

*Universidad Nacional Autónoma de Nicaragua*  
*Unan-León*  
*Facultad de Ciencias Puras*  
*Computación*



*Título:*

*Comercio Electrónico.*

*Monografía para optar al título de Licenciado en  
computación.*

*Autores:*

*Luis Salvador Matamoros Cáceres.*

*Lesber Israel Gutiérrez Vilchez.*

*William José Osejo Palma.*

*Tutor:*

*Msc. Martín Ibarra Padilla.*

*León, Nicaragua 2003.*



## **DEDICATORIA**

Este trabajo lo dedico a mi padre, Salvador Matamoros y a mi madre, Lesbia Cáceres. Gracias por todo, por haberme brindado un buen ejemplo y por haberme querido tanto. Todo lo que he logrado no hubiese sido posible sin su ayuda. Que Dios los bendiga.

A mis hermanos Oscar, Maura Elisa, Wilmer y Vladimir, quienes también vienen estudiando y espero que logren sus metas.

Luis Salvador Matamoros Cáceres.



## AGRADECIMIENTO

En primer lugar agradezco a Dios por haberme dado la vida y todos los conocimientos para poder culminar con mi carrera universitaria.

Agradezco profundamente a mis padres **Salvador Matamoros** y **Lesbia Cáceres** quienes fueron el pilar y la fuente de inspiración para que yo pudiera sobresalir en mis estudios.

A mis amigos y compañeros de este trabajo monográfico Lesber y William que siempre me brindaron su apoyo incondicional a lo largo de estos años.

Luis Salvador Matamoros Cáceres.



## DEDICATORIA

**Esta monografía está dedicada:**

**A DIOS** nuestro padre celestial, nuestro guía y luz, a quien debemos nuestra existencia y a nuestra madre y madre de Dios **Virgen Maria** porque ellos me dieron las fuerzas, la sabiduría y la fortaleza en los momentos mas difíciles de la elaboración de esta monografía.

**A mis padres**, por haberme brindado su apoyo incondicional a lo largo de la carrera, ya que con gran esfuerzo permitieron que llegara a este momento como es concluir mi trabajo monográfico.

**A mis hermanos**, porque al igual que mis padres siempre estuvieron con migo en los mementos difíciles para brindarme su incondicional apoyo.

Lesber Israel Gutiérrez Vilchez.



## AGRADECIMIENTO

Realmente es imposible agradecer en este pequeño espacio a todas las personas que de una u otra manera me ayudaron en la culminación de mi carrera, también pienso que no tengo palabras para expresar lo agradecido que estoy con todos ellos.

Agradezco a **Dios** y a la **Virgen Maria** porque gracias a ellos existimos en esta vida y porque al creer en ellos siempre encontré las fuerzas necesarias en los momentos difíciles de mi carrera y en el transcurso de la elaboración de este trabajo monográfico.

Agradezco muy especialmente a mis padres por ser ellos quienes como padres y amigos me han orientado por el camino de la vida a través de sus valiosos consejos.

A mis hermanos por el apoyo incondicional que me brindaron siempre y en todo momento.

A todos los profesores que a largo de la carrera supieron transmitirme sus conocimientos para llegar a formar el profesional que hoy soy.

A todos mis amigos y compañeros de clases que de una u otra manera me ayudaron durante toda mi carrera y a la elaboración de esta monografía.

A mi tutor: Martín Ibarra por aceptar ser guía en la elaboración de este trabajo, por los consejos que siempre nos brindo para que nuestra monografía pudiera llegar a culminarse.

Así pues, agradezco a todas las personas que siempre me brindaron su apoyo, a las personas de las que siempre escuché palabras de aliento en los momentos más difíciles y que gracias a ellos pude salir adelante, muchísimas gracias porque sin su apoyo no hubiese llegado a ser lo que soy ahora, a todos ellos “GRACIAS TOTALES”.

Lesber Israel Gutiérrez Vilchez.



## DEDICATORIA

El presente trabajo monográfico lo dedico en primer lugar a Dios por haberme dado la vida, y la oportunidad de haber culminado satisfactoriamente mis estudios Universitarios.

A la memoria de mi padre Sr. Esteban Filemón Osejo Oviedo, quien fue y seguirá siendo una fuente de motivación y valor para superar las situaciones adversas que se presenten en la vida.

A mi Madre Sra. Luz Palma por haberme brindado su apoyo moral y un respaldo incondicional en el transcurso de mi carrera Universitaria.

A mis hermanos Jose Esteban Osejo, Yamileth Osejo y Paula Osejo por la confianza y orientación que me brindaron.

William José Osejo Palma.



## AGRADECIMIENTO

Agradezco a Dios por haberme dado las fuerzas y el conocimiento en la culminación de mis estudios.

A todos los profesores que de una u otra manera estuvieron involucrados en el proceso educativo y formativo de mi persona en el transcurso de mi carrera Universitaria, en especial a nuestro tutor Msc. Martín Ibarra por todas las facilidades y orientaciones brindadas en el desarrollo de nuestro trabajo monográfico.

Agradezco en especial a mis compañeros y amigos de monografía por haberme brindado su apoyo incondicional no solo en el aspecto formativo como profesionales sino también en el aspecto personal, de igual manera a todos los compañeros de clases por todas las vivencias compartidas.

Gracias a todas aquellas personas que colaboraron de una u otra manera en la conclusión de este proceso profesional, gracias amigos de San Juan de Dios por haberme brindado un poco de serenidad y paz personal.

William José Osejo Palma.



**CONTENIDO**

**I. INTRODUCCIÓN..... 8**

**II. ANTECEDENTES..... 11**

**III. JUSTIFICACIÓN ..... 12**

**IV. OBJETIVOS ..... 13**

**V. MARCO TEÓRICO ..... 14**

**VI. MATERIAL Y MÉTODO ..... 37**

**VII. DISEÑO ..... 54**

**XIII. CODIFICACIÓN ..... 62**

**XIII. CONCLUSIÓN..... 71**

**IX. BIBLIOGRAFÍA ..... 73**

**X. ANEXOS..... 74**





## I. INTRODUCCIÓN

En los últimos años Internet ha venido siendo uno de los medios más utilizados para la investigación económica, a demás es usado por muchas organizaciones o empresas para comercializar sus productos.

En un principio existían solamente sitios Web basados en páginas estáticas, o sea que únicamente se mostraba información sin que el usuario pudiera transferir datos, estas páginas solo permitían que las empresas pudieran promocionar sus productos, pero no permitían hacer ventas por Internet, pero se presentó la necesidad de que a demás de promocionar los productos estos pudieran ser vendidos usando Internet. Esto conlleva a la utilización de tecnologías que permitan recoger información proveniente de clientes y almacenarla en una base de datos en un servidor, o extraer la información de una base de datos y presentársela al cliente.

Con la expansión de la tecnología informática, Internet, los consumidores buscan satisfacer sus necesidades a través de ella, sencillamente comprando sin moverse de sus casas. Hoy en día pequeñas y medianas empresas ofrecen sus productos por la red, lo cual aumenta sus ofertas y posiblemente la demanda de dichos productos, o sea que Internet es un medio sencillo y económicamente factible para realizar ventas.

Nuestro trabajo se basa en la creación de una página Web para vender por Internet, esta página Web va a permitir que la empresa que adquiera esta aplicación pueda promover y vender sus productos, ya que actualmente son miles de usuarios los que tienen acceso a Internet. Esta página Web permitirá que al cliente le sea fácil realizar la compra ya que tendrá integrada también una ayuda en la que se detallará como realizar la compra en la página Web.



Estas páginas Web también son conocidas como "*tiendas virtuales*". Estas consisten de un programa que le permitirá vender sus productos a través del Internet y ofrece las siguientes funciones entre otras:

- Administra los productos que se ofertan.
- Calcula impuestos.
- Calcula los costos de envío.

Para la creación de esta aplicación haremos el uso de ASP (Pages Active Server) la cual es una tecnología diseñada para la creación de páginas dinámicas, en el cual se basa nuestro trabajo (Comercio Electrónico). En ASP, todas las páginas Web pueden ser diseñadas con editores de HTML, puesto que las instrucciones ejecutables y el código HTML están suficientemente delimitados. Así mismo, pueden utilizarse diversos lenguajes de programación de la funcionalidad de las páginas activas. Entre estos lenguajes se encuentra Visual Basic Script y Java Script.



## II. ANTECEDENTES

La globalización y desarrollo tecnológico implica una serie de cambios y las diversas entidades comerciales no se pueden mantener al margen. La manera convencional de las diferentes empresas de ofertar sus productos, actualmente genera una serie de inconvenientes a los diversos clientes en general y por consiguiente afecta el nivel de ventas de la mismas. Uno de estos inconvenientes es la división geográfica. La empresa oferta sus productos pero esta publicidad se difunde mayormente a los clientes de esa localidad, pero ¿Qué sucede con los posibles clientes interesados de otra ciudad o región? Tendrían que viajar hasta el lugar donde se encuentre dicha empresa para visualizar dicho producto y posiblemente adquirirlo, implicando pérdida de tiempo y dinero.



### **III. JUSTIFICACIÓN**

Nuestro proyecto ofrece una alternativa de publicidad y mercadeo a las entidades comerciales contribuyendo a su desarrollo económico, a través de la creación de una página web, la publicación de la misma en Internet permitirá que las empresas oferten sus productos no solamente en la región o ciudad en donde se encuentren, sino dicha publicidad se visualizará en cualquier parte del mundo en donde se encuentre un computador conectado a Internet generando un mayor interés en posibles clientes, al ahorrarles muchos inconvenientes, por consiguiente se generaría un posible aumento de ventas en la empresas.



## **IV. OBJETIVOS**

### **GENERAL**

- Diseñar una página Web para realizar ventas por Internet utilizando una de las diversas tecnologías para la creación de páginas dinámicas.

### **ESPECÍFICOS**

- Proporcionar facilidades de compra a los usuarios, a través de Internet.
- Permitir al cliente comprar sus productos desde cualquier lugar donde halla Internet y recibirlos en su casa.
- Incentivar a las empresas a utilizar el Comercio Electrónico, como una alternativa importante de ventas y publicidad.



## V. MARCO TEÓRICO

Para comprender mejor la elaboración de este trabajo debemos conocer ciertos términos, a continuación describimos los siguientes conceptos:

**Red:** es un sistema de comunicación entre computadoras, que permite compartir información y recursos.

**Internet:** es una enorme red que conecta redes y computadoras distribuidas por todo el mundo, permitiéndonos comunicarnos, buscar y transferir información.

**Servidor:** Ordenador con ciertas características predominantes que mediante un sistema operativo de red permite administrar y controlar ciertos ordenadores pequeños llamados clientes, con el objetivo de que estos compartan recursos del servidor y puedan al mismo tiempo comunicarse entre sí para algunos propósitos.

**Cliente:** Ordenadores pequeños, que mediante un programa y algunos periféricos son capaces de acceder al servidor para algunos propósitos (en algunos casos se debe entender como cliente a un navegador de Internet, ya que es un programa que ese ejecuta en el cliente haciendo una conexión a un servidor Web).

**Cliente/Servidor:** Son los sistemas que pueden compartir los recursos de sistemas muy distintos de una organización e incluso mas allá de los límites de una organización.

**DBMS(Data Base Management System):** Sistemas generadores de Bancos o Bases de Datos. Son programas que se ocupan de acceder y actualizar las bases de datos, como Access, SQL Server, Visual Fox Pro, Oracle. Actúan como una interfaz entre el programa de aplicación y la Base de Datos. Por ejemplo, cuando un programa debe actualizar una información en la Base de Datos, no lo hace directamente, sino que le pide al DBMS hacerlo, pues este es quien conoce como es la base de datos por dentro.



Para llevar a cabo el desarrollo de la aplicación utilizamos el “Ciclo de vida básico” o “Modelo en Cascada”.

Este modelo descompone el desarrollo en varias etapas:

**Ingeniería y modelado de sistemas / información:** el software siempre forma parte de un sistema más grande, el trabajo comienza estableciendo requisitos de todos los elementos del sistema y asignando al software algún grupo de estos requisitos.

**Análisis de los requisitos del software:** permite especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software, proporciona modelos que pueden traducirse en el diseño de datos, arquitectónico, de interfaz y procedimental. Finalmente proporciona al diseñador y al cliente los medios para valorar la calidad una vez que se ha construido el software.

**Diseño:** se centra en cuatro atributos distintos de un programa: estructura de datos, arquitectura del software, representaciones de interfaz y detalle procedimental.

**Generación de código:** realiza la tarea de traducir el diseño en una forma legible por la máquina.

**Pruebas:** se centra en los procesos lógicos internos del software asegurando que todas las sentencias se han comprobado, la realización de las pruebas es para la detección de errores y sentirse seguro de que la entrada definida produzca resultados reales de acuerdo con los resultados requeridos.

**Mantenimiento:** el software puede sufrir cambios una vez entregado al cliente, por que se pueden encontrar errores, o por que el software debe adaptarse a los cambios de su entorno externo o por que el cliente desea hacerle cambios.

**Fases del diseño.**

**Diseño de datos:** la actividad principal del diseño de datos es seleccionar representaciones lógicas de objetos de datos identificadas durante la fase de definición de requisitos.

