



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
"ACATLÁN"**

**Creación de Páginas Web Dinámicas con
Tecnología JSP (JavaServer Pages)**

T E S I S
QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN MATEMÁTICAS
APLICADAS Y COMPUTACIÓN

Presenta:

SANDOR

TRUJILLO

QUINTANA

Asesor: Lic. Gustavo Gudiño Ramírez

Septiembre de 2006



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Mamá:

Gracias por todo el apoyo que me haz brindado, todo el cariño y la paciencia que me haz ofrecido, has sido un pilar muy importante en mi formación tanto profesional como persona, este éxito que obtengo no lo podría haber obtenido si tu ayuda. Siempre te lo agradeceré.

Papá:

Gracias por tener fe en mi, gracias a ti que me enseñaste que con el trabajo y la dedicación es posible cumplir nuestras metas, gracias a ti me convertí en un hombre de bien, este éxito que obtengo también es tuyo.

Hermanos:

Guillermina, Marco, Elvia y Reyna, por su gran ejemplo y dedicación, por estar siempre presentes cuando he necesitado de apoyo. La influencia de ustedes contribuyo a la formar de mi carácter como persona que ahora gracias a eso he logrado obtener una más de mis metas por su invaluable ayuda.

Mis profesores:

Gracias a ellos fui encontrando mi vocación, donde cada uno de ellos tuvo una pequeña contribución para el proceso de mi formación académica, gracias por mostrarme el camino correcto, donde la dirección que elegí, fue con la ayuda de ustedes y se que fue la más acertada.

Índice

Introducción	1
1. Antecedentes	4
1.1. Antecedentes de Internet.	4
1.2. Evolución técnica	5
1.2.1. Protocolos de Internet.....	7
1.2.2. Como funciona Internet.....	8
1.2.3. Internet y otras tecnologías de redes informáticas.....	10
1.3. Comercio Electrónico	12
1.3.1. Tipos y modelos de negocio electrónico.....	13
1.3.2. Ventajas del comercio electrónico.....	25
1.4. Páginas Web	32
1.4.1. Páginas Web estáticas.....	33
1.4.2. Páginas Web dinámicas.....	40
1.4.3. JavaScript.....	41
1.4.4. CGI.....	45
1.4.5. ASP.....	46
1.4.6. PHP.....	47
2. JavaServer Pages	50
2.1. Java	50
2.1.1. Clases y Métodos.....	51
2.1.2. Encapsulación.....	52
2.1.3. Herencia.....	52
2.1.4. Polimorfismo.....	52
2.1.5. Portabilidad.....	52
2.1.6. Seguridad.....	53
2.1.7. Estabilidad.....	53
2.1.8. Memoria.....	53
2.1.9. Multihilos.....	53
2.2. Applets	54
2.3. Servlets	56
2.4. JavaServer Pages (JSP)	58
2.4.1. <i>Tag Library</i>	62
2.5. JDBC	63
2.5.1. <i>Drivers</i>	63
2.6. Administrador de servicios Web	64
3. Aplicación	67
3.1. Creación de un sistema para la administración de recursos de una biblioteca usando tecnología JSP	67
3.2. Lista de datos	73
3.3. Módulo altas	75
3.4. Módulo bajas	78
3.5. Módulo cambios	80
3.6. Módulo consultas	82
3.7. Módulo configurar	82
3.7.1. Descripción de los módulos de configuración.....	83
3.8. Búsqueda	95

4. Conclusión.....	97
4.1. Resultados obtenidos.....	97
4.2. Mejoras para el futuro.....	97
4.3. Nuevas líneas de investigación.....	97
4.4. Comentarios finales.....	98
Anexo A Acrónimos.....	100
Anexo B Código Fuente Seleccionado.....	102
Referencias de Información.....	106
Referencias Bibliográfica.....	106
Referencia Electrónicas.....	108
Referencias de Internet.....	110

Introducción

Una de las grandes tendencias tecnológicas en estos días, es la publicación de páginas Web. En los últimos años, la publicación de las páginas Web tuvo un crecimiento exponencial, pero no todas han tenido el éxito esperado y no todas han sobrevivido. Muchas han desaparecido, otras en cambio han aprendido a evolucionar a lo largo de los años hasta poderse consolidar en el mercado. Las razones de este éxito o fracaso son muy variadas, pero lo cierto es que la tendencia del Web sigue adelante a pesar de todo. Hoy en día el potencial del Internet es cada vez mas grande.

La industria en general ya reconoce la necesidad de contar con una presencia Web. Ello no solo en términos de imagen sin en la interacción con procesos y sistemas de otras empresas. Lo anterior trasciende las barreras geográficas, ya no se trata de nivel nacional sino de nivel global. Sin embargo, saber cual es el camino que debe seguirse para poder alcanzar las metas establecidas es una de las decisiones más difíciles de tomar.

