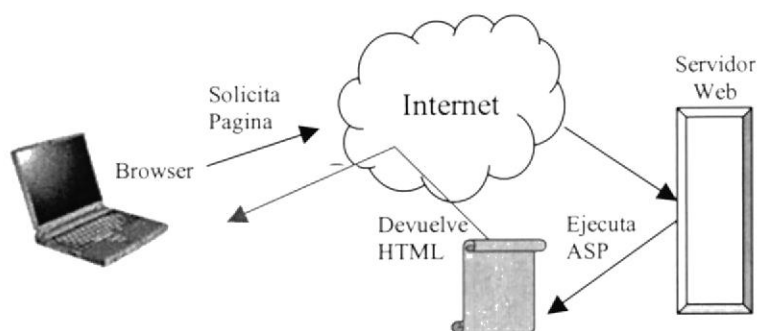


Diagrama D.1. Arquitectura ASP

2.1. Etiquetas HTML

Las paginas HTML están formadas por un lenguaje de etiquetas, estas mismas etiquetas se las puede usar en una página ASP.

2.2. Secuencias de comandos del servidor

Una secuencia de comandos del servidor es una serie de instrucciones que se utiliza para enviar al servidor Web comandos de forma secuencial.

En los archivos .asp, las secuencias de comandos se separan del texto y de las etiquetas HTML mediante delimitadores. Un delimitador es un carácter o una secuencia de caracteres que marca el principio o el final de una unidad. En el caso de HTML, dichos delimitadores son los símbolos menor que (<) y mayor que (>), que enmarcan las etiquetas HTML.

ASP utiliza los delimitadores <% y %> para enmarcar los comandos. Dentro de los delimitadores puede incluir cualquier comando válido dentro del lenguaje de secuencia de comandos que esté utilizando. El ejemplo siguiente muestra una página HTML sencilla que contiene un comando de secuencia de comandos:

2.3. VBScript

Las funciones básicas de VBScripts son iguales a las de Visual Basic

Variables

La forma de declarar variables es por medio de la palabra 'Dim', aclaremos que la declaración de variables no es necesaria, salvo que al inicio de nuestra página ASP pongamos la palabra clave 'Option Explicit' que hace que la declaración de variables sea necesaria si no nos daría un error.

2.4. El objeto Response

Este objeto ASP es del tipo integrado, es decir no tenemos que crear una instancia del mismo para poder utilizarlo, su función es enviar información al cliente (navegador).

Las principales propiedades del objeto Response son:

Buffer

Esta propiedad es del tipo boolean (true/false) e indica si la página se almacena en el buffer del servidor antes de ser enviada, es decir si esta true envía la página al cliente recién cuando esta toda cargada y si esta false la envía a medida que la procesa. Se debe utilizar antes de haber enviado algún resultado al cliente. Si el servidor utilizado tiene la versión de ASP 2.0 el valor predeterminado de esta propiedad es false y en cambio la versión es la 3.0 (Windows 2000) el valor predeterminado es true.

```
<%Response.Buffer = TRUE%>
```

Expires

Esta propiedad indica el tiempo, en minutos, que la página se guarda el cache del navegador, si este valor se pone a 0 no se utiliza el cache del navegador

```
<% Response.Expires = 5%>
```

ExpiresAbsolute

Esta propiedad indica hasta que día y hora la página se guarda en el cache del navegador.

```
<% Response.ExpiresAbsolute=#May 5, 2001 14:30:00# %>
```

Cookies

Creo una cookie en la maquina del usuario. Una cookies es un archivo con formato de texto en el cual el se puede guardar información. Las cookies por lo general se utilizan para controles de acceso, personalización de páginas y estadísticas. El siguiente ejemplo crea una cookie llamada 'fecha' la cual contiene el valor devuelto por la funcion 'date' que nos devuelve la fecha actual

```
<% Response.Cookies("fecha") = date %>
```

Esta cookie se mantiene en la maquina del usuario hasta que termine su session, borre las del navegador o se cumpla el tiempo establecido para esa cookies, a continuación veremos la forma de establecer la duración de una cookie

```
<% Response.cookies("fecha").expires="2/12/2001"%>
```

Esto hace que la cookie se borre en la fecha indicada, si creamos otra cookie llamada 'fecha' esta sobrescribirá a la otra

End

Este método detiene el proceso de la pagina ASP y envía lo ya procesado al navegador

```
<%Response.End%>
```