**Diseño Arquitectónico:** su objetivo primario es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos. Combina la estructura del programa y las estructuras de datos definiendo interfaces que permiten el flujo de datos a través del programa.

**Diseño de interfaz:** el diseño de interfaz se centra en tres áreas importantes: el diseño de interfaz entre los módulos software, diseño de interfaz entre el software y otros productos y consumidores no humanos de información y el diseño de la interfaz entre el hombre y la computadora.

**Diseño procedimental:** debe especificar los detalles procedimentales sin ambigüedades, por lo que no se recomienda utilizar solamente el lenguaje natural, sino que se debe utilizar un modo más restringidos de representar el detalle procedimental ya sea con algoritmos o diagramas de flujos.

**El Diagrama de Flujo de Datos (DFD):** Es una técnica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida. Se puede utilizar el DFD para representar un Sistema o Software a cualquier nivel de abstracción.

**Diccionario de Datos:** Es un listado organizado de todos los elementos de datos que son necesarios para el sistema, con definiciones precisas y rigurosas que permiten que el usuario y el analista del sistema tengan una misma comprensión de las entradas, salidas, de los componentes de los almacenes y de los cálculos intermedios.





## **Comercio Electrónico**

El comercio electrónico es cualquier actividad de intercambio comercial en la que las órdenes de compra / venta y pagos se realizan a través de un medio telemático, los cuales incluyen servicios financieros y bancarios suministrados por Internet. El comercio electrónico es la venta a distancia aprovechando las grandes ventajas que proporcionan las nuevas tecnologías de la información, como la ampliación de la oferta, la interactividad y la inmediatez de la compra, con la particularidad que se puede comprar y vender a quién se quiera, y, dónde y cuándo se quiera. Es toda forma de transacción comercial o intercambio de información, mediante el uso de Nueva Tecnología de Comunicación entre empresas, consumidores y administración pública.

## **Tienda Virtual**

Una tienda virtual es una aplicación orientada al comercio electrónico y dispuesta físicamente en la red Internet. El concepto virtual tiene que ver con el sistema electrónico sobre el que se han construido los cimientos de nuestra tienda. Al tratarse de un concepto que físicamente no existe se establece por norma la idea de concepto virtual.

## **¿Quién puede vender por Internet?**

Todo el mundo. Internet no es monopolio de organizaciones privadas, públicas o país alguno.

## **¿Quién puede administrar un comercio electrónico por Internet?**

Todos aquellos que dispongan de unos conocimientos básicos sobre el manejo de Internet (correo electrónico y servicio WWW). A medida que el comercio electrónico se ha hecho más popular han aparecido nuevas herramientas de gestión y administración remota de tiendas virtuales que son fáciles, cómodos y rápidos de manejar. A menudo se recomienda que el comerciante tenga conocimientos básicos en diseño de Bases de Datos.



Conocimientos que, sin duda, le ayudarán a entender mejor el funcionamiento interno de la aplicación.

### **¿Qué organismo controla el comercio electrónico por Internet?**

Aunque existen múltiples asociaciones y organismos alrededor del mundo que estudian y analizan el comercio electrónico por Internet no existe organismo alguno que controle este tipo de actividad por Internet. Al no existir control sobre esta actividad el comerciante puede decidir en todo momento que tarjetas de crédito ha decidido aceptar, que solución de comercio electrónico va a implementar en Internet y como va a diseñar el sitio Web empresarial. Todo está permitido no existiendo imposición alguna ni impuesto adicional para llevar a cabo esta actividad.

### **Protocolo de comunicaciones**

Un protocolo de comunicaciones es un estándar de comunicaciones informáticas empleado por diferentes ordenadores (marcas, modelos, sistemas operativos, etc. ) para su entendimiento entre sí. Un protocolo viene a ser como un lenguaje que entienden todos los ordenadores sea cual sea su fabricante o sea cual sea su sistema operativo siempre que estén conectados entre sí. El protocolo de comunicaciones por excelencia empleado por todos los ordenadores conectados a Internet es el protocolo TCP / IP.

### **Dominio propio**

Un dominio propio es un nombre con el que se identifica un sitio Web único en la red Internet. Los nombres de dominio están basados en el servicio DNS. DNS significa Domain Name System (Sistema de Nombres por Dominios). Consiste en una base de datos que relaciona la dirección IP de cada máquina conectada a Internet con su nombre. Todos los ordenadores conectados a Internet han de poseer al menos una dirección IP. Ésta consiste en cuatro números separados por puntos. No pueden haber dos ordenadores con la misma dirección, pues podrían aparecer conflictos y estos no se podrían



comunicar. Los nombres están formados por palabras separadas por puntos. La parte final del nombre es el dominio al cual está conectado el ordenador. Cuando introducimos un nombre en nuestro navegador (Ej. [www.unanleon.edu.ni](http://www.unanleon.edu.ni)), nuestra máquina pregunta al servidor de DNS, cual es el número correspondiente a esa dirección. Cuando el servidor de DNS le responde, nuestra máquina ya puede contactar directamente con el destinatario. El DNS se creó para simplificar el nombre con el que accede a las máquinas.

### **Páginas Web estáticas y dinámicas**

Las páginas Web son textos ASCII escritos en el Lenguaje HTML (Hiptertext Markup Language), que se transfieren entre los servidores de WWW y los navegadores de los clientes mediante el protocolo HTTP (Hiptertext Transfer Protocol). La investigación y desarrollo en el campo de la creación y mantenimiento de páginas Web es uno de los mas dinámicos en el ámbito de Internet, debido a sus consecuencias comerciales y de utilización de la red.

Inicialmente las páginas eran estáticas, en el sentido de que, a efectos de usuario, el único proceso realizado era el de visualización de sus contenidos(escritos en el lenguaje HTML) por parte del explorador del cliente.

Las páginas estáticas se siguen utilizando ampliamente debido a que forman la base necesaria para la presentación de datos en muchos tipos de situaciones. También influye decisivamente la sencillez con que se pueden crear, instalar y mantener.

En el momento en que requiere una interacción mayor entre los usuarios y el sistema que soporta las páginas web, surge la necesidad de reunir y procesar las peticiones del cliente con el fin de ofrecerle informaciones mejor dirigidas, escogidas y elaboradas. Por ejemplo en una universidad que proporcione las notas de los alumnos a través de las páginas web de cada departamento, un estudiante podría conseguir sus calificaciones navegando por cada departamento y asignatura entre gran cantidad de páginas web estáticas relacionadas jerárquicamente. Una alternativa mas elegante y sencilla para el



alumno sería preguntarle sus datos en un formulario y ejecutar alguna aplicación en el servidor que seleccionara sus calificaciones entre las distintas asignaturas en las que se ha matriculado, para por fin, presentarle todas las calificaciones obtenidas a raíz de la petición sencilla y directa realizada por el usuario.

El ejemplo anterior ilustra la conveniencia de dotar de alguna fase de procesamiento al intercambio de información entre los usuarios y el servidor de páginas web, con lo que introducimos en concepto de páginas web activas en el servidor, que los modelos ASP (Active Server Pages) soportan.

La capacidad de procesamiento que sustenta las páginas dinámicas se puede llevar a cabo siguiendo alguno de estos modelos:

- Procesamiento en el equipo del usuario (Páginas activas en el cliente).
- Procesamiento en el equipo donde reside el servidor de web(páginas activas en el servidor).
- Procesamiento mixto (Páginas dinámicas en el cliente + páginas activas en el servidor).

La ventaja principal de las páginas dinámicas en el cliente es la descarga de trabajo, que le proporciona al equipo servidor la posibilidad de traspasar cómputos a los usuarios. Imaginemos una gran cantidad entidad bancaria que proporciona páginas web capaces de calcular amortizaciones de capital y evolución de intereses de préstamo según las condiciones particulares de consulta de cada cliente. ¿Qué tipo de computador sería necesario para soportar decenas de accesos simultáneos en paralelo a otras posibles aplicaciones?. Desde luego supondría una gran descarga traspasar la ejecución del programa a los equipos clientes.

Otra importante ventaja se basa en el ahorro de comunicaciones(ancho de banda) que se puede experimentar en muchas aplicaciones que ejecutan procesos en el cliente, y evitan de esta manera realizar continuos trasposos de información con el servidor de web. Un



ejemplo muy sencillo de esta situación es la creación de una página web “calculadora” que con una sola página activa en el cliente puede funcionar de forma autónoma (una vez cargada) en el equipo del usuario. De otra manera se necesitaría realizar un continuo traspaso de datos (pulsaciones) desde el equipo cliente al servidor, y de resultados (operaciones realizadas) del servidor al cliente. Esta situación refleja uno de los principios existentes en el campo de las comunicaciones, que establece una relación inversa entre capacidad de cómputo disponible y ancho de banda necesario.

A pesar de las innegables ventajas de la distribución del cómputo, existen diversas razones que nos llevan a crear páginas activas en el servidor:

Existe información de naturaleza centralizada. El ejemplo de las calificaciones de los alumnos en una universidad forma parte de los casos de este tipo. Aunque se pueden realizar diseños de páginas web que minimicen el ancho de banda necesario para las consultas o el procesamiento del servidor, los datos y una buena parte de los cálculos no pueden ser eliminados del lugar “Central”. Distribuir la información de un servidor a varios servidores (por ejemplo, uno por departamento) no significa en absoluto traspasar datos y procesamiento al cliente.

Presuponer una adecuada capacidad de cómputo en el equipo del usuario supone en muchas ocasiones una temeridad que puede hacerte perder numerosos clientes que, por ejemplo, han optado por adquirir hardware barato y sencillo cuya interfaz sea un mando a distancia y su propio televisor.

Las páginas activas en el cliente se basan actualmente en tecnologías muy dependientes del explorador y la plataforma del usuario, de esta manera hoy Visual Basic. Los ActiveX no funcionan en todas las plataformas.

Aunque los usuarios cuenten con la plataforma y el explorador adecuados no siempre están dispuestos a introducir componentes ejecutables (ActiveX fundamentalmente) en sus equipos, lo que disminuye considerablemente la potencia de las páginas activas en el cliente.



Cuando el tamaño del programa es grande y el tiempo de computo corto, puede resultar mas rentable traspasar los resultados obtenidos tras ejecutar el código en el servidor, en lugar de enviar el programa al cliente.

### **Páginas dinámicas de cliente**

Son las páginas dinámicas que se procesan en el cliente. En estas páginas toda la carga de procesamiento de los efectos y funcionalidades la soporta el navegador.

El código necesario para crear los efectos y funcionalidades se incluye dentro del mismo archivo HTML y es llamado SCRIPT. Cuando una página HTML contiene scripts de cliente, el navegador se encarga de interpretarlos y ejecutarlos para realizar los efectos y funcionalidades.

Las páginas dinámicas de cliente se escriben en dos lenguajes de programación principalmente: Javascript y Visual Basic Script (VBScript).

Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja, ya que cada navegador tiene sus propias características, incluso cada versión, y lo que puede funcionar en un navegador puede no funcionar en otro.

Como ventaja se puede decir que estas páginas descargan al servidor algunos trabajos, ofrecen respuestas inmediatas a las acciones del usuario y permiten la utilización de algunos recursos de la máquina local.

### **Páginas dinámicas de servidor**

Podemos hablar también de páginas dinámicas del servidor, que son reconocidas, interpretadas y ejecutadas por el propio servidor.



Las páginas del servidor son útiles en muchas ocasiones. Con ellas se puede hacer todo tipo de aplicaciones Web. Desde agendas a foros, sistemas de documentación, estadísticas, juegos, chats, etc. Son especialmente útiles en trabajos que se tiene que acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad los cálculos no se pueden realizar en el ordenador del usuario.

Es importante destacar que las páginas dinámicas de servidor son necesarias porque para hacer la mayoría de las aplicaciones web se debe tener acceso a muchos recursos externos al ordenador del cliente, principalmente bases de datos alojadas en servidores de Internet. Un caso claro es un banco: no tiene ningún sentido que el cliente tenga acceso a toda la base de datos, sólo a la información que le concierne.

Las páginas dinámicas del servidor se suelen escribir en el mismo archivo HTML, mezclado con el código HTML, al igual que ocurría en las páginas del cliente. Cuando una página es solicitada por parte de un cliente, el servidor ejecuta los scripts y se genera una página resultado, que solamente contiene código HTML. Este resultado final es el que se envía al cliente y puede ser interpretado sin lugar a errores ni incompatibilidades, puesto que sólo contiene HTML

Luego es el servidor el que maneja toda la información de las bases de datos y cualquier otro recurso, como imágenes o servidores de correo y luego envía al cliente una página web con los resultados de todas las operaciones.

Para escribir páginas dinámicas de servidor existen varios lenguajes, entre ellos tenemos: Common Gateway Interface (CGI) comúnmente escritos en Perl, Active Server Pages (ASP), Hipertext Preprocesor (PHP), y Java Server Pages (JSP).

Las ventajas de este tipo de programación son que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable.



Como desventajas se puede señalar que será necesario un servidor más potente y con más capacidades que el necesario para las páginas de cliente. Además, estos servidores podrán soportar menos usuarios concurrentes, porque se requerirá más tiempo de procesamiento para cada uno.