Lo cierto es que hay que saber elegir la tecnología apropiada para poder construir un sitio Web. Es necesario considerar costos, rendimiento, versatilidad y alcances. Existe en el mercado toda una gama de productos que cubren absolutamente todas las necesidades para el desarrollo de un sitio Web. La decisión de utilizar una u otra tecnología puede ser la clave entre el éxito y el fracaso.

En la actualidad una de las herramientas que mas éxito y versatilidad ha alcanzado, es el uso de un lenguaje de programación orientado a objetos llamado Java. Esto se debe al potencial y alcance que ha generado desde que fue concebido.

Sin duda alguna, una de las claves del éxito que ha llevado a Java a ser uno de los lenguajes de programación más populares es la gran gama de tecnologías que ofrece. Por mencionar unos ejemplos se pueden crear aplicaciones para una PC stand alone, para el servidor mas avanzado o hasta para la telefonía celular o tener acceso a una base de datos para llevar un simple inventario. Para poder tomar la decisión sobre la utilización de esta tecnología, se deben realizar estudios para saber si la solución es la más apropiada para su uso. Es por ello que se realizó una investigación con el objetivo

de evaluar las ventajas y desventajas de Java en términos de su funcionalidad orientada a la creación de páginas Web dinámicas así como la conectividad al acceso a bases de datos. Se demostrará por que Java es la mejor herramienta para el desarrollo de páginas Web, la más poderosa y por lo tanto la mas recomendada para su uso.

Con el fin de llevar a cabo la anterior evaluación, se examinará la funcionalidad del lenguaje con la creación de un caso práctico que contemple el uso de esta tecnología así como las herramientas que facilitan el uso de este lenguaje. El propósito final será el comprobar cuales son las ventajas y limitantes de la herramienta en el marco del problema.

Esta investigación se enfocara a evaluar el uso de esta tecnología y proponerla como una opción viable para cualquier tipo de proyecto que requiera el uso de tecnología de páginas Web dinámicas, sin importar el costo o las dimensiones de este, teniendo aquí una aplicación terminada para realizar las pruebas necesarias, haciendo énfasis en sus características y atributos más importantes que la hacen una buena herramienta de desarrollo para crear páginas Web dinámicas. Se evaluarán diferentes aspectos del lenguaje analizando así, sus atributos propios, obviamente no es la solución para todos los problemas en cuanto a lo que el Web concierne, no es un lenguaje infalible, pero si tiene ciertas características que lo hacen una herramienta poderosa. En esta investigación se llegara a una conclusión, pero la última decisión la tiene el usuario final del sistema, para el cual están destinadas todas las aplicaciones que se puedan desarrollar con ella.

La evolución de las tecnologías para el desarrollo de software, parece tener un flujo constante, pero algo de lo que se podría estar seguro, es que el uso de esta herramienta, se ira incorporando a cualquier tendencia o cambio tecnológico que llegara a surgir.

La organización de este trabajo comprende 4 capítulos.

Capítulo 1. Antecedentes

Presenta los antecedentes de Internet, así como tecnologías que fueron surgiendo para complementar dicho concepto. La utilidad de Internet al servicio de los negocios así como los diferentes giros de empresas que surgen como consecuencia.

Capítulo 2. Java Server Pages

Presenta las principales características de Java, los aspectos trascendentes e importantes que ofrece el lenguaje, así como las tecnologías de Java y que van a ser utilizadas para la creación del sistema.

Capítulo 3. Aplicación

Creación de un sistema para administración de una Biblioteca, con una serie de módulos que sirven al propósito de demostrar como se ve reflejada la tecnología Java para la creación de aplicaciones. Establecer los puntos necesarios para evaluar a Java para la creación de aplicaciones Web

Capítulo 4. Conclusión

Se evaluarán los resultados obtenidos con base en las características propias de Java con la finalidad de recopilar la información necesaria y llegar a una respuesta final, evaluando si se llegaron a cumplir los requerimientos necesarios para hacer de Java una herramienta recomendable para la creación de aplicaciones Web.

1 ANTECEDENTES

1.1 Antecedentes de Internet

Los primeros indicios de Internet se encuentran en Estados Unidos, en la década de los 60's, en una red llamada ARPANET, desarrollada y financiada por la Agencia de Proyectos de Investigación del Departamento de Defensa, llamada inicialmente ARPA (*Advanced Research Project Agency*) y más adelante DARPA (*Defense Advanced Research Project Agency*), esta agencia proponía la creación de una red de comunicación con varias instalaciones alrededor de todo el país, con el propósito de que en caso de alguna posible contingencia de origen natural o por la intervención de algún posible enemigo, el sistema siguiera funcionando. El funcionamiento de este proyecto consistía en que ninguna instalación debía poseer privilegios sobre alguna otra, es decir que en caso del mal funcionamiento de una de ellas, las demás debieran tener suficiente autonomía en el flujo de la información redireccionándose, desviando el contacto con la posible central que haya quedado inutilizada o destruida.