Flush

Este método envía el contenido del buffer al navegador siempre y cuando la propiedad Buffer se true, si la propiedad Buffer es igual a false se producirá un error

```
<%Response.Flush%>
```

Redirect (cadena_url)

Este método redirecciona la pagina actual a la pasada como parámetro.

```
<%Response.Redirect "otra_pagina.asp"%>
```

Write (cadena)

Este método escribe en el navegador la cadena pasada como parámetro, tengan en cuenta que también se pueden enviar etiquetas HTML y concatenar variable con el operador & (VBScript).

```
<%Response.Write ("<b>hola!!!</b>"%>
```

2.5. El objeto Request

Este objeto ASP es del tipo integrado, es decir no tenemos que crear una instancia del mismo para poder utilizarlo, por medio de este objeto puedes tener acceso a la información que el navegador pasa al servidor.

Las principales propiedades del objeto Request son:

Form

Esta propiedad es una colección que contiene todos los elementos de un formulario (<form>) pasado al archivo ASP con el método POST (<form method='post'>), la forma de acceder a el valor de un elemento del formulario es la siguiente: <%=Request.Form("nombre_elemento")%>

QueryString

Esta colección contiene todos los valores pasados como parámetros a una pagina ASP. La formas de pasar parámetros a una pagina ASP son varias una es por medio de formularios utilizando el método 'GET' y la otra es generándolo 'manualmente'.

http://www.servidor_xxx.com/pagina_asp.asp?nombre=jose&apellido=per ez Esa es la manera de pasar parámetros a una pagina ASP, luego de la direccion URL se pone el signo '?' y a continuación todos los parámetros con sus respectivos valores separados por el símbolo '&'.

ServerVariables

Esta colección nos da acceso a todas las variables de entorno del servidor.

```
<%
```

```
Response.Write (Request.ServerVariables("REMOTE_ADDR"))
```

```
Response.Write (Request.ServerVariables("SERVER_SOFTWARE"))
```

```
%>
```

2.6.- El objeto Server

Este objeto ASP es del tipo integrado, es decir no tenemos que crear una instancia del mismo para poder utilizarlo, por medio de este objeto puedes tener acceso a funciones útiles del servidor.

La propiedad y métodos principales del objeto Server es:

ScriptTimeout

Esta propiedad determina el tiempo máximo en segundos que se espera para ejecutar la página ASP. Si la página tarda más segundos en ser procesada se produce un error. El valor predeterminado es 90 segundos.

```
<% Server.ScriptTimeout = 25 %>
```

CreateObject (id_objeto)

Este es uno de los métodos más importantes y nos permite crear objetos a partir de los cuales vamos a trabajar. Como único parámetro se especifica el nombre del objeto a crear. Al crear el objeto este se le asigna a la variable establecida por medio de la palabra clave SET, esta variable hereda todos los métodos, propiedades y eventos del objeto creado.

```
<% Set BDD = Server.CreateObject("ADODB.Connection") %>
```

Execute (path)

Este método ejecuta una página ASP dentro de otra, su único parámetro es la ruta de la página a ejecutar. Este método solo está disponible en la versión 3.0 de ASP.

HTMLEncode (cadena)

Este método evita que la cadena pasada como parámetro sea interpretada como lenguaje HTML.

```
<% Response.Write Server.HTMLEncode("<b>sin negrita</b>") %>
```

MapPath (path)

Devuelve la ruta completa del archivo pasado como parámetro.

```
<% Response.Write Server.MapPath("calendario\dias.asp") %>
```

2.7.- El objeto Session

Este objeto ASP es del tipo integrado, es decir no tenemos que crear una instancia del mismo para poder utilizarlo, el mismo sirve para compartir información de un usuario entre las páginas del sitio. Por defecto el objeto session se crea al ingresar el usuario a nuestro sitio y se destruye cuando lo abandona o cuando pasan 20 minutos sin actividad del usuario.