## **HTML**

El HyperText Markup Language es el lenguaje para la creación de páginas Web. Con este lenguaje conseguimos dar forma al texto que queremos publicar en la red, pudiéndole añadir fotografías, fondos, viñetas, etc. También con HTML podemos conseguir información del cliente utilizando formularios, lo que resulta de gran utilidad para poder interactuar con el usuario.

## **JavaScript**

JavaScript es un subconjunto de Java creado para trabajar en entornos Web. Al igual que su predecesor, es un lenguaje orientado a objetos. Esto significa que la programación se basa en objetos.

Javascript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.





### **Visual Basic Script(VBScript)**

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de Javascript.

Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows. Tanto su sintaxis como la manera de trabajar están muy inspirados en él. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero.

El modo de funcionamiento de Visual Basic Script para construir efectos especiales en páginas web es muy similar al utilizado en Javascript y los recursos a los que se puede acceder también son los mismos: el navegador.

Como decimos, no debemos utilizar este lenguaje en la mayoría de las ocasiones, aunque un caso donde tendría sentido utilizar Visual Basic Script sería la construcción de una Intranet donde sepamos con toda seguridad que los navegadores que se van a conectar serán siempre Internet Explorer. En este caso, un programador habitual de Visual Basic tendría más facilidades para realizar los scripts utilizando Visual Basic Script en lugar de Javascript.

### **Internet Information Server(IIS)**

Se puede definir *Internet Information Server* como el software estándar que soporta comunicaciones en Internet sobre el sistema operativo Windows. IIS se ha convertido en unos de los soportes de servidores Web y FTP más utilizados. Los servicios básicos que proporciona IIS son: Ghopher, FTP y WWW. El de menor utilización es el Ghopher, cuya principal ventaja reside en la velocidad que proporciona en las búsquedas de información. FTP sigue siendo adecuado para la transferencia rápida de archivos. Sin duda WWW es el servicio más utilizado por su potencia y sencillez de manejo.



## Seguridad con IIS

**Comunicaciones seguras:** *Capa de sockets seguros* (SSL) 3.0 y Seguridad de capa de transporte (TLS) proporcionan una forma segura para intercambiar información entre clientes y servidores. Además, SSL 3.0 y TLS proporcionan al servidor la forma de comprobar quien es el cliente *antes* de que el usuario inicie una sesión en el servidor. En IIS 5.0, los certificados de cliente están expuesto a ISAPI y a las páginas Active Server, de forma que los programadores puedan efectuar el seguimiento de los usuarios a través de los sitios. IIS 5.0 también puede asignar el certificado del cliente a una cuenta de usuario de Windows, de forma que los administradores puedan controlar el acceso a los recursos del sistema basado en el certificado del cliente.

**Criptografía activada por servidor:** La criptografía activada por servidor (SGC, Server-Gated Cryptography) es una extensión de SSL que permite a instituciones financieras con versiones de exportación de IIS utilizar un fuerte cifrado (128-bit). Aunque las capacidades SGC están integradas en IIS 5.0, se requiere un certificado SGC especial para utilizar SGC.

**Asistentes de seguridad:** los asistentes de seguridad simplifican las tareas de administración del servidor.

El asistente para certificados de servidor Web simplifica las tareas de administración de certificados, como la creación de peticiones de certificados y la administración del ciclo de vida del certificado.

El asistente para permisos facilita la configuración del acceso al sitio Web mediante la asignación de directivas de acceso a directorios virtuales y archivos. El asistente para permisos también puede actualizar los permisos de archivo NTFS para reflejar estas directivas de acceso a Web.



El asistente para CTL le ayudará a configurar listas de certificados de confianza (CTL). CTL es una lista de entidades emisoras de certificados (CA) de confianza para un determinado directorio. Las CTL son especialmente útiles para los proveedores de servicios Internet (ISP) que tienen varios sitios Web en el servidor y que necesitan una lista diferente de entidades emisoras de certificados aprobados para cada sitio.

**Restricciones de dominio de Internet e IP:** puede otorgar o denegar accesos Web a equipos individuales, grupos de equipos o dominios enteros.

**Kerberos v5 Compatibilidad con el protocolo de autenticación:** IIS está totalmente integrado con el protocolo de autenticación de Kerberos v5 implementado en Microsoft(r) Windows(r) 2000, que le permite pasar las credenciales de autenticación entre los equipos conectados que ejecutan Windows.

**Fortezza:** El estándar de seguridad del gobierno de EE.UU., habitualmente llamado Fortezza, es compatible con IIS 5.0. Este estándar satisface la arquitectura de seguridad Defense Message System con un mecanismo criptográfico que proporciona la confidencialidad de mensajes, integridad, autenticación y el control de acceso a mensajes, componentes y sistemas. Estas características se pueden implementar con el software del explorador y el servidor y con hardware de tarjeta PCMCIA.

### **Active Serve Pages(ASP)**

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o JScript.

ASP es una tecnología dinámica funcionando principalmente del lado del servidor, un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para



crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

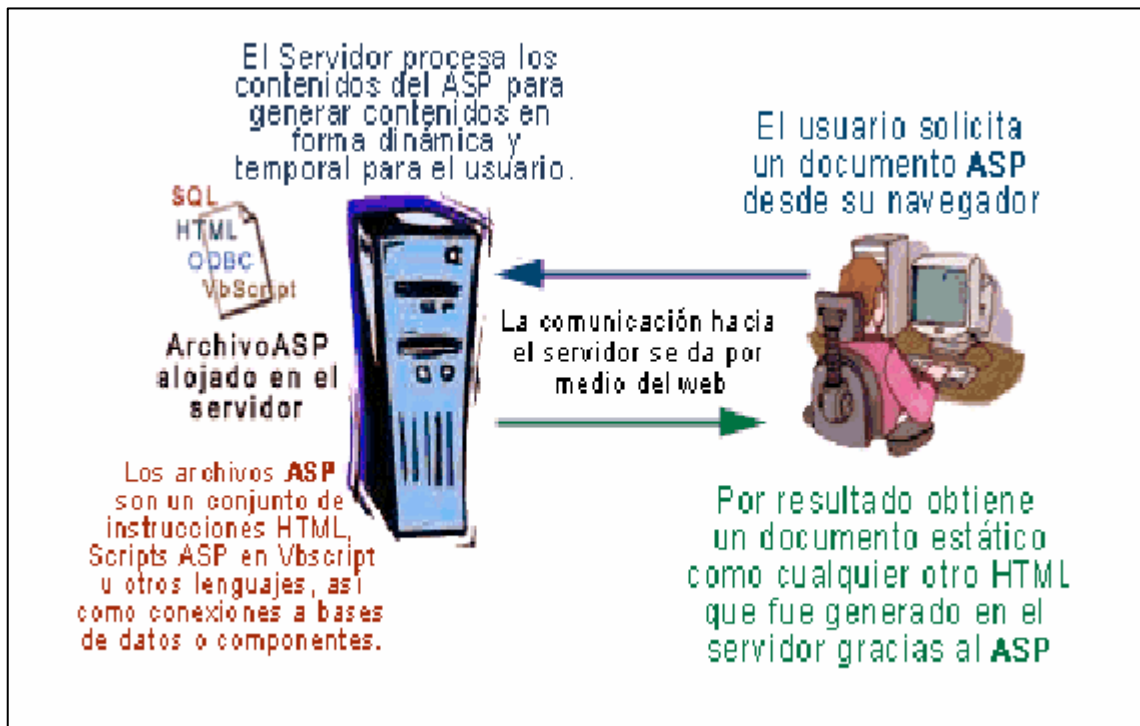
- La sigla ASP significa Active Server Pages.
- Microsoft es el creador.
- Se ejecuta en el servidor donde esta alojada la página web.
- El cliente (navegador del usuario) no nota diferencias con una página normal (HTML).
- Por medio de ASP se puede tener acceso a bases de datos.
- Para su implementación se utiliza lenguaje script como VBScript o JScript.
- Se puede utilizar HTML y ASP en una misma página.
- Es totalmente ampliable gracias a que es compatible con la tecnología ActiveX.

### **Funcionamiento de ASP**

- Un usuario por medio del navegador solicita una página ASP.
- Esta solicitud llega al servidor el cual tiene alojada la página pedida.
- Este servidor procesa la página ASP y devuelve código HTML.
- El usuario visualiza la página en su navegador.

Para insertar instrucciones ASP dentro del código HTML se incluye encerrado entre “<%Código ASP%>”. Estos comandos son los que procesa el servidor antes de enviar la página al cliente(Navegador)

El funcionamiento de ASP se presenta en el siguiente gráfico:



Para el usuario no existe diferencia en ASP y HTML, por que a su navegador siempre llega código HTML puro, el único que requiere un trabajo extra es el servidor, el cual tiene que procesar el código ASP y transformarlo en HTML para su posterior envío al cliente.

El tipo de servidores que emplean este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows NT, aunque también se puede utilizar en un PC con Windows 98 si instalamos un servidor denominado Personal Web Server. Incluso en sistemas Linux podemos utilizar las ASP si instalamos un componente denominado Chilisoft, aunque parece claro que será mejor trabajar sobre el servidor web para el que está pensado: Internet Information Server.

Con las ASP podemos realizar muchos tipos de aplicaciones distintas. Nos permite acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor.



En ASP, todas las páginas web pueden ser diseñadas con editores de HTML, puesto que las instrucciones ejecutables y el código HTML están suficientemente delimitados. Así mismo pueden utilizarse diversos lenguajes para la programación de la funcionalidad de las páginas activas. Entre estos se encuentran Visual Basic Script y Java Script.

Los desarrollos realizados con ASP no necesitan compilaciones que retarden el proceso de producción, y los errores de programación no provocan la caída del servidor. De esta manera, por ejemplo, se hace un uso muy simple de ODBC para el acceso a distintos tipos de bases de datos.

ASP permite compatibilizar la creación de páginas web activas en el cliente y en el servidor, pudiéndose así balancear la carga de proceso y de comunicaciones según los deseos del diseñador. También resulta posible utilizar diversos tipos de lenguaje de programación de scripts en una misma página; aunque esta práctica no resulta recomendable, en algunas ocasiones puede resultar beneficiosa.

Para que un usuario realice una petición de página web, deberá proporcionar en su explorador una dirección que indique un fichero con extensión “.asp”.

Cuando se trabaja con Internet Information Sever(IIS) y Active Server Pages(ASP), el servidor de web analiza las peticiones de página que recibe. Si se encuentra con una solicitud de páginas de extensión “.asp” en lugar de “.html”, entonces se apoya en la aplicación ISAPI que sirve de soporte de ejecución de las páginas ASP.

La aplicación ISAPI de ASP diferencia las líneas HTML de las instrucciones que dan la funcionalidad dinámica a las páginas activas. Cuando determina el lenguaje en el que se encuentran los programas (SCRIPTS), da paso al motor de ejecución de scripts adecuado (Visual Basic Script, Java Script, etc.). Los motores de ejecución de scripts se encargan de realizar el análisis sintáctico y la compilación de las instrucciones ejecutables. Existe una memoria caché de páginas recientemente procesadas que permite aumentar las



prestaciones ASP, evitando repetir los procesos de separación de instrucciones, análisis sintáctico y compilación de las páginas más utilizadas.

### **Acceso a Bases de Datos con ASP**

Una de las mejores características que nos ofrece ASP es la facilidad en el uso de Base de Datos. Esto se realiza utilizando Active Data Objects (ADO), componentes Active, los cuales ofrecen un acceso uniforme y sencillo a una Base de Datos de cualquier tipo, ya Access, SQL Server, Oracle, etc.

### **Administrador de orígenes de datos(ODBC)**

La aplicación de administración de orígenes de datos es una utilidad general de Windows. Resulta necesaria para poder acceder de forma sencilla a distintas bases de datos, por lo que su funcionalidad es importante en la programación de páginas activas en el servidor.

ODBC es una interfaz que permite a las aplicaciones acceder a distintos tipos de bases de datos, para ello se necesitan los diferentes controladores de ODBC con los que se obtiene información de los orígenes de datos.

### **Creación de DSN**

Para acceder a nuestra base de datos a través de una interfaz Web/DBMS utilizando ASP es necesario tener un vínculo con nuestra base de datos, esto se realiza a través de los controladores. Los controladores de bases de datos(programas que pasan información desde la aplicación Web a una base de datos) utilizan un nombre de origen de datos(DSN) para encontrar e identificar una base de datos ODBC en particular.

Normalmente, el DSN contiene información de configuración de la base de datos, seguridad de usuarios y ubicación, y puede tener la forma de una entrada en el registro del sistema de Windows o de un archivo de texto.



Con ODBC se puede elegir el tipo de DNS a crear: Usuario, Sistemas o Archivo.

Los DNS de Usuario y de Sistemas residen en el registro del sistema de Windows. Los DNS de Sistema permiten que todos los usuarios que han iniciado una sesión en un servidor concreto tengan acceso a una base de datos, mientras que los DNS de Usuario limitan la conectividad con la base de datos a los usuarios que dispongan de las credenciales de seguridad apropiadas. Los DNS de Archivos, que tiene la forma de archivos de texto, proporcionan acceso a varios usuarios y son fácilmente transferibles entre servidor y otro mediante la copia de los archivos DNS.