En julio de 1961 se publicó en el MIT (Massachusetts Institute of Technology), el primer documento sobre la teoría del Sistema de Paquetes y en 1964 el primer libro sobre el tema. El sistema de paquetes consiste en la fragmentación o empaquetamiento de los datos para así, agilizar el flujo de la información. La palabra "package" (Paquete) fue adoptada por el trabajo de la NPL (National Physic Lab) y la medición de la velocidad de la línea, kilobits per second (kbps), fue propuesta por ARPANET. Cabe señalar que a finales de 1966 en el DARPA se desarrolló el concepto de redes de computadoras, para crear ARPANET, mismo que se publicó en 1967.

Fue hasta 1969, cuando se conectaron juntas cuatro computadoras Host para el ARPANET inicial, rápidamente se agregaron computadoras al ARPANET durante los años siguientes, y el trabajo procedió en completar la funcionalidad de un protocolo Host-to-Host y otro software para la red. En diciembre de 1970 la Network Working Group (NWG) terminó el protocolo inicial Host-to-Host de ARPANET, llamado el Protocolo de Control de Red (*Network Control Protocol*, NCP). A los sitios de ARPANET se les implementó el protocolo NCP durante el periodo 1971-1972.¹

¹ <http://www.webmasterbase.com/article.php> [18/02/2004]

En 1972 se organizó exitosa demostración de ARPANET en la Conferencia de Comunicación de Computadora Internacional (International Computer Communication Conference, ICC). Ésta fue la primera demostración pública de la nueva tecnología de redes. En ese mismo año se dió origen de una nueva aplicación, el correo electrónico.

En 1973 la DARPA inició un programa de investigación orientado hacia nuevas técnicas y tecnologías para conectar una serie de redes de trabajo de cualquier tipo, el objetivo fue desarrollar protocolos que permitieran la comunicación entre computadoras a través de múltiples redes de trabajo. Esto fue llamado "*Proyect Internetting*", asimismo a este sistema se le conocido como Internet. Este sistema de Protocolos que fue desarrollado en el curso de la investigación fue conocido como el TCP/IP, después de dos protocolos iniciales desarrollados el *Transmission Control Protocol* (TCP) y el *Internet Protocol* (IP).²

Después de 1989, se comenzó a integrar el apoyo a otras suites de protocolos dentro de la fabrica de redes básicas. El presente énfasis en el sistema está en los multiprotocolos que trabajan entre sí, y en particular, con la integración de los protocolos de Interconexión de Sistemas Abiertos (*Open Systems Interconnection*, OSI) dentro de la arquitectura del sistema. Cerca de 100 protocolos de TCP/IP se hicieron disponibles a finales de los 80's para el dominio público. A principios de los 90's, las implementaciones protocolares OSI se hicieron publicas y en 1991, Internet había crecido hasta incluir unas 5,000 redes en más de tres docenas de países y ha servido a más de 700,000 computadoras Host utilizadas por más de 4,000,000 personas.

A finales de los 80's, la población de usuarios de Internet y de redes se extendió internacionalmente y comenzó a incluir medios comerciales. De hecho, la mayor parte de los sistemas hasta hoy, esta hecho con el apoyo de instituciones educativas, de investigación, de negocios y de organizaciones gubernamentales de todo el mundo.

1.2 Evolución técnica de Internet

Después de varios años de historia, Internet ha funcionado gracias a la colaboración conjunta entre cooperativas. Ciertas funciones importantes han sido críticas para su

² <http://www.webmasterbase.com/article.php> [18/02/2004]

funcionamiento, las especificaciones de los protocolos por las cuales, los componentes del sistema operan, éstos se desarrollaron originalmente en el programa de investigación de DARPA, pero en los últimos años, este trabajo se ha emprendido con una amplia base con apoyo de las agencias Gubernamentales en varios países, industrias y la comunidad académica. El Consejo de Actividades de Internet (*Internet Activities Board*, IAB) se creó en 1983 para guiar la evolución del protocolo TCP/IP y proporcionar consejos de investigación para la comunidad de Internet.³

La IAB se ha reorganizado varias veces durante el curso de su existencia; ahora tiene dos componentes primarios: el *Internet Engineering Task Force* y el *Internet Research Task Force*. El anterior tiene la responsabilidad primaria de probar la evolución del protocolo TCP/IP, su regularización con la concurrencia del IAB, y la integración de otros protocolos en el funcionamiento de Internet (por ejemplo, los protocolos de Interconexión de Sistemas Abiertos). La *Internet Research Task Force* continúa organizando y explorando conceptos en redes bajo la guía del *Internet Activities Board* y con apoyo de varias agencias gubernamentales.

Existen funciones importantes para el funcionamiento de IAB: la publicación de documentos describiendo Internet y la asignación y registro de varios identificadores necesarios para el funcionamiento del protocolo. A lo largo del desarrollo de Internet, sus protocolos y otros aspectos de su funcionamiento se han documentado, primero en una serie de documentos llamados *Internet Experiment Notes* y después en una serie de documentos llamados *Requests For Comment* (RFC). Este último fue usado inicialmente para documentar los protocolos del primer paquete desarrollado por DARPA, la documentación de ARPANET, comenzó en 1969, y se ha convertido en el principal archivo de información sobre el Internet. En la actualidad, la función de publicaciones es proporcionada por un editor de RFC.