Entre las propiedades del objeto Session destacan:

SessionID

Esta propiedad devuelve un ID que identifica a un usuario. Este ID es único, no se puede modificar y lo crea automáticamente el servidor.

```
<% Response.Write "ID Session = " & Session.SessionID %>
```

Timeout

Esta propiedad determina el tiempo en minutos que durara la sesion del usuario antes de ser destruida, por defecto su valor es de 20 minutos.

```
<% Session.Timeout = 30 %>
```

Y como principal método de este objeto podemos señalar:

Abandon

Este método provoca el abandono forzado de la session, destruye todas las variables de sesion. Ejemplo `<%Session.Abandon %>`

3.- Ventajas y desventajas

- Permite que los programas funcionen en lenguajes de programación que no son soportados en su browser, es decir, todo el conjunto de sentencias en marcadas o etiquetadas por el delimitador que indica que es ASP, son ejecutadas en el servidor para luego enviar al browser HTML estándar.
- Se puede corregir, cambiar o agregar dinámicamente cualquier contenido en una página web.
- Permite dar respuesta a las preguntas o datos que los usuarios ingresan en formas (cajas de texto, cuadros combinados,...etc.) de HTML.

- Se puede tener acceso a cualquier dato o base de datos y que los resultados sean mostrados en un browser.
- Otra de las ventajas es su simplicidad y velocidad.
- También proporciona seguridad ya que el código ASP no se puede ver en el browser.
- Puesto que los archivos ASP se vuelven claramente HTML, puede ser vistos en cualquier browser.
- La programación lista de ASP puede reducir al mínimo el tráfico de la red.
- Es muy versátil para la utilización de plantillas (templates) para la construcción de páginas web dinámicas.
- Como inconveniente destacar que al ser VBScript un lenguaje propietario de Microsoft, sólo es procesado por defecto por los servidores de esta casa comercial, ya sea en las versiones de Internet Information Server (IIS) o en las de Personal Web Server (PWS). Aunque es posible ampliar otro tipo de servidores mediante extensiones para conseguir que soporten páginas ASP.
- Con la tecnología ASP, Se puede agregar, modificar, quitar, obtener dinámicamente cualquier contenido en una página web.
- Se puede tener acceso a archivos que se encuentren en el servidor como una base de datos, a la que se le puede envía sentencias SQL y los resultados mostrarlos en el browser.

- El código ASP no se muestra en el browser, ya que el código ASP se ejecuta en el servidor y este solo envía la estructura HTML para que sea presentada en el browser.
- Es muy versátil para la utilización de plantillas (templates) para la construcción de páginas web dinámicas.
- Existe muchos Web site que brindar material que se puede utilizar como soporte ante cualquier tipo de desconocimiento, o duda en el desarrollo del proyecto.
- Es una tecnología que no necesita de una aplicación para poder diseñar un pagina web, con un editor de textos es suficiente, para cambiar de formato .Html a .Asp, solo es necesario cambiar la extensión.
- Nos permite permite escribir scripts sin importar si el browser soporta el lenguaje que se utiliza en ese scripts, es decir, todo el conjunto de sentencias que están etiquetadas por el delimitador que indica que es ASP, son ejecutadas en el servidor.

APENDICE E



- ***COMERCIO ELECTRONICO***
- ***MODELOS DE NEGOCIOS (B2C)***
- ***HOSTING***

Introducción

En los últimos años, la economía mundial ha realizado adelantos significativos en lo que se refiere a tecnología de información, manejado por el crecimiento de Internet. Esto ha contribuido a crear una economía rápida y confiable. Se ha sugerido, incluso, que estos adelantos podrían llevar a un boom de la nueva economía durante los próximos veinte y cinco años.

El comercio electrónico se lo define como cualquier actividad de intercambio comercial en la que las órdenes de compra / venta y pagos se realizan a través de un medio informático, los cuales incluyen servicios financieros y bancarios suministrados por Internet.

Con las nuevas tecnologías, el tiempo y la distancia dejan de ser obstáculos, los contenidos pueden dirigirse a una audiencia masiva o a un pequeño grupo de expertos y buscar un alcance mundial o meramente local. Las redes mundiales de información, como Internet no conoce fronteras y por el hecho de estar abiertas a todos los usuarios, reducen las iniciativas puramente nacionales, ante su carácter eminentemente internacional, tal como lo muestran las estadísticas cada 7 segundos un nuevo usuario aborda la superautopista de la información.