Como el DNS de sistemas permite que el usuario anónimo que crea IIS al instalarse en nuestro servidor puedan acceder a nuestra base de datos, utilizamos un DNS de Sistema para acceder a nuestra base de datos.

### **Pasos para crear un DNS de Sistema**

Primeramente debe abrirse el **Panel de Control**. Luego hacer doble clic en la carpeta **Herramientas Administrativas**, hacer doble clic en el icono **ODBC** y seleccionar la pestaña **DNS de Sistema**. Hacer clic en Agregar, elegir el controlador de la base de datos (en nuestro caso utilizamos **Microsoft Access Driver (\*.mdb)**) y hacer clic en **Finalizar**. Posteriormente escribir el nombre del DNS, y finalmente se selecciona la base de datos y hacer clic en el botón **Aceptar**.

### **Seguridad en el Comercio Electrónico**

Uno de los principales factores que han frenado el crecimiento masivo del comercio electrónico a través de Internet es la preocupación de los usuarios por la seguridad en las transacciones comerciales, tanto en lo que se refiere a la información que deben proporcionar para efectuar el pago como a la autenticidad en la identidad de la empresa a la que la proporcionarían sus datos.





Si bien es muy simple afirmar que esta preocupación es irracional y que utilizar la tarjeta de crédito para efectuar compras en línea es tan seguro como entregarla en cualquier establecimiento comercial para el pago de una compra o consumo tradicional, existen sistemas, protocolos y procedimientos que permiten obtener un nivel de seguridad suficiente para que enviar por Internet el número de la tarjeta sea mucho más confiable que dictarlo por teléfono, dejar firmado en blanco un pagaré bancario que garantice el pago de servicios antes de ocuparlos o entregar a un mesero la tarjeta para liquidar un consumo de bebidas y alimentos.

La seguridad en la transmisión de datos por Internet se basa en el hecho de poder encriptar los mensajes que se envían en la red entre un servidor y un cliente y que solo ellos puedan descifrar los contenidos a través de una clave común conocida sólo por los dos.

Los sistemas de comercio electrónico seguro más utilizados actualmente están basados en SSL (Secure Socket Layer), protocolo desarrollado por Netscape que encripta los datos, codificándolos antes de enviarlos y decodificándolos al llegar a su destino. Al conectarse a un web site que utiliza SSL, si el browser está habilitado para soportarlo, se establece un canal de comunicación seguro a través del cual toda la información se transmitirá de manera codificada: aunque alguien interceptara la transmisión necesitaría tener la clave que se requiere para descifrarla, pero ésta únicamente está al alcance del browser y del servidor que se están comunicando. En la actualidad este protocolo está soportado por los navegadores de mayor uso alrededor del mundo: Netscape y Explorer, en sus versiones 3 y 4.

### **Secure Socket Layer(SSL)**

SSL(Secure Socket Layer): Fue diseñado y propuesto en 1994 por Netscape Communications Corporation junto con su primera versión del navegador. Sin embargo fue hasta su tercera versión conocida como SSL v3.0 que supero los problemas de seguridad y limitaciones de sus predecesores. En su estado actual proporciona cifrado de datos,



autenticación de servidores integridad de mensaje y opcionalmente autenticación de cliente para conexiones TCP/IP.

SSL v3.0 goza de gran popularidad por lo que se encuentra ampliamente extendido en Internet. Viene soportado por los principales navegadores del mercado Netscape Navigator 3.0 o superior y Internet Explorer 3.0 o superior.

SSL es la solución básica que permite la encriptación de todas las transacciones de datos entre el cliente y el servidor. Los números de tarjetas de crédito, por ejemplo, se transmiten a través de la red en código. El pago SSL es el estándar en el 95% de las transacciones en Internet. No es diferente de una transacción en una tienda convencional.

El logro más importante del protocolo es proporcionar privacidad y confianza entre las partes que se comunican. El protocolo permite al servidor y al usuario autenticar y negociar entre ambas partes un algoritmo de encriptación y llaves criptográficas, antes de que se transmita o reciba cualquier información. Es decir, una vez en línea y habiéndose decidido hacer la operación, el navegador se conecta a un servidor "Seguro" de Internet, el cual ha sido autenticado o validado por una tercera empresa certificada a tal efecto. El servidor seguro usa su llave privada y genera una sesión segura de conexión con el usuario; el navegador decodifica la llave enviada por el servidor y si lo descifra correctamente, esto quiere decir que solo ese servidor pudo haber enviado dicha llave. Al haberse establecido lo anterior, se abre un canal o conexión segura, y toda la información que se cruce entre las partes estará encriptada o protegida.

El rasgo que distingue a SSL de otros protocolos para comunicaciones seguras, como el hoy prácticamente extinto S-HTTP, es que se ubica en la pila OSI entre los niveles de transporte (TCP/IP) y de aplicación (donde se encuentran los conocidos protocolos HTTP para Web, FTP para transferencia de ficheros, SMTP para correo electrónico, Telnet para conexión a máquinas remotas, etc.). Gracias a esta característica, SSL resulta muy flexible, ya que puede servir para securizar potencialmente otros servicios además de



HTTP para Web, sin más que hacer pequeñas modificaciones en el programa que utilice el protocolo de transporte de datos TCP.

S.S.L. proporciona sus servicios de seguridad en base a dos tecnologías de cifrado distintas:

**Criptografía de Clave Pública (asimétrica).**- Para la autenticación y cifrado de la clave de sesión que se utiliza para cifrar datos que vienen y van al servidor una vez establecido el canal seguro.

**Criptografía de Clave Secreta (simétrica).**- Para el intercambio de datos entre el servidor y el cliente.

Este protocolo sigue las siguientes fases:

- 1) **La fase Hola**, usada para ponerse de acuerdo sobre el conjunto de algoritmos para garantizar la confidencialidad e integridad y para la autenticación mutua. El navegador le informa al servidor de los algoritmos que posee disponibles. Normalmente, se utilizarán los más fuertes que se puedan acordar entre las dos partes. En función de las posibilidades criptográficas del navegador, el servidor elegirá un conjunto u otro de algoritmos con una cierta longitud de claves.
- 2) **La fase de autenticación**, en la que el servidor envía al navegador su certificado x.509v3 que contiene su clave pública y solicita a su vez al cliente su certificado X.509v3 (sólo si la aplicación exige la autenticación de cliente).
- 3) **La fase de producción de clave de sesión**, en la que el cliente envía al servidor una clave maestra a partir de la cual se generará la clave de sesión para cifrar los datos intercambiados posteriormente mediante el algoritmo de cifrado simétrico acordado en la fase 1. El navegador envía cifrada esta clave maestra usando la clave pública del servidor que extrajo de su certificado en la fase 2. Más adelante,



ambos generarán idénticas claves de sesión a partir de la clave maestra generada por el navegador.

- 4) **La fase Fin**, en la que se verifica mutuamente la autenticidad de las partes implicadas y que el canal seguro ha sido correctamente establecido. Una vez finalizada, ya se puede comenzar la sesión segura.

De ahí en adelante, durante la sesión segura abierta, SSL proporciona un canal de comunicaciones seguro entre los servidores Web y los clientes (los navegadores) a través del cual se intercambiará cifrada la información relevante, como el URL y los contenidos del documento solicitado, los contenidos de cualquier formulario enviado desde el navegador, las cookies enviadas desde el navegador al servidor y viceversa y los contenidos de las cabeceras HTTP.

#### **VENTAJAS CON SEGURIDAD S.S.L.**

- SSL v3.0 goza de gran popularidad y se encuentra ampliamente extendido en Internet, ya que viene soportado por los dos principales navegadores del mercado, Netscape Navigator 3.0 ó superior así como por Internet Explorer 3.0 ó superior.
- SSL proporciona un canal de comunicaciones seguro entre los servidores web y los clientes (los navegadores), pero su uso no se limita a la transmisión de páginas web. Al encontrarse entre los niveles de transporte y de aplicación, potencialmente SSL puede servir para securizar otros servicios, como FTP, correo, telnet, etc.
- El usuario no necesita realizar ninguna acción especial para invocar el protocolo SSL, basta con seguir un enlace o abrir una página cuya dirección empieza por https://. El navegador se encarga del resto.



## VI. MATERIAL Y MÉTODO

### Materiales

*Hardware:* para llevar a cabo esta aplicación utilizaremos una computadora con las siguientes características.

- IBM
- Celeron
- Velocidad 366 Mhz.
- 5 GB de Disco Duro.
- 64 MB de RAM

*Software:* el software en el que se desarrollará la aplicación será:

- Windows 2000 Professional.
- HTML
- JavaScript
- Visual BasicScript
- Gestor de bases de datos Access 2000.
- Se utilizará la tecnología ASP para acceder a la Base de Datos desde Internet.
- Internet Information Server 5.0
- Procesador de texto Microsoft Word 2000.
- Microsoft FrontPage 2000.

Para publicar la Página Web en Internet se implementará un servidor con Windows 2000 Server.



## Diseño metodológico

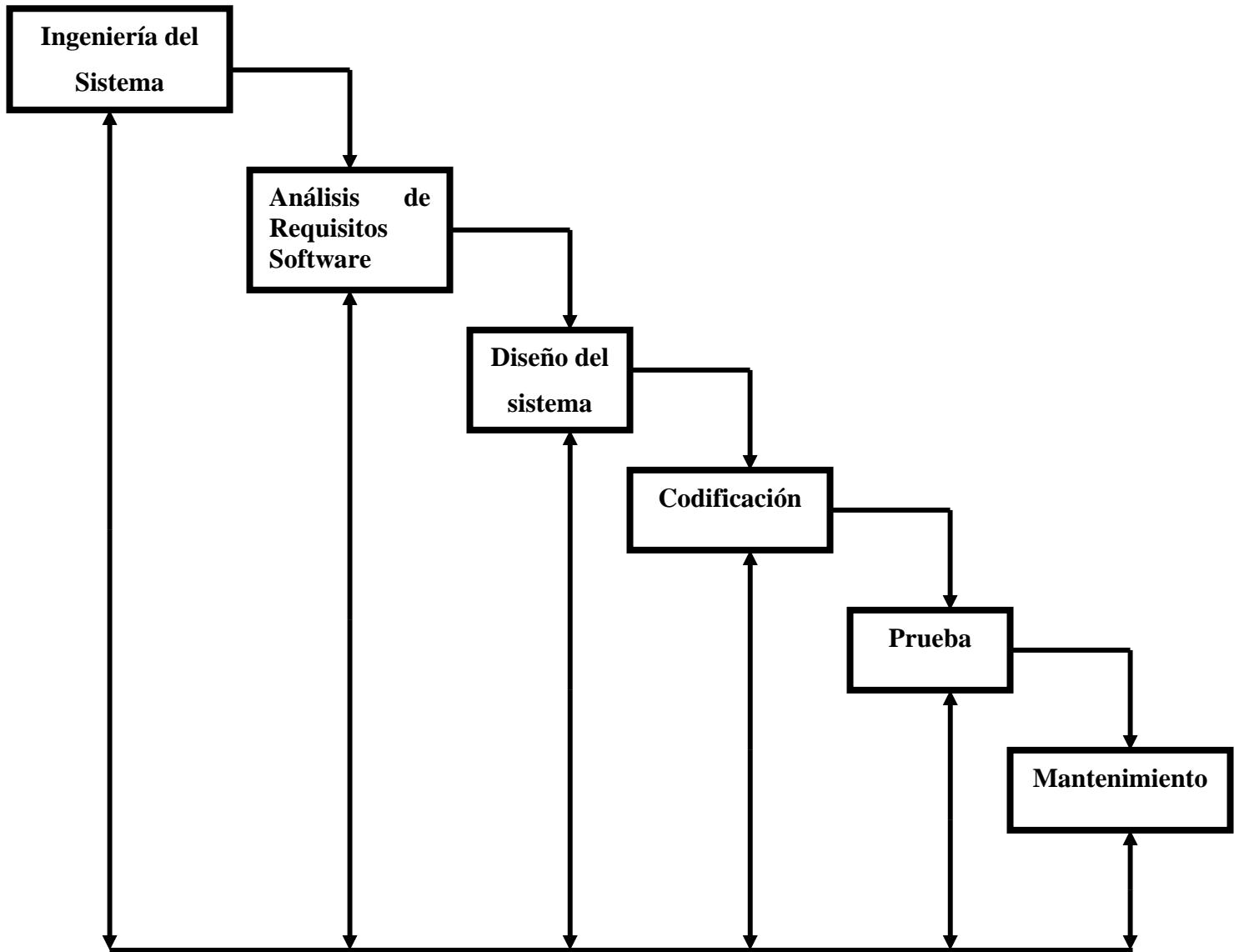
Para llevar a cabo el desarrollo de nuestro trabajo utilizaremos el *Ciclo de Vida Clásico del Software* o *Modelo de Cascada*, ya que este modelo nos permite realizar cada una de las etapas del software de una manera independiente.