El registro de identificadores es proporcionado por la *Internet Assigned Numbers Authority* (IANA) quien ha delegado una parte de esta responsabilidad a un Registro de Internet (*Internet Registry*, IR) que actúa como un almacén central para la información de Internet y que proporciona la asignación central de red e identificadores de sistemas autónomos, en algunos casos para subsidiar registros localizados en varios países. El Registro de Internet (IR) también proporciona mantenimiento central del

³ <http://www.webmasterbase.com/article.php> [18/02/2004]

Sistema de Nombres de Dominio (DNS)⁴ la fuente de los datos, el cual apunta al distribuidor subsidiario del servidor DNS en todas partes de Internet. El DNS distribuye la base de datos en uso, entre otras cosas el uso de alias, para asociar el Host y el nombre de las redes con su propia dirección de Internet y es crítico para el funcionamiento de los protocolos TCP/IP de alto nivel incluyendo el correo electrónico.

Existe un gran número de Centros de Información de Redes (*Network Information Center* NIC) localizado a lo largo de Internet para proveer documentación, guía, consejo y ayuda. Como Internet continúa creciendo internacionalmente, la necesidad de la alta calidad de las funciones del NIC se incrementan. Aunque la comunidad inicial de usuarios de Internet fue dirigida a la rama de ciencias computacionales e ingeniería, sus usuarios ahora comprenden una amplia gama de disciplinas en las ciencias, artes, literatura, negocios, la administración militar y de gubernamental.

1.2.1 Protocolos de Internet

Internet es una extensa red de conexiones entre computadoras independientes unas con otras, con diferentes y variadas formas de comunicarse, esto se logra, mediante el uso de protocolos. Los protocolos de Internet son un conjunto de reglas definidas entre dos dispositivos que tienen como finalidad comunicarse entre ambos. Los protocolos de uso común son: HTTP (Hiper Text Transfer Protocol), TCP/IP (Transmission Control Protocol/Internet Protocol), FTP (File Transfer Protocol), WAP (Wireless Application Protocol), etc.

El HTTP es el protocolo más utilizado para el acceso a páginas HTML, es del tipo request/response (petición/respuesta), por medio del cual un navegador Web realiza una petición para solicitar un documento y el servidor responde a esa petición enviando el archivo solicitado. Es un protocolo que se utiliza principalmente para el intercambio de archivos multimedia.

El TCP/IP se utiliza para realizar los enlaces, comunicación e intercambio de información entre dos computadoras, no importando las características o la plataforma en la que esté basada cada una de ellas, de hecho ésta fue la principal razón de su creación para poder realizar una conexión entre dos computadoras aunque la

⁴ Domain Name System

arquitectura o el sistema operativo sean incompatibles. El protocolo realiza el intercambio de información al dividirla en pequeños paquetes, para facilitar su envío.

El protocolo que se utiliza para el intercambio de archivos y manipulación de directorios entre dos computadoras y que permite la comunicación y el acceso a los archivos que estén almacenados en el lado del servidor para su libre acceso (siempre y cuando en el lado del servidor se otorguen los permisos necesarios para el acceso a la información) es el FTP.

El WAP es un protocolo que permite el acceso a Internet a través de móviles que toleren esta tecnología, mediante el lenguaje WML⁵. Las principales limitaciones son: la capacidad de transmitir información de texto e imágenes muy básicas, baja velocidad, sólo se puede acceder a páginas en formato WML.

1.2.2 Como Funciona Internet

Existen los TLD (Top Level Domains) que son los identificadores de país (de dos letras como son uk, mx, ar, br. Etc.) más los dominios especiales (como por ejemplo: edu, com, mil, gov, .gob, org, int, net, tv, etc.).

En 1989 en el CERN por Timothy (Tim) John Berners-Lee, se desarrolló el primer protocolo de intercambio de archivos, el HTML⁶, además de crear el HTML⁷ un lenguaje basado en etiquetas para la creación de las páginas web, creando también el primer navegador. En 1993 se crea el primer navegador Web de uso comercial, mejor conocido como Mosaic por Marc Andreessen (el predecesor de Netscape), desarrollado en el NCSA. Siendo este el navegador que popularizó el uso del Internet.

Existen 2 tipos modos de uso, cliente y servidor. El *cliente* es el usuario que esta conectado desde su PC y realiza operaciones de petición (request) para obtener una respuesta (response) por parte del *servidor* que es el que atiende las peticiones solicitadas (es en el lado del servidor donde se almacena la información referente al contenido de las páginas Web).

⁵ Wireless Markup Language

⁶ Hyper Text Transfer Protocol

⁷ Hyper Text Markup Language

En 1984 se crean los DNS ⁸, proporcionando sufijos a las direcciones electrónicas para su fácil acceso, ya que anteriormente se tenía que insertar la dirección electrónica que consistía en una cadena de varios números.