Como es de amplio conocimiento, el comercio electrónico, desde el punto de vista empresarial, tiene fundamentalmente dos esquemas: el Business to Business (B2B) y el Business to Consumer (B2C). A pesar que el B2B representa casi el 80% de las transacciones del comercio electrónico, con proyecciones cercanas al billón de dólares para el año 2003, la parte correspondiente al B2C es un 20% nada despreciable, que representará para fines del presente año, transacciones por un valor de 44 mil millones de dólares (según estadísticas del Gartner Group). Además, se estima que el B2C generará transacciones por casi 40,000 millones de dólares para el presente año, llegando a los 400,000 millones para el 2004 a nivel global, de acuerdo a estudios de Giga Information Group.

1.- Modelos de Negocios

1.1.- Modelos B2C

Los modelos de empresa B2C que existen en la actualidad de acuerdo a su presencia en el mundo físico se pueden clasificar de la siguiente manera:

- **Empresas bricks and mortar**, que son aquellas que existen en el mundo físico exclusivamente.
- **Empresas clicks and mortar**, las cuales tienen presencia en el mundo físico y en el mundo virtual, Ej.: Barnes & Noble, E. Wong, SagaFalabella.
- **Empresas virtuales**, las cuales tienen presencia como vendedores exclusivamente en el mundo virtual; aunque cuentan con oficinas, almacenes, etc., no tienen tiendas constituidas. Ej.: Amazon.com, Mallperu.com. Diacos.com.
- **Tiendas virtuales especializadas**: Este tipo de tiendas opta por una estrategia de marketing donde su ventaja competitiva radica en el enfoque de venta de determinado tipo de productos a un sector específico. Así por ejemplo, contamos con Espiral.com, empresa dedicada exclusivamente a la venta de libros en línea, en especial a consumidores latinoamericanos.
- **Tiendas virtuales diversificadas**: Este es un tipo de empresa que pudo haber tenido en mente nacer como una tienda por departamentos virtual, o sencillamente ha perdido el enfoque de su negocio, y por ende ha perdido su posicionamiento por una tendencia de extensión de línea.. Veamos a Amazon.com; empezó siendo una librería on-line, y ahora también vende CDs, software, juguetes e incluso muebles .

- **Multitiendas o Malls virtuales:** Es una tienda virtual que contiene sub-tiendas dentro de su página web. Estas sub-tiendas se tratan por lo general de otras empresas que han hecho joint-ventures con la principal. El concepto de multitiendas no debe confundirse con una página que contenga los *links* hacia otras tiendas; en el caso de las multitiendas, todas o casi todas las tiendas se encuentran en un mismo servidor. Este tipo de soluciones comerciales integrales orientadas al consumidor final se encuentran aún en fase de experimentación, pero se espera que sea la manera más económica y fácil de implementar un negocio B2C en un futuro próximo.

- **Portales:** Algunos portales están empezando a contar con áreas dedicadas exclusivamente al comercio B2C y lo hacen a través de un esquema similar al de mall virtual, siendo un ejemplo de ello Yahoo! Shopping.

- **Páginas de servicios pagados:** Tales como servicios noticiosos, turísticos o financieros.

Se denomina B2C al comercialización de productos entre empresas vía electrónica, este ha sido uno de los conceptos más alentadores y respaldados financieramente en Internet.

La entrega de productos , o cualquier servicio vinculado entre empresas, dependerá en gran parte de unas de las variables más observadas en cualquier ámbito: **costos**. Existen empresas que como verticalnet.com que toman la postura de integrar los diversos componentes de una industria para generar una sinergia entre sus productores y consumidores logrando una conexión negocio a negocio (Business 2 Bussiness)

La primera labor para generar portales de este tipo , sería acumular a los diversos proveedores, algunas razones por las cuales sería conveniente unirse a un portal de este tipo:

- La posibilidad de conquistar a ciertos clientes que en la actualidad puedan estar fuera de su alcance
- Ofrecer mayores descripciones acerca de los terrenos y servicios ofrecidos ya sea vía fotografías o video; en el proceso de reemplazar a los documentos de papel por un catálogo dinámico en el web.
- La oportunidad de competir en todo el mercado sin importar regiones geográficas o contactos con la industria.