Las etapas a seguir para el desarrollo de esta página web son las siguientes:

- *Análisis del Sistema*
- *Diseño*
- *Codificación*
- *Prueba*
- *Mantenimiento*



Gráficamente el modelo de ciclo de vida clásico se representa de la siguiente manera:





## VII. ANÁLISIS

### Especificación de Requisitos Software

#### 1. Introducción

##### 1.1 Propósito

Definir detalladamente los requisitos del software para el Comercio Electrónico, el cual consiste en la creación de una página web dinámica, por medio de la cual los usuarios puedan realizar compras de los productos a través de Internet. La funcionalidad de esta aplicación se centra en proporcionar a cada usuario la posibilidad de navegar por varias páginas que ofrecen productos, de tal manera que se pueda ir seleccionando producto por producto e ir agregándolos a un “Carrito de Compra”. La aplicación se encargará de llevar el control de la cantidad de productos, así como la cantidad de dinero que se gastará en la compra.

##### 1.2 Alcance

El nombre de la aplicación será: **COMERCIO ELECTRÓNICO**

Las funciones que realizará la aplicación web son las siguientes:

- a. Presentar una página web principal, con la presentación de los productos por categorías, imágenes de algunos productos, con los respectivos precios, también se presentará un logotipo alusivo a la empresa que adquiera la aplicación.
- b. Mostrar los productos de cada categoría.
- c. Proporcionar una descripción de cada producto.





- d. Agregar un producto al “Carrito de Compras”.
- e. Borrar productos del “Carrito de Compras”.
- f. Buscar un determinado producto.
- g. Capturar los datos del cliente.
- h. Enviar los datos del cliente a una Base de Datos.

### 1.3 Definiciones acrónimos y abreviaturas.

**Carrito de Compras:** es el lugar donde se almacenarán todos los productos que el cliente desea comprar.

**Cliente:** son todos los usuarios que ingresan a la página web.

**Enlaces:** son marcas con las cuales podemos movernos por la página web simplemente dando clic con el ratón.

**Producto:** se refiere a todos los productos que se ofertarán o estarán a la venta en la página web.

**Categoría:** Es la agrupación o clasificación de los productos que se ofrecen en la página web.

**Navegador:** Es la aplicación que se utiliza para visualizar las páginas web.



## 1.4 Visión General.

Se realizará una descripción general del sistema para posteriormente estudiar y detallar cada uno de los requisitos específicos de forma individual.

## 2. Descripción General.

### 2.1 Relaciones del producto

La aplicación será instalada en un servidor en la Web, para que cada cliente pueda acceder a ella a través de Internet desde cualquier computador que tenga acceso a Internet.

El equipo donde se desarrollará la aplicación tiene las siguientes características:

IBM

Procesador Celeron 266 Mhz.

5 GB de disco Duro

64 Mb de RAM.

### 2.2 Funciones del producto.

- a. Cuando se de clic en uno de los enlaces de la página Web inmediatamente se presentará el contenido de la página relacionado con el enlace.
- b. Al hacer clic en el lugar “Enviar al carrito”, inmediatamente se presentará el carrito de compra con todos los productos agregados y el total a pagar.
- c. Cuando se valla a realizar la compra se procederá a realizar la facturación en la Web, aquí se procederá a introducir los datos del cliente necesarios para realizar la compra.



- d. Cuando el cliente borre un producto del carrito de compra inmediatamente se actualizará el carrito de compra con los productos que contenga este.
- e. Cuando el usuario escoja la opción de enviar, los datos serán enviados a una Base de Datos.

### **2.3 Características del usuario.**

Los usuarios de la aplicación serán todas las personas que ingresen a la página Web y por lo tanto algunas personas conocerán como comprar en Internet y otras no, por tanto se incluirá una ayuda en la aplicación donde se detallará como usar la aplicación o como debe comprar en línea.

### **2.4 Restricciones Generales.**

El software en el que se desarrollará la aplicación será:

Windows 2000

HTML

JavaScript

VisulBasicScrpit

Microsoft Access 2000.

Se utilizará la tecnología ASP.



### **3. Requisitos Específicos.**

#### **3.1 Requisitos Funcionales**

##### **3.1.1 Presentar la Página Web.**

###### **3.1.1.1 Introducción.**

Aquí se presentará la página principal de la aplicación. La página se presentará cuando el usuario escriba la dirección de la página en el Navegador.

Ej.: [www.MiTienda.com](http://www.MiTienda.com)

###### **3.1.1.2 Entradas.**

Dirección de la página Web.

###### **3.1.1.3 Proceso.**

Se mostrará la página principal de la aplicación donde el cliente podrá navegar en ella para ver los productos y realizar la compra.

###### **3.1.1.4 Salida.**

Se presentará una página con todas las categorías de los productos ofertados, también se mostrarán imágenes de algunos productos, además opciones, como la ayuda de la aplicación y logotipo de la empresa.



### **3.1.2 Mostrar los productos de cada categoría.(Por medio de enlaces).**

#### **3.1.2.1 Introducción.**

Esta función tiene como objetivo que el cliente pueda hacer clic en una categoría de producto y la aplicación muestre en la página Web todos los producto de esa categoría.

#### **3.1.2.2 Entradas.**

Nombre de la categoría.

#### **3.1.2.3 Proceso.**

Se presentará un enlace en cada categoría de producto donde el cliente tendrá la opción de hacer clic sobre uno de esos enlaces para poder ver todos lo productos de la categoría seleccionada.

#### **3.1.2.4 Salida.**

Se mostrará una página Web con todos lo productos de la categoría seleccionada, además de toda las opciones necesaria para poder comprar un producto de la categoría.

### **3.1.3 Proporcionar una descripción de cada producto.**

#### **3.1.3.1 Introducción.**

Esta función tiene como objetivo proporcionar al cliente una descripción completa del producto que este desee.

**3.1.3.2 Entrada.**

Nombre del producto.

**3.1.3.3 Proceso.**

Se presentará en la página Web un enlace con el nombre del producto donde el cliente solamente hará clic con el ratón sobre el enlace.

**3.1.2.4 Salida.**

Una vez que el cliente haga clic sobre el nombre de algún producto se mostrará una página Web con una descripción del producto.

**3.1.4 Agregar un producto al Carrito de Compra.****3.1.4.1 Introducción.**

Esta función permitirá al usuario seleccionar cualquiera de los productos presentados en la página Web y agregarlo al carrito de compra mostrando detalles relacionados con el producto agregado.

**3.1.4.2 Entradas.**

Nombre producto.

**3.1.4.3 Proceso.**

Se presentará en la página Web una opción con el nombre de “Enviar al Carrito” donde el cliente podrá hacer un clic e inmediatamente se agregará ese producto al



carrito de compra. Aquí también se realizarán las operaciones para calcular el total a pagar.

#### **3.1.4.4 Salidas.**

Se presentará una tabla llamada carrito de compra con todos los productos agregados y los detalles específicos relacionados con cada producto tales como: cantidad, nombre del producto, precio unitario y finalmente el precio total a pagar por la compra de los productos.

### **3.1.5 Borrar un producto del Carrito de Compras.**

#### **3.1.5.1 Introducción.**

Esta función permitirá al cliente borrar cualquier producto que este contenido en el carrito de compra que por cualquier razón ya no desee comprarlo.

#### **3.1.5.2 Entradas.**

Nombre del producto.

#### **3.1.5.3 Proceso.**

Se mostrará en la página Web una opción llamada “Borrar” donde el cliente podrá hacer clic y de esta manera se elimina ese producto del carrito de compra.

#### **3.1.5.4 Salida.**

La aplicación Web mostrará nuevamente el carrito de compra pero sin el producto que se ha borrado.



### **3.1.6 Buscar un determinado producto.**

#### **3.1.6.1 Introducción**

Esta función tiene como objetivo que el usuario pueda realizar una búsqueda de un producto para facilitar la compra, de esta manera el usuario puede encontrar un producto sin conocer en que categoría se encuentra el producto.

#### **3.1.6.2 Entradas**

Nombre del producto.

#### **3.1.6.1 Proceso**

Se presentará una página en la que el cliente tendrá la opción de introducir el nombre del producto que desea buscar.

#### **3.1.6.1 Salida**

Se mostrará una página con el producto que el usuario busca, si no se encuentra el producto se enviará un mensaje comunicándole al cliente que no se encontró el producto.

### **3.1.7 Capturar datos del cliente.**

#### **3.1.7.1 Introducción.**

Esta función permitirá capturar todos los datos del cliente necesarios para poder realizar el pedido de los productos, o sea para poder comprar.





### **3.1.7.2 Entrada.**

Los datos de entrada serán:

Nombre

Apellido

Cédula

Departamento

Municipio

Dirección

Teléfono

E-mail

Forma de Pago

### **3.1.7.3 Proceso**

Se mostrará en la página Web un formulario en el que el cliente podrá introducir los datos.

### **3.1.7.4 Salida**

Se presentarán todos los datos introducidos por el cliente, a demás de los productos que el cliente ha seleccionado y se mostrará una opción para que el cliente pueda enviar los datos para realizar el pedido.

## **3.1.8 Enviar los datos**

### **3.1.8.1 Introducción.**

Esta función permitirá enviar los datos introducidos por el cliente hacia una Base de Datos, para así realizar el pedido de los productos seleccionados.



### **3.1.8.2 Entradas.**

Los datos necesarios para que esta función se lleve a cabo son los mismos de la función anterior.

### **3.1.8.3 Proceso.**

Se mostrará en la página Web una opción que diga “Enviar” donde el cliente simplemente hará clic con el ratón y de esta manera se enviarán los datos.

### **3.1.8.4 Salida**

Los datos se almacenarán en una Base de Datos, y también se enviará por pantalla un mensaje confirmando que los datos han sido enviados.

## **3.2 Requisitos de funcionamiento.**

**3.2.1 Requisitos Estáticos:** el número de terminales que estarán conectados a la aplicación, es decir el número de clientes que estarán accediendo a la página Web simultáneamente será indeterminado.

**3.2.2 Requisitos Dinámicos:** El tiempo de acceso dependerá de la velocidad del procesador del P.C de donde se está accediendo a la página Web.

## **3.3 Atributos**

### **3.3.1 Seguridad**

La información presentada en la página Web lógicamente no podrá ser modificada por ningún usuario (Únicamente la necesaria para realizar la compra), solamente el diseñador de la página Web tendrá la opción de entrar directamente al servidor



donde está la página Web para que este pueda modificar o actualizar la página Web.

### **3.3.2 Mantenimiento**

La modificación de uno de los requisitos que se han establecido en este documento deberá ser reflejado en este mismo documento.

### **3.3.3 Ayuda en línea.**

Todos los procesos o las funciones que realiza esta aplicación contará con una ayuda en línea para facilitarle al cliente la navegación por la página Web.

## **3.4 Otros Requisitos.**

### **3.4.1 Base de datos.**

Para almacenar los datos de los productos se hará uso de una base de datos.

### **3.4.2 Operaciones.**

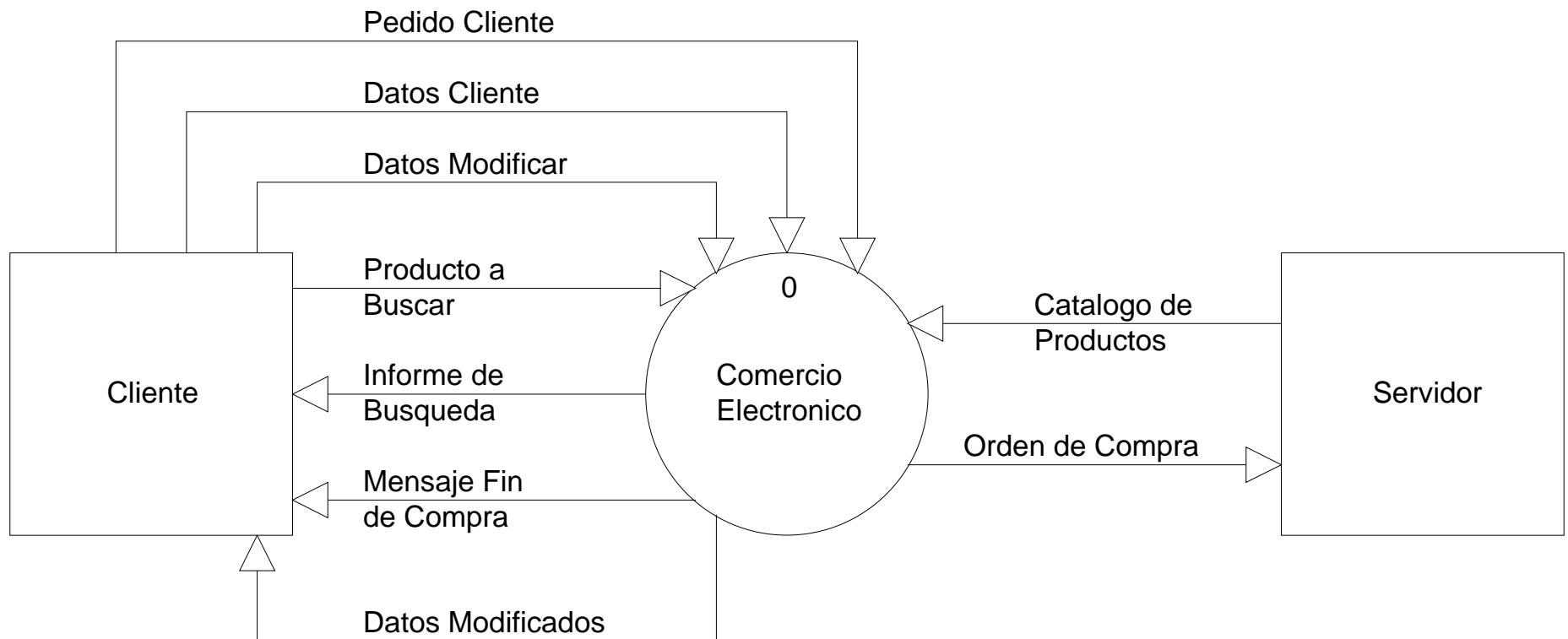
Las operaciones en la base de datos se realizarán solo con autorización, a través del uso de contraseñas para evitar la pérdida o modificación innecesaria de los datos.