Respecto al DNS, es el sistema empleado en Internet para poder asignar y usar universalmente nombres únicos que se refieren a los equipos conectados a la red. Actualmente el DNS es una gran base de datos distribuida jerárquicamente por Internet; existen infinidad de servidores que interactúan entre sí para encontrar y facilitar a las aplicaciones clientes que los consultan la traducción de un nombre a su dirección de red IP asociada con la que poder efectuar la conexión deseada.

La finalidad del DNS es permitir el incremento del sistema de nombres de Internet, por medio de una distribución jerárquica de dominios delegados. Los dominios son entidades administrativas cuyo propósito es subdividir la carga de gestión de un administrador central repartiéndola entre distintos subadministradores. De ésta forma se crean distintos niveles de dominios delegados, donde cada administrador asigna nombres unívocos a su nivel separado por puntos.

Una URL (Uniform Resource Locator) es una referencia textual hacia algún segmento de información (sea una página Web o documento multimedia) que se encuentra localizado en Internet. El URI (Uniform Resource Identifier)⁹ Es un identificador global único para un objeto físico o abstracto en el Web. Una URL es como una URI que especifica la recuperación por medio de un protocolo (HTTP o HTTPS para aplicaciones Web) y la localización física de un objeto.

Se han creado nuevas iniciativas para lanzar otro experimento sobre futuras tecnologías de red. Internet2 (I2) es una incipiente plataforma de pruebas para aplicaciones Internet avanzadas, como QoS (Quality of Service, calidad de servicio). I2 está construido sobre ATM (modo de transferencia asíncrona), un estándar en redes rápidas de conmutación de paquetes. QoS es la capacidad de una red IP de garantizar

⁸ Domain Name System

⁹ Eric Armstrong, Stephanie Bodoff, Debbie Carson, Maydene Fisher, Scott Fordin, Dale Green, Kim Haase, Eric Jendrock. The Java Web Services Tutorial. [en línea] Sun Microsystems, Febrero 2003. <http://Java.sun.com/webservices/tutorial.html>

servicio ininterrumpido cuando sea necesario para aplicaciones que requieren un gran ancho de banda como la vídeo conferencia. Entre los participantes en I2 se encuentran 100 universidades estadounidenses, cada una de las cuales invertirá 500.000 dólares así como otros socios empresariales como Advanced Network & Services, Cisco e IBM.

Los Network Computers (NC) y los NetPCs son dispositivos de red económicos que permiten que más usuarios se conecten a la red. Para que el comercio electrónico sea difundido, los dispositivos de red deben ser mas accesibles como la televisión y la telefonía. Los NetPCs son PC's con dispositivos limitados para disminuir su costo. Viene con TCP/IP, tarjetas inteligentes y Java. El software y los contenidos residen en servidores y los NC's se traen de ellos lo que necesitan. Algunos comparan los NC's con "terminales tontas" de los mainframes. Una "terminal tonta" consistía sólo en un monitor y un teclado, donde todos los procesos se realizaban en el servidor (el mainframe).

La comercialización de Internet está forzando a las empresas a diferenciar sus productos creando productos incompatibles con los demás. Al contrario que la Internet pública donde los estándares eran abiertos, las firmas empresariales intentan capturar y dominar el mercado con sus productos propietarios. En un entorno de este tipo, TCP/IP tendría una oportunidad muy pequeña de convertirse en un estándar y abrir definitivamente la economía digital.

El Intercambio Electrónico de Datos (EDI) es un sistema automático de intercambio de datos entre empresas. Las dos áreas primarias de EDI son el intercambio de datos y la transferencia electrónica de fondos (EFT) utilizada entre bancos, que generan un gran volumen en un relativamente pequeño número de datos y una relación a largo plazo. La alta seguridad de EDI ofrece transacciones muy robustas en comparación con Internet debido a que EDI funciona sobre redes de valor añadido privadas y cerradas

1.2.3 Internet y otras tecnologías de redes informáticas

Existen medios de comunicación muy similares a Internet en concepto, pero estos estan enfocados a un pequeño sector, tanto a nivel gubernamental o empresarial.