Estructura

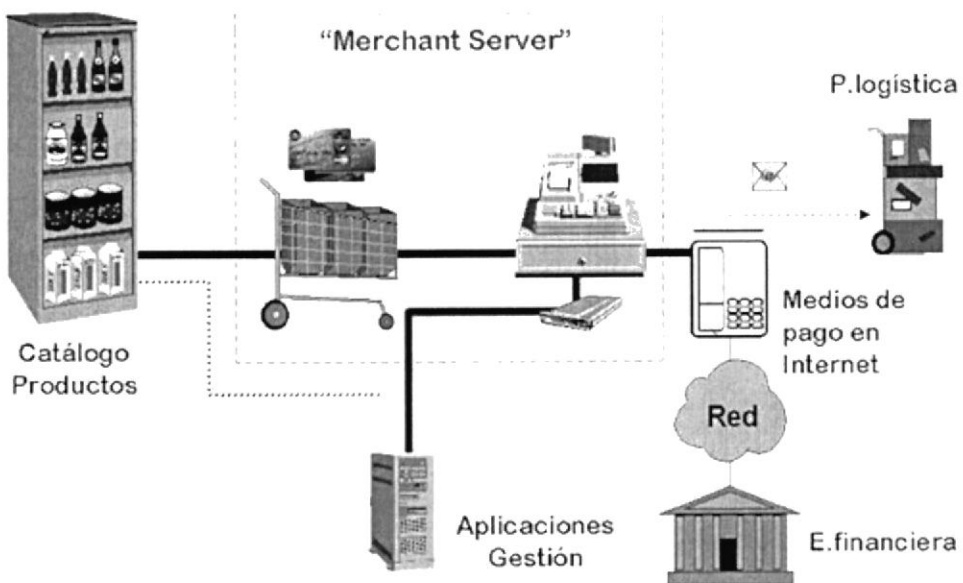


Figura E.1 Estructura del Comercio electrónico

El **catálogo de productos** que recoge nuestra oferta de productos y servicios ha de estar disponible en formato digital y puede ser de dos tipos:

- **Estático:** desarrollado en página HTML que se muestran cuando el usuario las va seleccionando. Es más práctico cuando el número de productos es muy reducido y se van a realizar pocas modificaciones al catálogo.

- **Dinámico:** catálogos digitales con referencias almacenadas en: ficheros de texto(más sencillos), hojas de cálculo o bases de datos (más complejos). Las páginas se crean automáticamente en función de las selecciones que hace el cliente. Se prefiere este tipo de productos cuando el número de referencias y la cantidad de modificaciones son elevados.

El catálogo de productos y las páginas HTML que describen los servicios y proporcionan información sobre la empresa, se almacenan en el disco duro de un servidor. Este servidor puede estar compartido con otros usuarios o ser propiedad de la empresa. En cualquiera de los dos casos y, salvo excepciones, residirá en las instalaciones de un ISP que es quien proporciona el Servicio de Hospedaje ("hosting") y de conexión del servidor a Internet. También es posible que este proveedor gestione el mantenimiento del dominio (nombre de nuestro servidor) y de otros elementos de comunicaciones (ejemplo: dirección IP, buzones de correo electrónico, comunicaciones especiales,...).

2.- Comercio Electrónico

2.1.- Leyes en el Ecuador

Actualmente, en nuestro país existe una ley aprobada que controla las transacciones en la red , esta Ley tiene como filosofía principal los siguientes puntos:

a.- Formalizar y establecer la legalidad de las transacciones de todo tipo realizadas por medios electrónicos. Para esto se reconoce la validez y eficacia jurídica de instrumentos como firmas electrónicas, documentos electrónicos, mensajes de datos, contratos electrónicos, certificados y certificadores digitales, etc.

b.- Regular y normar las actividades de los emisores de certificados electrónicos y otras actividades relacionadas con el Comercio

Electrónico, estableciendo reglas, responsabilidades técnicas, legales y formalidades a contemplar.

c.- Establecer las condiciones jurídicas de los contratos por medios electrónicos en lo relacionado a lugar de ejecución, jurisprudencia vigente, derechos y responsabilidades de los contratantes de acuerdo y en compatibilidad con las actuales normas vigentes y los modelos de negocios por medios electrónicos.

d.- Establecer expresamente los derechos y deberes de los consumidores en los negocios por medios electrónicos.

e.- Tipificar las conductas delictivas de varios tipos que pueden aparecer y determinar las penas y sanciones respectivas.

f.- Observar los principios básicos de legislación internacional para permitir la validez de instrumentos de otras jurisdicciones previo el cumplimiento de requisitos básicos fundamentales sin imponer trabas o restricciones de ningún tipo al comercio internacional o sus componentes por medios electrónicos. Esto se logra observando los principios de la ley modelo de UNCITRAL legislación adoptada como base por la mayoría de legislaciones a nivel internacional y que contempla:

- Principio de la neutralidad tecnológica
- No discriminación del Comercio Electrónico y sus diferentes modalidades
- Internacionalización del Comercio y sus formalidades por medios electrónicos.

g.- La ley es general, no reglamentaria y abierta. Así por ejemplo, una vez que los documentos electrónicos sean válidos, corresponderá a las instituciones responsables reglamentar que condiciones deben cumplir estos documentos para considerarse una factura electrónica, una oferta, un contrato electrónico, un vale, voucher o pagaré electrónico, una orden

de compra legal, etc. Igualmente al determinar la validez de las firmas electrónicas, éstas se asimilan a la firmas manuscritas en todos los cuerpos legales que las contemplan y adquieren la misma eficacia e implicaciones jurídicas.

h.- Respecto de los medios electrónicos de pago, estos se basan igualmente en las condiciones básicas que se contemplan en la ley como son, las condiciones técnicas del envío y recepción de datos y mensajes, los certificados electrónicos y otros elementos relacionados y la validez de los contratos, firmas y documentos electrónicos. La reglamentación de la actividad financiera no se establece en la ley y corresponderá a las instituciones relacionadas como Banco Central, SuperIntendencia de Bancos y otros, establecerla.

i.- En temas de especialidad de otros cuerpos legales, la ley propone modificaciones o adiciones para adaptarlas al nuevo entorno técnico y/o se refiere a ellas para legislar lo que corresponda. Tal es el caso del código penal, la ley de propiedad intelectual o la ley de protección al consumidor.

2.1.1 Alternativas para el Ecuador

En vista de que en el Ecuador las leyes, y la infraestructura tecnológica no se encuentran aun tan desarrolladas como en otros países que pueden aplicar la globalización, se ha ideado y se usa con gran frecuencia el esquema de las cotizaciones online, o las ordenes de pedido, que se encuentran bajo el esquema de no realizar una transacción completamente online, sino con la ayuda de terceros por ejemplo contar el servicio de una empresa de entrega por ejemplo: "Servi-entrega", la cual tendrá la función de entregar el producto y recoger el cheque o el recibo firmado con el número de la tarjeta de crédito. Actualmente hay una nueva alternativa que es las empresas proveedoras de cuentas de tipo mercantil o mas comúnmente conocidas

como "Merchan Account" como por ejemplo "PayPal", el mismo que se describirá más adelante con mayor detalles.

El primer esquema mencionado, funciona básicamente en los siguientes pasos:

- El cliente encuentra su artículo en el catálogo
- Hace su orden o cotización en línea
- La empresa entrega personalmente el artículo con sus indicaciones, y realiza el cobro del mismo.

Como ejemplo de algunos sitios ecuatorianos que manejan el esquema de órdenes de pedido o cotizaciones online , se encuentran los siguientes:

- www.mochejeans.com
- www.e-giftco.com
- www.olmedoalvarez.com
- www.ventasvirtuales.com

2.2 Ventajas

Las ventajas y oportunidades del comercio electrónico se equiparan a las de la red misma en:

- Presencia y elección globalizada;
- Aumento de competitividad y calidad de servicios;
- Adecuación generalizada/productos y servicios personalizados;
- Cadenas de entrega más cortas o inexistentes;
- Respuesta inmediata a las necesidades;
- Reducción de precios;
- Nuevos negocios, productos o servicios, etc.

Ya desde una perspectiva jurídica algunas de las ventajas del comercio electrónico son:

- Homologación con documentos de formato tradicional.
- Legalidad de mensajes de datos o documentos electrónicos (mensajes de correo electrónico, ordenes de compra, etc.)
- Los certificados digitales que garantizan tecnológicamente la identidad inequívoca de su propietario (receptor o emisor)
- Se evita la suplantación, vía infraestructura tecnológica o a través del registro público de claves.
- Imposibilidad de revocación: afirmar eventualmente que no se recibió o no se envió el mensaje de datos, hecho que es fácilmente comprobable a través de tecnología.

No es exacta la apreciación de que el comercio electrónico solo sirve para operaciones de comercio exterior. Prueba de ello es el gran interés que varias empresas nacionales (en distintos sectores tienen en el tema), por la perspectiva de crecimiento de ventas y oportunidades que tienen dentro de nuestro país.