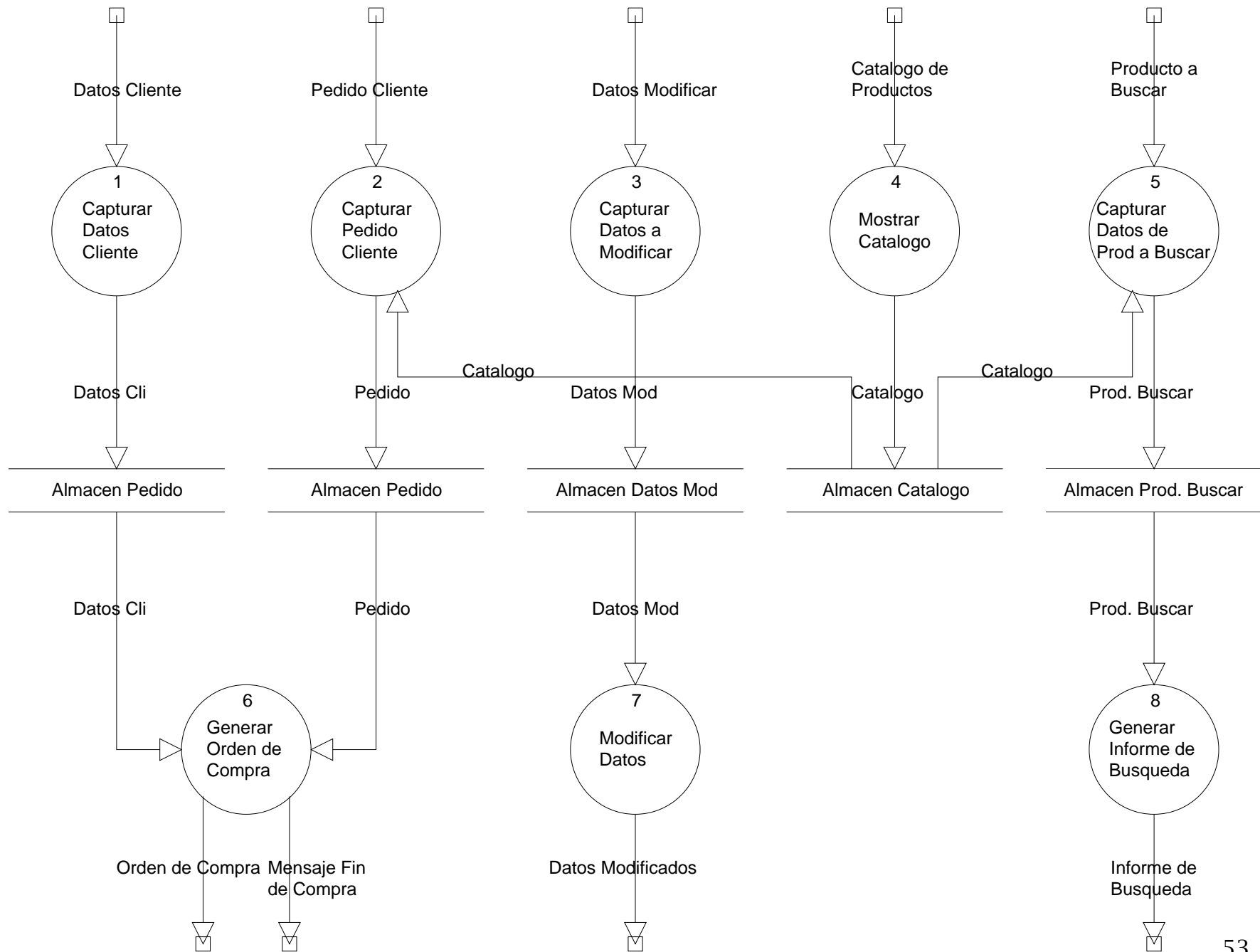


# DIAGRAMA DE FLUJO DE DATOS

## Diagrama de Contexto

### Nivel 0







### VIII. DISEÑO

#### Estructura de las tablas

Las siguientes tablas contienen la estructura de los datos del cliente y el pedido del cliente los cuales serán almacenados cuando el cliente envíe los datos.

Nombre del campo	Tipo de datos	Descripción
NUMERO	Texto	NUMERO QUE IDENTIFICA A UN UNICO CLIENTE
NOMBRE	Texto	NOMBRE DEL CLIENTE
APELLIDO	Texto	APELLIDO DEL CLIENTE
CEDULA	Texto	IDENTIFICACION DEL CLIENTE
DEPARTAMENTO	Texto	DEPARTAMENTO DONDE SE ENVIARAN LOS PRODUCTOS
MUNICIPIO	Texto	MUNICIPIO DONDE SE ENVIARAN LOS PRODUCTOS
DIRECCION	Texto	DIRECCION DEL CLIENTE
TELEFONO	Texto	NUMERO TELEFONICO DEL CLIENTE
MAIL	Texto	CORREO ELECTRONICO DEL CLIENTE
PAGO	Texto	FORMA DE PAGO
TARJETA	Texto	NUMERO DE TARJETA DE CREDITO

Nombre del campo	Tipo de datos	Descripción
NUMERO	Texto	NUMERO QUE IDENTIFICA AL CLIENTE
CLIENTE	Texto	NOMBRE DEL CLIENTE
PRODUCTO	Texto	NOMBRE DEL PRODUCTO QUE PIDE EL CLIENTE
CANTIDAD	Texto	CANTIDAD DEL PRODUCTO SOLICITADA POR EL CLIENTE
PRECIO	Texto	PRECIO DEL PRODUCTO
TOTAL	Texto	TOTAL A PAGAR POR LA COMPRA



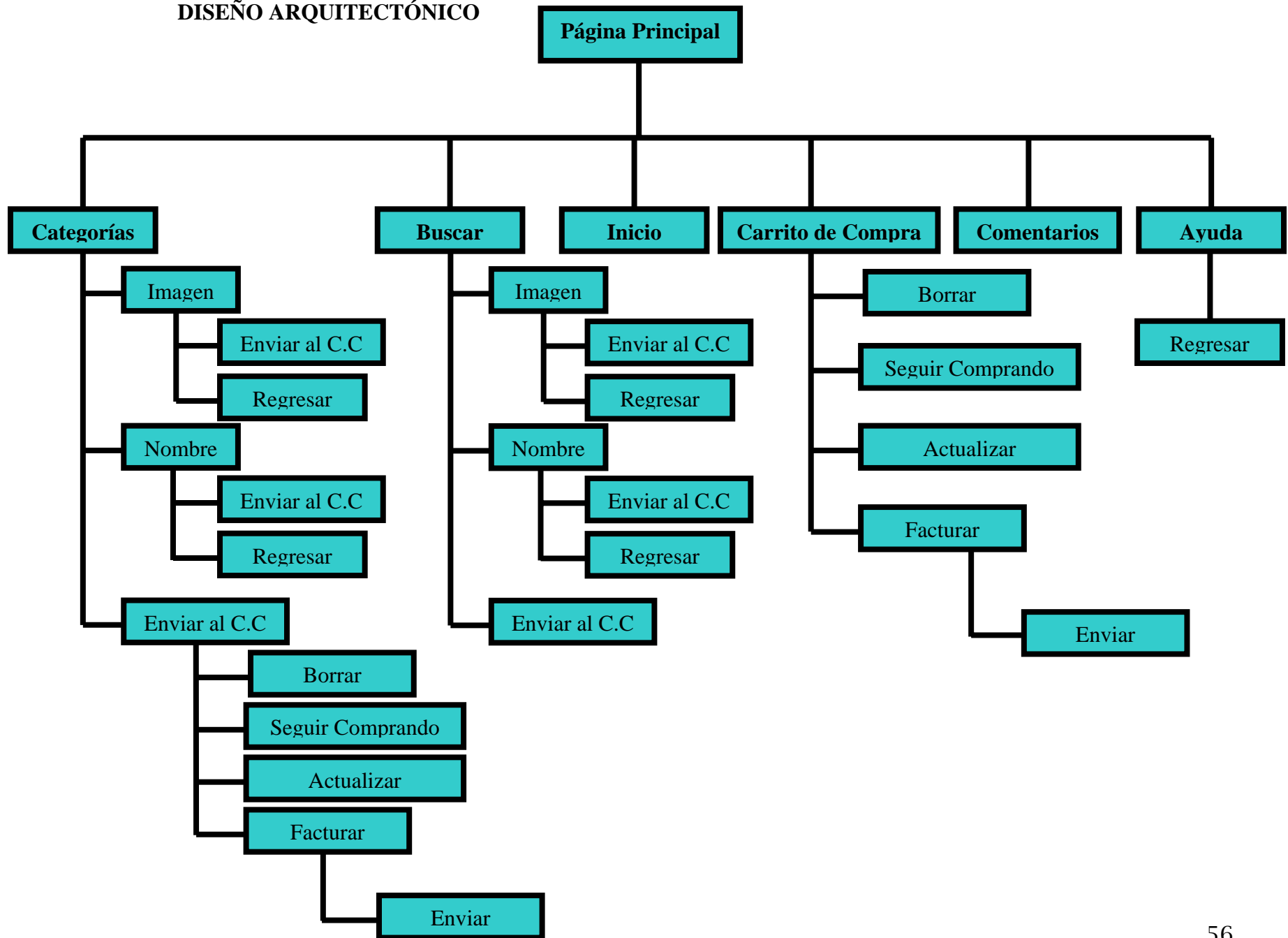
Las siguientes tablas contienen información de los productos que serán mostrados al cliente a través de la página Web.

Nombre del campo	Tipo de datos	Descripción
CodGrupo	Texto	Clave Primaria de la tabla. Código de una Categoría de Producto.
Descripcion	Texto	Nombre del Grupo o Categoría.
FotoGrupo	Texto	Fografía del Grupo o Categoría.
Mostrar	Sí/No	Permitirá mostrar o no una Categoría en la Página Web.

Nombre del campo	Tipo de datos	Descripción
ID_PRODUCTO	Texto	Clave Primaria de la tabla. Código del productos
NOMBRE	Texto	Nombre del producto
FOTO	Texto	Fotografía del producto
PRECIO	Numérico	Precio unitario del producto
CATEGORIA	Texto	Grupo o Categoría a la que pertenece el producto
DESCRIPCION	Texto	Descripcion mas detallada del producto
PALABRAS_CLAVES	Texto	Palabras que se utilizaran en la busqueda de un determinado producto
MOSTRAR_INICIAL	Sí/No	Permitira decidir si mostrar o no el producto en la pagina inicial



DISEÑO ARQUITECTÓNICO







## DISEÑO DE INTERFAZ

Esta es la página principal de la aplicación, aquí se muestran las opciones necesarias para que el usuario pueda interactuar con la página y así poder buscar y seleccionar los productos que desee comprar.



A continuación se detalla el funcionamiento de cada una de las opciones de la página principal:



**PRINCIPAL:** permite al usuario retornar a la página principal de la aplicación.

**CARRITO:** esta opción le permite al usuario ver el carrito de compra para verificar los productos que ha seleccionado.

**CONTACTENOS:** permite al usuario enviar un correo electrónico a la empresa propietaria de la tienda virtual.

**AYUDA:** proporciona al usuario la ayuda necesaria para que este pueda utilizar de forma fácil y correcta las opciones de la aplicación.

**BUSCAR:** esta opción le permite al usuario realizar una búsqueda de un determinado producto.

**Mas Productos:** por medio de esta opción el usuario podrá visualizar otra lista de productos, los que se encuentran agrupados por categorías.

**Enviar al Carrito:** permite que el usuario pueda agregar un determinado producto al carrito de compra.



Al hacer clic en la opción *Enviar al Carrito* se mostrará la siguiente página:

**E-Commerce.com**

Principal Carrito Contáctenos Ayuda

# E-Commerce

Buscar

PRODUCTO	CANTIDAD	PRECIO	SUB-TOTAL	BORRAR
Cannon Bubble-Jet BJ-10e	<input type="text" value="1"/>	1500.5	1500.5	Uno Todos
Modem Interno	<input type="text" value="2"/>	1000	2000	Uno Todos
TOTAL			C\$ 3500.5	

Seguir Comprando Facturar

Actualizar

Borrar Un Producto Local intranet

Start Bienvenidos a la tien... Documento1 - Microsoft W... EN 04:15 p.m.

Las opciones de esta página son las siguientes:

La caja de texto “Cantidad” permite que el usuario pueda modificar la cantidad de productos que desee comprar; posteriormente deberá hacer clic en la opción Actualizar, esta opción permite calcular la cantidad a pagar de cada producto y el total a pagar por toda la compra.