Una Intranet es una red de acceso local o privado, creado para el acceso exclusivo de una organización. La Intranet posee la misma tecnología que Internet, con la diferencia que esta restringida solamente a usuarios dentro de la organización. La Intranet es una aplicación de la tecnología Internet a una red cerrada. En ella se administra la información para el libre acceso dentro de la organización. Facilita el flujo de datos interno, ya que el tráfico es menor en comparación con el Internet, la información fluye más rápidamente. Permite una total comunicación con todo el personal de la organización. Proporciona una forma relativamente eficiente en costos de conexión y administración de redes corporativas e información distribuida. Una Intranet es una red cerrada, limitada a la extensión de una empresa, pero que utiliza estándares abiertos como TCP/IP en lugar de los tradicionales protocolos propietarios utilizados en las redes LAN (red de área local) y en la WAN (redes de área extensa, normalmente tipo LAN conectadas por cable, teléfono y conexiones sin cable), La Intranet utiliza los protocolos de comunicación (TCP/IP), el sistema de red (red IP) y el interfaz de usuario (navegadores, correo electrónico, etc.) de Internet. El aspecto clave de una Intranet en su aplicación corporativa es la facilidad en la integración de procesos. Una compañía con un grupo de negocios móvil tiene que procesar una entrada continua de datos de los clientes e integrar esos datos en el proceso de manufactura, abastecimiento y envío. Eso puede hacerse en redes no necesariamente mediante Internet, pero la Intranet ofrecen un costo más bajo y un acceso fácil. La administración de la información corporativa puede convertirse ahora de manera multi-funcional. La sorprendente velocidad a que ha crecido Intranet entre los usuarios corporativos demuestra la fuerza de la propia Internet. Varias investigaciones indican que el gasto en Intranet corporativa supera el nivel de inversión en negocios Web orientados al consumidor (tiendas Web, escaparates, subastas, etc.). La compatibilidad de la Intranet corporativas con Internet será un factor significativo en la economía digital. En este sentido, la información del productor y vendedor será fácilmente accesible desde fuera, haciendo más fácil administrar y diseminar información a los consumidores. La Intranet utilizan nombres de dominio que pueden no ser reconocibles por servidores externos. Para identificar a los servidores, la solución más popular es utilizar el protocolo ligero de acceso a directorios (LDAP). LDAP fue desarrollado para manejar varios sistemas de archivos incompatibles a la vez.

Una Extranet posee las mismas características que la Intranet, teniendo además la capacidad de extender su alcance a otras localidades fuera de la organización,

restringiendo o no el acceso a la información. Una Extranet es una WAN privada que utiliza protocolos públicos, es decir, una Extranet es una red virtual privada entre grupos privados basada en redes y protocolos abiertos. Para asegurar la seguridad y privacidad, una Extranet se apoya en canales seguros utilizando protocolos especiales e identificadores digitales. En una línea, una Extranet es una calle privada construida sobre suelo público.

1.3 Comercio electrónico

El proceso de vender bienes y servicios por Internet se la llama comercio electrónico (e-commerce) o negocio electrónico, el comercio electrónico va más allá de vender productos mediante un catálogo electrónico. Entre los beneficios de la interactividad de Internet están la comodidad y una mayor eficiencia, el mejor servicio al cliente, costos de transacción más bajos y nuevas oportunidades para relacionarse.

El mercado electrónico parece ser un mercado muy ventajoso, donde hay numerosos vendedores y compradores internacionales, que a su vez poseen abundante información sobre el mercado y los productos, donde los intermediarios no son necesarios. Un mercado de este tipo es muy competitivo y eficiente, sin embargo, un examen más detenido indica que la búsqueda del consumidor no es muy eficiente. Aunque los vendedores al mayoreo y al menudeo puede que no sean necesarios, hay otros tipos de intermediarios esenciales para que el mercado electrónico funcione adecuadamente. Todos estos factores añaden costos a la transacción.

Las empresas centradas en el comercio electrónico comenzaron hace más de dos décadas con la introducción del intercambio electrónico de datos (EDI) entre firmas comerciales (envío y recibo de pedidos). El comercio electrónico orientado al consumidor tiene también una larga historia: la utilización de un cajero electrónico o el pago por tarjeta de crédito, es una transacción electrónica. EDI¹⁰ y ATM¹¹, operan en un sistema cerrado, son un medio de comunicación más conveniente, estrictamente entre las partes involucradas.

¹⁰ Electronic Data Interchange

¹¹ Asynchronous Transfer Mode

Los procesos del comercio electrónico actual, están básicamente orientados a las computadoras personales debido al origen de Internet. La primera etapa de la expansión del comercio electrónico reside en la base instalada de usuarios de computadoras. El comercio electrónico es un procedimiento mediante el cual una determinada empresa o particular realiza la venta de sus productos de forma automática a través de Internet.

Dentro de una aplicación de comercio electrónico se pueden distinguir dos partes diferenciadas. Por un lado, la aplicación de comercio propiamente dicha. Se considera dentro de esta categoría al programa que es capaz de contabilizar de forma automática la acumulación de productos adquiridos por el cliente, los gastos de envío, etcétera. Dependiendo de lo sofisticada que sea la aplicación permitirá o no realizar control de stocks, búsqueda de productos, etc. Por otro lado, la implementación de los diferentes procesos de pago. Dentro de los procesos de pago más habituales pueden destacarse los siguientes: cheque, transferencia bancaria, pago contra reembolso y, por supuesto, la pasarela de pago. Aunque cada uno de ellos tienen sus ventajas e inconvenientes, puede destacarse la evidente importancia del pago por tarjeta de crédito, ya que permite que durante el proceso se compruebe la validez de la tarjeta y la existencia de fondos, y que el cobro se realice en tiempo real.