Sin embargo, la inseguridad jurídica que rodea al comercio electrónico ha impedido su desarrollo. La acogida que tiene lamentablemente se ha visto truncada, limitando así las transacciones en la red, al acceso a portales con fines informativos y compra o utilización de productos y servicios con compañías que operan desde Estados Unidos y que por tanto pueden acceder a la denominada "*merchant account*" que es la única vía posible para poder cobrar vía tarjeta de crédito las operaciones originadas en el Internet.

En su generalidad, las leyes que regulan el comercio electrónico no son tecnológicamente neutras. Por el contrario: no podrían existir sin el apoyo y afluencia de una infraestructura tecnológica con la que se funde, a tal punto que solamente a través de medios tecnológicos puede certificarse la integridad y autenticidad de una firma digital o de un documento digital, mecanismos éstos que día a día van perfeccionándose rezagando indiscriminadamente a la ley que los regula.

En el caso del Ecuador, al igual que Colombia y otros países con legislaciones sobre comercio electrónico como Italia, Alemania, España, se ha adoptado como esquema de seguridad la Infraestructura de Clave Pública (Public Key

Infraestructure). Esto significa que la ley establece la existencia de entidades certificadoras, legalmente facultadas para generar firmas digitales, sobre la base de dos claves: una de conocimiento público y otra secreta.

Una de las principales ventajas que se puede mencionar en el esquema actual que maneja el Ecuador, es que estamos avanzando en este medio del comercio electrónico, aunque no se realiza el esquema completamente como en los países que están más avanzados tecnológicamente y formando parte de la globalización, pero damos oportunidad y facilidad a las pequeñas y grandes empresas a comercializar sus productos tanto a manera local como internacional, una gran ventaja es que contamos con la ayuda de otras empresas que se encargan de toda la parte transaccional como lo mencionamos anteriormente los merchant account, o la empresa verisign para convertir su sitio web seguro.

2.3 Desventajas Problemática

A continuación detallaremos las pocas desventajas que se le han determinado al comercio electrónico B2C

- Costo inicial de la tecnología
- El reemplazo del computador por el recurso humano y el producto de una imagen : Nunca será lo mismo el trato persona a persona que persona a computadora, la presentación de un producto al ser atendido, y el contacto físico con el artículo en cuestión son factores que quizás aunque el cliente tenga todos los servicios posibles, él se puede sentir incomodo.
- Un medio sin alcance para todo público : La oportunidad de que una persona vea, visite, y compre en una tienda virtual no es muy alta, desafortunadamente en Ecuador no se cuenta con una estructura económica y mucho menos con una infraestructura tecnológica que se tenga al alcance de la gente lo cual hace un privilegio de pocos y una desventaja de muchos.

- La desconfianza ante los medio electrónicos: Las personas tienen desconfianza en poner su numero de tarjeta de crédito en su pagina Web, donde se tiene el mito de que esos datos son fáciles de obtener.

A manera mas especifica a continuación mencionaremos ciertas desventajas con respecto a los medios de pago en internet en nuestro país:

- Si la empresa quiere realizar ventas online de manera segura tiene dos opciones una de ellas es conseguir su "Merchant Account", la cual suele ser de un alto costo monetario y tecnológico, una vez que posea esta cuenta debe contratar el servicio de Verisign o alguna empresa similar que le ofrezca los algoritmos de encriptación necesarios para respaldar toda la seguridad en sus transacciones.
- En caso de que la empresa desea enfocarse en el esquema de órdenes o cotización online, debe contratar el servicio de una empresa de entrega por ejemplo : Servientrega.

Podemos concluir que las pequeñas desventajas que hemos mencionado anteriormente con respecto al B2C, son superables y sacrificables si comparamos con las enormes ventajas que ofrece el mismo, además gracias al B2C tenemos la oportunidad de formar parte de la globalización, y de esta manera ser parte activa del crecimiento de nuestro país.

3.- HOSTING

Todas las páginas de Internet que usted y sus clientes visitan a diario, deben de estar almacenados dentro de un servidor web, que no es otra cosa que una computadora con capacidad de atender peticiones a través del protocolo HTTP.

Las grandes empresas poseen centros de cómputo que les permiten atender las necesidades de toda su corporación en un solo lugar. Sin embargo, los costos asociados de mantenimiento, conectividad y desarrollo de estos sitios, hacen imposible de mantener un centro de esta naturaleza a un negocio mediano o

pequeño, como es el a que está orientado este software para construir tiendas virtuales.