Para borrar un producto del Carrito de Compra se presentan dos opciones:

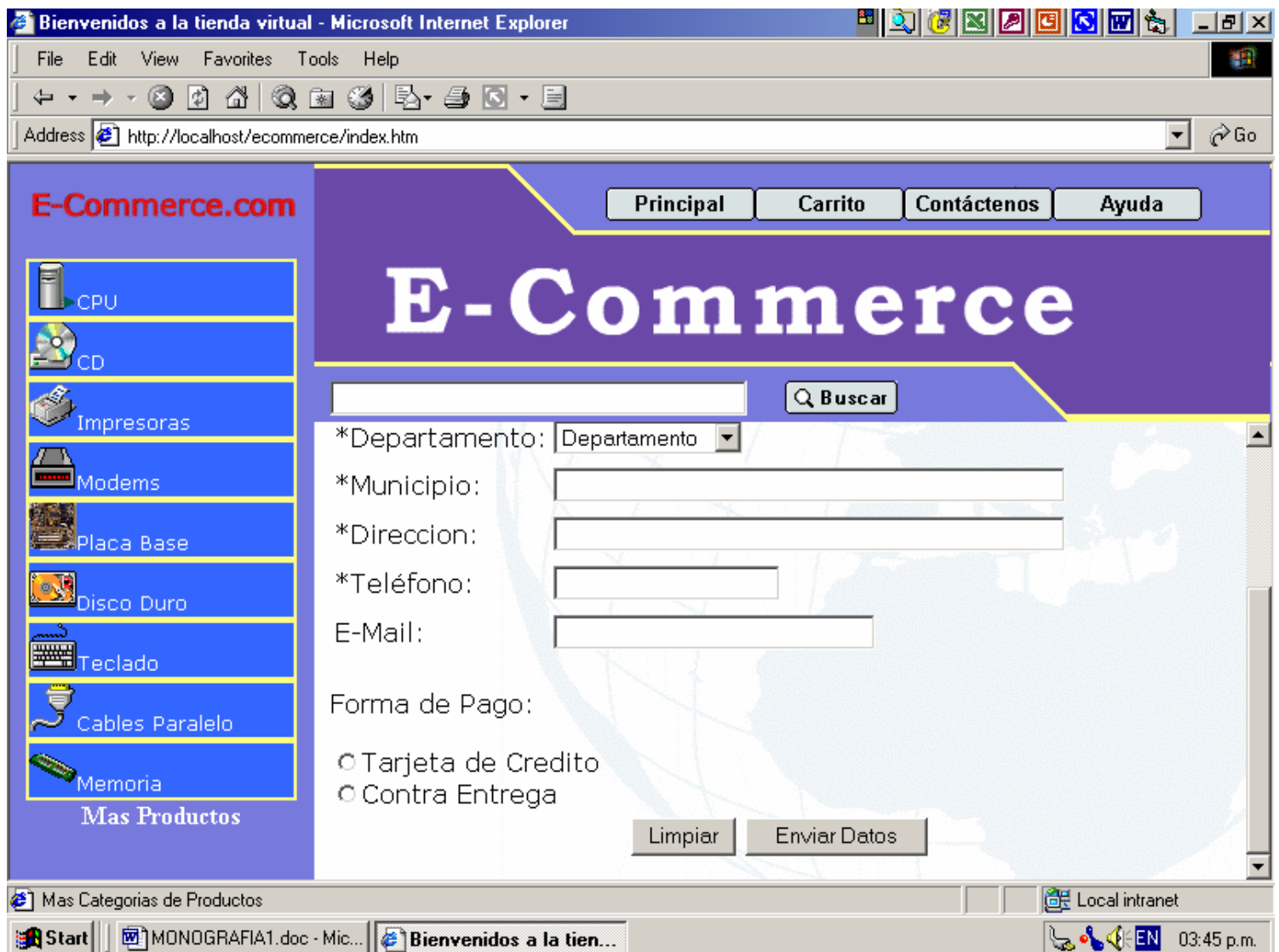


**Uno:** permite al usuario borrar solamente una unidad de un determinado producto.

**Todos:** permite borrar todas las unidades de un determinado producto.

**Seguir Comprando:** permite al usuario retornar a la página principal.

**Facturar:** al hacer clic en la opción facturar se mostrará el siguiente formulario, en el cual el usuario deberá introducir los datos personales:



Este formulario presenta dos opciones:



**Limpiar:** permite al usuario borrar los datos introducidos en el formulario.

**Enviar Datos:** esta opción le permite al usuario enviar tanto sus datos personales como los datos de los productos seleccionados para ser almacenados en una Base de Datos.



## IX. CODIFICACIÓN

```
<%Sub MostrarCatalogo
'Muestra el catálogo con enlaces para ir comprando

Dim oRs

set oRs=Server.CreateObject("ADODB.Recordset")

Set ors=oconn.Execute("SELECT * FROM productos WHERE mostrar_principal=yes")%>

<html>

<head>

<title></title>
<script language=JScript>
var message="Derechos Reservados. @E-Commerce 2003";
function click(e)
{
    if (document.all)
    {
        if (event.button == 2)
        {
            alert(message);
            return false;
        }
    }
}
document.onmousedown=click;
</script>
</head>

<FORM ACTION="carrito.asp?ac=ver" METHOD=POST target=principal1
name="formu">

<BODY background="imagenes/fondo.gif">
<TABLE BORDER=0 CELLSPACING="4" CELLPADDING="0" width="563">
<TR>
<TD bgcolor="#0000CC" width="153"></TD>
<TH width="196" bgcolor="#0000CC"><p align="center"><font color="white"
size="2">PRODUCTOS</font></TH>
<TH width="71" bgcolor="#0000CC"><p align="center"><font color="white"
size="2">PRECIO</font></TH>
<TD bgcolor="#0000CC" width="118"></TD>
```



```
<%do until oRs.EOF%>
  <TR>
    <TD width="153"><a
href="carrito.asp?ac=ampliar&produc=<%=ors("Id_producto")%>" target="principal1"><image
border=1 width="150" src=<%=ors("Foto")%> alt="<%=ors("temp")%>" height=150
width=110></a></TD>
    <TD width="196"><p align="center"><a
href="carrito.asp?ac=ampliar&produc=<%=ORS("ID_Producto")%>"
target="principal1"><%=oRs("Nombre")%></a></TD>
    <TD width="71"><p align="center">C$<%=ors("Precio")%></TD>
    <TD width="118"><p align="left"><A HREF =
"carrito.asp?ac=meter&prod=<%=oRs("Id_producto")%>"><font size="2">Enviar al
Carrito</font></A></TD>

<%
oRs.MoveNext
LOOP%>
  </TABLE>

<%
oRs.Close
set oRs=nothing%>

<%
End Sub
```



Esta función permite mostrar el carrito de compras con los detalles de los productos que este contiene y las distintas opciones para borrar un producto, actualizar y comprar.

```
<%Sub MostrarCarrito
' Muestra el contenido del carrito de la compra
Dim oRs,SQL
Dim clave
Dim total,subtotal,gastos
total=0
subtotal=0
gastos=0
set oRs=Server.CreateObject("ADODB.Recordset")%>
<body Background="imagenes/fondo.gif">

<%if request.form="" then%>
    <FORM ACTION=carrito.asp?ac=ver METHOD=POST target=principal1
name="formu">
    <%if(carrito.count=0) then%>
        <table border="0" width="100%" cellpadding="0">
        <tr bgcolor="#0000cc" width="100" height="10">
        <td></td>
        </tr>
        <tr bgcolor="#FFFFFF" width="100%">
        <td align="Center"><Font face="verdana" size="4" >Su Carrito de Compra No
Contiene Productos</font></td>
        </tr>
        <tr bgcolor="#0000cc" width="100" height="10">
        <td></td>
        </tr>
        </table>
    <%Else%>

    <table border="0" width="100%" cellpadding="2">
    <tr>
    <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">PRODUCTO</font></td>
    <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">CANTIDAD</font></td>
    <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">PRECIO</font></td>
    <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">SUB-TOTAL</font></td>
    <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">BORRAR</font></td>
    </tr>
    <tr>
```





```

<td width="20%"></td>
<td width="20%"></td>
<td width="20%"></td>
<td width="20%"></td>
<td width="20%"></td>
</tr>
<%
For Each clave in carrito

SQL="SELECT * FROM productos WHERE id_producto like '%"&clave&"%"
set oRs=oConn.Execute(SQL)%>

<TR>
<TD align="center"><font size="2"><%=oRs("Nombre")%></font
type="verdana"></TD>
<TD align="center"><font size="2"><Input size="2" name="cant"
value="<%=carrito(clave)%>" onKeyPress="if (event.keyCode<48 || event.keyCode>57)
event.returnValue = false;"></font></TD>
<TD align="center"><font size="2"><%=oRs.Fields("Precio")%></font></TD>

<%
' Multiplico el precio por la cantidad. FormatNumber me asegura que es un n° válido
' y convierto todo a tipo Single con CSng.
subtotal=CSng(FormatNumber(carrito(clave))*oRs.Fields("Precio"))
total=total+subtotal%>

<TD align="center"><font size="2"><%= subtotal %></font></TD>
<TD align="center">
<p align="center"><font size="2">
<A HREF="carrito.asp?ac=borrar&prod=<%=clave%>&cant=1" style="text-
decoration:none" OnMouseOver="window.status='Borrar Un Producto'; return true;">Uno</A>
<A HREF="carrito.asp?ac=borrar&prod=<%=clave%>&cant=<%=carrito(clave)%>"
style="text-decoration:none" OnMouseOver="window.status='Borrar Todos Los Productos';
return true;">Todos</A>
</font>
</p>
</TD>

<%
Next

if carrito.Count > 0 then
oRs.Close
set oRs=nothing
End if%>

<TR>
<td width="20%" align="center"></td>

```



```

<td width="20%" align="center"></td>
<td width="20%" align="center"></td>
<TR>
<td width="20%" bgcolor="#0000CC" align="center">&nbsp;</td>
<td width="20%" bgcolor="#0000CC" align="center">&nbsp;</td>
<td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">TOTAL</font></td>
<TD width="20%" bgcolor="#0000CC" align="center">
<font size="2" color="white">

<% Response.Write("C$ ")
Response.Write(total+gastos)%>
</font>
</TD>
<td width="20%" bgcolor="#0000CC" align="center">&nbsp;</td>
</TABLE>
<%
'si no he llamado desde PasarPorCaja
If accion<>"fin" then%>
<TABLE border="0" align="center" height="68">
<TR>
<TD height="62"><a href="carrito.asp" target="_self"></a></TD>
<TD width="50" height="62"></TD>
<TD width="50" height="62"></TD>
<TD width="50" height="62"></TD>
<TD width="50" height="62"></TD>
<TD width="50" height="62"></TD>
<TD width="50" height="62"></TD>
<TD height="62">&nbsp;</TD>
<p><a href="carrito.asp?ac=fin" target="principal1"></a>
</p>
<p align="center"></TD>
</TR>
</TABLE>
<center><Input type="submit" value="Actualizar"></center>
</form>
<% End if
End If%>
<%ELSE%>
<FORM ACTION=carrito.asp?ac=ver METHOD=POST target=principal1
name="formu">
<table border="0" width="100%" cellpadding="2">
<tr>
<td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">PRODUCTO</font></td>

```



```

<tr>
  <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">PRODUCTO</font></td>
  <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">CANTIDAD</font></td>
  <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">PRECIO</font></td>
  <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">SUB-TOTAL</font></td>
  <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">BORRAR</font></td>
</tr>
<tr>
<td width="20%"></td>
<td width="20%"></td>
<td width="20%"></td>
<td width="20%"></td>
<td width="20%"></td>
</tr>

<% For Each clave in carrito%>

<%SQL="SELECT * FROM productos WHERE id_producto like '%" &clave&"%"
set oRs=oConn.Execute(SQL)%>

  <TR>
  <TD align="center"><font size="2"><%=oRs("Nombre")%></font
type="verdana"></TD>
  <TD align="center"><font size="2"><Input size="2" name="cant"
value="<%=Request.form("cant")(x)%>" onKeyPress="if (event.keyCode<48 ||
event.keyCode>57) event.returnValue = false;"></font></TD>
  <TD align="center"><font size="2"><%=oRs.Fields("Precio")%></font></TD>
<%

carrito(clave)=Request("cant")(x)

if(IsNumeric(carrito(clave)) <> True) then%>

  <Script language=JScript>
    alert("Debe introducir un numero")
    if(<%=x%>==1)
    {
      document.formu.cant.focus();
      document.formu.cant.select();
    }
    else