El *e-commerce* es el marketing, ventas y compras de productos y servicios en Internet, a diferencia del negocio electrónico (e-business), es usar electrónicamente la información para mejorar el desempeño, crear valor y establecer nuevas relaciones entre clientes y proveedores. El uso electrónico de la información para establecer relaciones tanto al interior de una organización, como con clientes y proveedores, a través de la conexión de empresas, instituciones, departamentos y/o unidades de negocio. Abarca todas las áreas de una organización, desde el abastecimiento hasta las ventas y mercadotecnia y servicios de atención al cliente, incluyendo las áreas administrativas de producción y operaciones, así como a las áreas de toma de decisiones.

1.3.1 Tipos y modelos de negocio electrónico

El término comercio electrónico ha evolucionado desde su significado original de compra electrónica al significado actual que abarca todos los aspectos de los procesos

de mercado y empresas Web. Comercio electrónico también se puede dar como sinónimo de ventas online, explicado de manera mas simplificada puede ser entendido como hacer negocios online o vender y comprar productos y servicios a través de sitios Web. Los productos comercializados pueden ser bienes o servicios. También pueden ser productos digitales como noticias, imágenes, sonido, bases de datos, software y todos los tipos de productos relativos a la información. El manejo del comercio electrónico es muy similar a las ventas por catálogo.

La definición orientada al proceso del comercio electrónico ofrece una visión más general de lo que representa el comercio electrónico. Los procesos internos de la empresa (como la manufactura, el inventariado, administración contable, etc.) y los procesos Negocio a Negocio (*Business to Business*) quedan afectados por la tecnología y la red en igual medida que los procesos Negocio a Consumidor (*Business to Consumer*). Incluso las funciones gubernamentales, educativas, sociales y procesos políticos sufren estos cambios.

Se define el comercio electrónico, como al intercambio de información entre dos puntos con la finalidad de adquirir un bien o un servicio mediante Internet. El intercambio de información se divide en los siguientes modelos:

- ❑ Business to Business "**B2B**" (*Negocio a Negocio*).
- ❑ Business to Consumer "**B2C**" (*Negocio a Consumidor*).
- ❑ Consumer to Business "**C2B**" (*Consumidor a Negocio*).
- ❑ Consumer to Consumer "**C2C**" (*Consumidor a Consumidor*).
- ❑ Business to Government "**B2G**" (*Negocio a Gobierno*).
- ❑ Consumer to Government "**C2G**" (*Consumidor a Gobierno*).
- ❑ Peer to Peer "**P2P**" (*Persona a Persona*).

El modelo Business to Business (B2B) se refiere principalmente al comercio entre proveedores, se utiliza principalmente para los distribuidores. Aquí se encuentra concertada la mayor cantidad de recursos económicos. Es el modelo con más

posibilidades de crecimiento. Optimiza las funciones de abastecimiento y distribución de la empresa. Relaciona de manera inteligente a compradores y vendedores.

El modelo Business to Consumer (B2C) se refiere al comercio dirigido al público en general, es decir al comercio enfocado a los clientes que adquieren un bien o un servicio para su propio consumo. Permite la eliminación de intermediarios a través del ofrecimiento de los productos directamente a los consumidores.

En el modelo Consumer to Business (C2B), el consumidor ofrece condiciones de compra o servicio a todos los proveedores del mercado, Los proveedores realizan las ofertas.

El modelo de comercio Consumer to Consumer (C2C) se utiliza principalmente en las páginas o portales dedicados a "subastas", esto es, una persona expone un bien o servicio con un precio inicial, por lo que otro individuo realiza ofertas para adquirir el producto; y es así como se realiza el intercambio de información de cliente a cliente utilizando como intermediario o mediador el sitio donde se esta realizando la subasta. Los consumidores interactúan directamente unos con otros. El principal problema es el cumplimiento de compromisos comerciales entre consumidores. Permite la comercialización de productos y servicios entre los mismos consumidores.

En el modelo Business to Government (B2G) incluyen todas las transacciones comerciales que se realizan entre los proveedores de bienes o servicios y el gobierno. Se establecen relaciones cooperativas para permitir la optimización de trámites burocráticos. Permite al gobierno adquirir bienes y servicios a un precio más bajo sin el desgaste de grandes aparatos burocráticos.

En el modelo Consumer to Government (C2G), es principalmente las operaciones que realizan los particulares con el gobierno (como por ejemplo el pago de impuestos vía Internet).

En el modelo Peer to Peer (P2P), se trata principalmente de un tipo de intercambio de información sin fines de lucro, es decir que no necesariamente implica una transacción comercial, esto es que una persona se comunica directamente con otra, principalmente se utiliza para el intercambio información o archivos de cualquier tipo sin intermediarios, mas que la utilización de un software (de uso comercial o gratuito) que permite las conexiones o enlaces.

Los modelos de negocio se podrían definirse como el giro al cual se encuentra orientado

□ ***e-shop***

Es la promoción y comercialización de bienes y servicios que realiza una empresa Web. Brinda la facilidad para seleccionar, ordenar y pagar. Frecuentemente se combina con mecanismos de marketing, ventas y distribución tradicionales. Sus utilidades son la reducción de costos, aumento de ventas y posiblemente la venta de espacios publicitarios.