Como consecuencia de ello, hace aproximadamente 11 años, en los Estados Unidos se popularizaron los silos de servidores, lugares en donde se coloca una gran cantidad de servidores, conectados a través de enlaces de muy alta velocidad, como consecuencia de esta tendencia, surgió el servicio de Hospedaje virtual

El Hospedaje virtual (conocido generalmente como **Hosting**), es una excelente alternativa para todas aquellas empresas que desean tener presencia completa en Internet, pero que no tienen el presupuesto necesario para implementar un servidor propio.

El hosting consiste en dividir la capacidad y el espacio de un servidor en varias partes, y hacer que cada una de ellas pueda funcionar como un servidor web "virtual". De esta forma, el costo de un servidor propio en un su empresa (de aproximadamente 400 dólares mensuales) se logra reducir drásticamente, sin afectar el rendimiento promedio del sitio.

A continuación listamos algunas direcciones de proveedores de hosting, y algunos planes para que tenga referencia de los costos:

Tabla E.1.-Lista de Precios de Hosting

Empresa	Servidor	Espacio	Transferencia	Precio anual
Arsys Internet	NT-2000	400 Mb	5 Gb	\$150.00
Arsys Internet	Unix-Linux	400 Mb	5 Gb	\$140.00
Urugrafix	Linux-Unix	400 MB	12 Gb	\$150.00
EasySpace	NT-2000	500 MB	5 GB	\$280.00
EasySpace	NT-2000	200 MB	5 GB	\$195.00

EasySpace	NT-2000	100 MB	5 GB	\$ 95.88
EasySpace	NT-2000	30 MB	5 GB	\$ 45.00

Tabla E.2.- Modelo de Alternativas que los proveedores de hosting ofrecen

Características	Plan A	Plan B	Plan C	Plan AA
Espacio en disco	50 MB	50 MB	50 MB	100 MB
CGI-BIN (Perl)	-	SI	SI	SI
Panel de control	SI	SI	SI	SI
Cold Fusion	SI	SI	SI	SI
ASP	SI	SI	SI	SI
Cuentas FTP Privadas	1	1	1	2
FTP Anonimo	-	-	-	SI
Buzones de Mail OP2.	10	10	20	50
Buzones para e-mail gratis	-	-	-	-
Transferencia mensual (GB)	1	1	1	1
Cuartos de Chat	-	-	-	-
Servidor de Foros	-	-	-	-
Tiendas AbleCommerce TM	-	-	-	-

Servidor dedicado Spectra	-	-	-	-
Servidor de Estadísticas	-	SI	SI	SI
Control remoto DNS	-	-	-	SI
Bases de datos MS Access (97/2000)	-	-	1	-
Bases de datos SLQ 7	-	-	-	1
Autenticador de directorios	-	-	-	-
Servidor dedicado real audio	-	-	-	-
Certificado SSL	-	-	-	-
Costo mensual (USD)	\$33.00	\$38.00	\$52.00	\$116.00
Plazos minimos de contratación (meses)	4	4	4	4

Entre los principales componentes de estos planes que ofrecen los proveedores de hosting mencionados en la lista anterior tenemos:

- Cuentas de correos
- Autoresponder
- Redirecciones
- Estadísticas completas
- Cgi-bin propio
- Soporte de Base de Datos
- Extensiones de Front Page

- Soporte de ASP
- Soporte Send mail

En el internet existen muchas direcciones de empresas, que ofrecen el servicio de alquiler de hosting para la publicación de su su sitio web, las mismas que tienen diferentes tipos de planes de acuerdo a las necesidades del cliente, para efectos de nuestro proyecto el tipo de hosting que se escoja debe permitir publicar base de datos; debido a que no utilizamos las tecnologías cgi y php, no existe ningún problema si el proveedor no le permite el uso de estas 2 tecnologías, ya que para el funcionamiento de envío de email en nuestro proyecto sólo se utiliza tecnología ASP.

En caso de que el hosting que se escoja está fuera del Ecuador, se debe realizar la transacción online, para obtener este servicio, posteriormente este proveedor enviará por email, los datos de número IP, user y password que servirán para publicar su tienda virtual por medio de una sesión FTP o cualquier otro software de transferencia de datos.