```



```
        {
            document.formu.cant(<%=x-1%>).focus();
            document.formu.cant(<%=x-1%>).select()
        }
    </Script>
    <%
    carrito(clave)=1
else%>
    <%if(carrito(clave)<1) then%>
        <Script language=JScript>
            alert("Debe introducir un numero positivo")
            if(<%=x%>==1)
            {
                document.formu.cant.focus();
                document.formu.cant.select();
            }
            else
            {
                document.formu.cant(<%=x-1%>).focus()
                document.formu.cant(<%=x-1%>).select()
            }
        </Script>
    <%
    carrito(clave)=1
    else
    if carrito(clave)>32767 then
        carrito(clave)=32767
    End if
    subtotal=CSng(FormatNumber(CInt(carrito(clave)))*oRs.Fields("Precio"))
end if
total=total+subtotal%>

        <TD align="center"><font size="2"><%= subtotal %></font></TD>
        <TD align="center">
        <p align="center"><font size="2">
        <A HREF="carrito.asp?ac=borrar&prod=<%=clave%>&cant=1" style="text-
decoration:none" OnMouseOver="window.status='Borrar Un Producto'; return true;">Uno</A>
        <A HREF="carrito.asp?ac=borrar&prod=<%=clave%>&cant=<%=carrito(clave)%>"
style="text-decoration:none" OnMouseOver="window.status='Borrar Todos Los Productos';
return true;">Todos</A>
        </font>
        </p>
        </TD>
    <%
end if
    x=x+1
Next
```



```

if carrito.Count > 0 then
  oRs.Close
  set oRs=nothing
End if%>
  <TR>
  <td width="20%" align="center"></td>
  <td width="20%" align="center"></td>
  <td width="20%" align="center"></td>
  <TR>
  <td width="20%" bgcolor="#0000CC" align="center">&nbsp;</td>
  <td width="20%" bgcolor="#0000CC" align="center">&nbsp;</td>
  <td width="20%" bgcolor="#0000CC" align="center"><font color="#FFFFFF"
size="2">TOTAL</font></td>
  <TD width="20%" bgcolor="#0000CC" align="center">
<font size="2" color="white">
  <% Response.Write("C$ ")
      Response.Write(total+gastos)%>
</font>
  </TD>
  <td width="20%" bgcolor="#0000CC" align="center">&nbsp;</td>
  </TABLE>
<% 'si no he llamado desde PasarPorCaja
if accion<>"fin" then%>
  <TABLE border="0" align="center" height="68">
  <TR>
  <TD height="62"><A HREF=carrito.asp></A></TD>
  <TD width="50" height="62"></TD>
  <TD width="50" height="62"></TD>
  <TD width="50" height="62"></TD>
  <TD width="50" height="62"></TD>
  <TD width="50" height="62"></TD>
  <TD width="50" height="62"></TD>
  <TD height="62">&nbsp;</TD>
  <p><A HREF=carrito.asp?ac=fin></A>
</p>
  <p align="center"></TD>
  </TR>
  </TABLE>
  <CENTER><Input type="submit" value="Actualizar"></CENTER>
  </FORM>
  </html>
  <%End if
      End if
End Sub%>

```



La siguiente función permite agregar un producto al carrito de compras

```
<%Sub MeterArticulo (id)
' Si ya tengo el artículo en el carrito sumo 1 a la cantidad
if carrito.Exists(id) then
    carrito(id)=carrito(id)+1
else
' Si no lo añado
    carrito.Add id, 1
    Session("t")=Session("t")+1
end if
End Sub%>
```

La siguiente función permite eliminar un producto al carrito de compras

```
<%Sub SacarArticulo (id, cantidad)
' Saca un numero "cantidad" de articulos "id" del carrito
if carrito.Exists(id) then
    if cantidad>1 then
'Quito todos así que quito ese articulo del carrito
        carrito.Remove(id)
        Session("t")=Session("t")-1
    else
'Quito uno
        carrito(id)=carrito(id)-1
        if carrito(id)<1 then
            carrito.Remove(id)
            Session("t")=Session("t")-1
        end if
    end if
end if
End Sub%>
```



## X. CONCLUSIÓN

Al iniciar nuestro trabajo nos planteamos crear una página Web dinámica para implementar el comercio electrónico.

Ahora que finalizamos nuestro proyecto hemos encontrado muchas ventajas que proporciona el comercio electrónico a cualquier empresa, ya que le permite tener una presencia global a través de Internet y por lo tanto puede ofertar sus productos de forma permanente las 24 horas del día. Además reduce la pérdida de tiempo al cliente ya que los productos los puede recibir en su propia casa sin necesidad de viajar.

Hoy en día existen protocolos que permiten hacer del comercio electrónico una forma segura para realizar compras por Internet, así el cliente puede enviar sus datos por Internet sin ningún temor a que sus datos sean capturados o alterados por alguien ajeno al destinatario.

En conclusión el comercio electrónico le permite al empresario:

- ✓ Desaparecer los límites geográficos para su negocio.
- ✓ Estar disponible las 24 horas del día.
- ✓ Hacer más sencilla la labor de los negocios con sus clientes.
- ✓ Menos inversión en los presupuestos publicitarios.
- ✓ Reducción de precios por el bajo coste del uso de Internet en comparación con otros medios de promoción, lo cual implica mayor competitividad.
- ✓ Globalización y acceso a mercados potenciales de millones de clientes.



## RECOMENDACIONES

- El servidor donde residirán los archivos de la aplicación debe poseer como mínimo las siguientes características para un funcionamiento óptimo del sistema:

Disco Duro: 20GB.

Memoria: 256MB.

Procesador: 1.2Ghz

Sistema Operativo: Windows 2000 Server.

- Revisar periódicamente la base de datos en donde se almacenarán los datos del cliente y sus pedidos para posteriormente realizar la entrega de los pedidos solicitados.
- Cambiar habitualmente los productos que se presentan en la página principal.





## XI. BIBLIOGRAFÍA

- Jesús Bobadilla Sancho. Alejandro Alcocer Jarabo. Creación de Aplicaciones Web en Windows NT. Active Server Pages. ALFAOMEGA-RAMA. 1999.
- Danny Goodman. Programación en JavaScript. España. AMAYA MULTIMEDIA. 1997.
- Roger S. Pressman. Ingeniería del Software. Un enfoque práctico. 4ta edición. España. McGraw-Hill

### Direcciones en Internet:

- <http://www.aspfacil.com>
- <http://www.asptutor.com>
- <http://www.todoparaweb.com>
- <http://www.lawebdelprogramador.com>
- <http://www.desarrolloweb.com>
- <http://www.webestilo.com>
- <http://www.argenet.com.ar/ecommerce>
- <http://www.monografias.com>



## XII. ANEXOS

Para la creación del “Carrito de Compras” en esta aplicación se utilizó un objeto Dictionary, a continuación se detalla una descripción de dicho objeto.

### **El Objeto Dictionary.**

El objeto Dictionary nos permite almacenar parejas(clave/elemento) de una forma sencilla y rápida. Visual Basic Script nos da gran libertad a la hora de crear nuestros diccionarios. En los diccionarios que creamos, cada clave estará asociada a un elemento. Cuando queremos utilizar ese elemento lo buscaremos mediante su clave. Los diccionarios facilitan mucho el almacenamiento de colecciones, ya que para encontrar cada elemento no tendremos que utilizar índice como en una matriz, que no nos dicen nada, sino que podremos utilizar claves que de alguna forma nos representen el elemento con el que están relacionadas. En un diccionario podemos guardar como elemento cualquier tipo de Visual Basic Script, y para acceder a el, es decir, como clave, podemos utilizar cualquier tipo de datos menos una matriz. Una vez creado el diccionario podremos acceder a sus elementos y modificarlos gracias a sus métodos y propiedades. Instanciar un diccionario es tan fácil como escribir lo siguiente:

```
SET nombre_diccionario=CreateObject(“Scripting.Dictionary”)
```

### **PROPIEDADES**

El objeto Dictionary tiene cuatro propiedades que nos sirven para conocer las características del diccionario que estamos creando. Con ellas podremos acceder a los datos que previamente hemos introducido, e incluso variarlos:

- **CopareMode:** Establece o devuelve el modo de comparación para comparar claves de cadenas. Podemos utilizarla de dos modos:

```
nombre_diccionario.CompareMode
```

En este caso la función nos devolverá un valor que representara el modo de comparación utilizado por nombre\_diccionario. Si nos devuelve un 1 será modo binario, si nos



devuelve un 2 modo texto y si nos devuelve un 3 base de datos. La segunda forma de utilizar esta propiedad es la siguiente:

`Nombre_diccionario.CompareMode=Modo`

En este caso lo que estaremos haciendo es imponer un modo de comparación al objeto Dictionary nombre\_diccionario. Podemos imponer cualquiera de los modos indicados anteriormente además de valores mayores para otros tipos de comparación. La única restricción que nos encontramos es que esto lo deberemos hacer antes de incluir ningún elemento en el diccionario.

- **Count:** Devuelve el numero de elementos de un diccionario. Aplicándolo a un objeto diccionario contara sus elementos. Es muy útil para realizar bucles que recorran todos los elementos de un diccionario. Su sintaxis es muy sencilla:

`nombre_diccionario.Count`

- **Item:** Establece o devuelve un elemento para la clave que le pasemos. Se puede utilizar de dos maneras distintas. La primera la utilizaremos para averiguar el elemento que corresponde a cierta clave:

`nombre_diccionario.Item(clave)`

De esta forma la propiedad Item nos devolverá el elemento asociado a Clave. Con esta propiedad podremos utilizar los elementos de un diccionario. Si Clave no existiese, se crearía una nueva clave en el diccionario con el elemento vacío. La segunda forma en la que podemos utilizar esta propiedad es la siguiente:

`nombre_diccionario.Item(clave)=elemento`

Con esta segunda forma de utilizar Item, lo que estamos haciendo es modificar una clave existente o incluir una nueva si no existiera. Es decir, si clave existe ya en el diccionario modificamos su elemento incluyendo *elemento* en el lugar del anterior. Si la clave no existiera estaríamos incluyendo una nueva entrada en el diccionario de valores *clave* y *elemento*.



- **Key:** Cambia el valor de una clave o crea una nueva si no existiera la antigua. Esta propiedad la utilizaremos cuando queramos modificar el valor de una clave. Key sustituirá el valor antiguo por el nuevo valor que le indiquemos:

```
nombre_diccionario.Key(clave)=nueva_clave
```

## METODOS

El objeto Dictionary incorpora seis métodos que utilizaremos para manejar los datos de nuestro diccionario. Con estos métodos podremos insertar o borrar datos:

- **Add:** Agrega una pareja de clave, elemento al diccionario. Utilizamos este método para llenar el diccionario de los datos que deseemos. Tendremos que indicar al método la clave y el elemento que queremos incluir:

```
nombre_diccionario.Add clave,elemento
```

Hay que tener en cuenta que la clave debe de ser nueva, ya que si existiera anteriormente en el diccionario se produciría un error.

- **Exists:** Devuelve True si la clave especificada existe en el objeto diccionario. Este método nos será útil por ejemplo para comprobar que un usuario introduce las claves del diccionario correctamente y así no encontrarnos luego con un error en el tiempo de ejecución o efectos no deseados al utilizar esas claves. La sintaxis de este método es la siguiente:

```
nombre_diccionario.Exists(clave)
```

- **Items:** Devuelve una matriz con todos los elementos de un diccionario. Después podremos acceder a los elementos del diccionario como si fuese una matriz cualquiera. En la matriz no existirán claves, estas serán sustituidas por los índices de la matriz que irán de 0 a numero de elementos menos uno. La sintaxis de este método es la siguiente:



`nombre_diccionario.Items`

- **Keys:** Devuelve una matriz con todas las claves de un diccionario. Este método sirve para complementar el anterior, ya que realiza la misma función que `Items` pero con las claves. Con este método podremos acceder una a una a las claves de la misma forma que un array. Las claves irán numeradas del 0 a número de elementos menos uno. La sintaxis de este método es la siguiente:

`nombre_diccionario.Keys`

- **Remove:** Elimina una pareja clave/elemento de un diccionario. Al método habrá que indicarle la clave del elemento que queremos borrar. Si no existe la clave se produciría un error. La sintaxis de este método es la siguiente:

`nombre_diccionario.Remove(clave)`

- **RemoveAll:** Elimina todas las parejas clave/elemento de un diccionario. Dejará el diccionario vacío pero este no dejará de ser un objeto `Dictionary`. La sintaxis de este método es la siguiente:

`nombre_diccionario.RemoveAll`



### Diferentes formas de establecer la conexión con una base de datos

Conexión a una base de datos ACCESS sin declarar DSN

```
strconn = "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &  
Server.MapPath("\dirBD\asptutor.mdb")  
set conn = Server.Createobject("adodb.connection")  
conn.open strconn
```

Conexión a una base de datos ACCESS sin declarar DSN usado Jet

```
strconn = ""Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &  
Server.MapPath("/dirDb/asptutor.mdb")  
set conn = Server.Createobject("adodb.connection")  
conn.open strconn
```

Conexión a una base de datos SQL SERVER sin declarar DSN

```
strconn = "Driver={SQL Server};Description=sqldemo;SERVER=127.0.0.1;  
UID=LoginID;PWD=Password;DATABASE=Database_Name  
set conn = server.createobject("adodb.connection")  
conn.open strconn
```

Conexión a una base de datos con DSN

```
set conn = server.createobject("adodb.connection")  
conn.open "comercio"
```

Declaración de un recordset con DSN y con conexión implícita (es decir, sin declarar el objeto connection)

```
set rs = server.createobject("adodb.recordset")  
rs.open "nombre_de_tabla", "Nombre_DSN"
```