□ ***e-procurement***

Es la solicitud y obtención de bienes y servicios que realiza una organización a través del Web. Frecuentemente lo realizan las entidades gubernamentales o grandes corporaciones, frecuentemente se combina con mecanismos de negociación y contratación electrónica, así como trabajo en conjunto. Sus utilidades vienen principalmente de la reducción de sus costos.

□ ***e-auction***

Es una implementación electrónica del mecanismo de subastas y ofertas. Integra las subastas con mecanismos de ofertas, contratación, pagos y distribución. Su fuente de ingresos es, para los operadores de sitio, la renta de la aplicación, cobro de comisiones y venta de espacios publicitarios; para el vendedor es el inventario reducido, descontinuado a bajo precio y para el comprador es el bajo costo.

□ ***e-mall***

Conjunto de e-shops, reunidas bajo un aspecto en común. Pueden estar especializados en algún segmento del mercado, proporcionar valor agregado. Su fuente principal de ingresos, para el operador del sitio es la renta de la aplicación, cobro de las comisiones y venta de servicios y espacios publicitarios; para el vendedor, que genera confianza, menores costos y el tráfico del sitio; para el comprador es el fácil acceso a diferentes opciones y la confianza que adquiere al ver una gran variedad de opciones.

❑ **Comercialización por terceros**

La comercialización se realiza a través de terceros, frecuentemente existe un canal adicional a los existentes, integra características para búsqueda, selección, contratación, pago de productos, así como para seguimiento y administración. Su fuente principal de ingresos son las comisiones por membresías, comisiones del servicio o comisiones por venta.

❑ **Comunidades Virtuales**

El valor y el contenido lo proporcionan los miembros de la comunidad, quienes agregan información al sitio. Pueden presentarse en combinación con otros modelos de comercio electrónico, sus principales ingresos son las cuotas por membresías o espacios publicitarios.

❑ **Proveedores de servicios a la cadena de valor**

Se especializa en alguna función específica de la cadena de valor como pagos electrónicos y logística. Agrega ventajas competitivas a las funciones de la cadena de valor. Su principal ingreso es una cuota o un porcentaje previamente concertado, ya sea fijo o sobre volumen de ventas.

❑ **Integrador de la cadena de valor**

Se enfoca en integrar los múltiples procesos de la cadena de valor, con el potencial de explotar la información que fluye entre dichos procesos como un valor agregado. Su principal ingreso es por medio de cuotas por consultoría o comisión por transacción.

❑ **Plataformas de colaboración**

Proporciona un grupo de herramientas y un ambiente de información para el trabajo en colaboración entre empresas. Se enfoca en funciones específicas o de soporte a un grupo de trabajo virtual. Su fuente de ingresos es a través de comisiones por el uso de la aplicación o por la venta misma.

❑ **Brokers de Información**

Proporciona una serie de servicios de información, tales como investigación, estadísticas, estudios especializados, asesorías, certificación, etc. La fuente de ingresos

es a través de suscripciones, venta de información, consultorías, venta de certificados, software, etc.

El modelo de negocios es quizás el más discutido y menos entendido aspecto del Web. Hay mucho que hablar acerca de cómo el modelo tradicional de negocios ha cambiado en el Web ¹².

Una forma genérica de modelos de negocios que se observan en el Web incluyen:

- Brokerage (Corredores o Agentes).
- Advertising (Publicidad)
- Infomediary (Información de Segunda Mano)
- Merchant (Comerciantes).
- Manufacturer (Fabricantes)
- Affiliate (Afilaciones).
- Community (Comunidades).
- Subscription (Suscriptores).
- Utility (Utilidad)

Dentro de cada una de estos modelos existen otros más especializados:

□ **Brokerage**

Los corredores (Brokers) son creadores de mercados, ellos intentan comprar y vender al mismo tiempo facilitando las transacciones. Un corredor hace dinero cargado a una cuenta por cada transacción cumplida.

□ **Compra / Venta finalizada**

Esto puede ser un corredor financiero en línea, donde los clientes compran y venden por medio de instrumentos financieros. También los agentes de viajes entrarían dentro

¹² Managing The Digital Enterprise, Professor Michael Rappa

de esta categoría. Los pagos al corredor son por una compra o venta finalizan con una transacción finalizada.

❑ **Intercambio de mercado**

En el intercambio de mercado, el corredor comúnmente carga al vendedor una transacción realizada en el valor de la venta

❑ **Comunidad de negocios de marcas.**

También llamada Comunidad Web Vertical, es como un sitio, cuya fuente de información y diálogo para un particular mercado vertical. La comunidad de mercados verticales contiene productos de información en guías de compras, proporcionando y produciendo directorios, noticias diarias de la industria y artículos, listas de trabajos y clasificados.

❑ **Comprador agregado (*Buyer Aggregator*)**

Un comprador agregado es como el proceso de realizar una compra de manera individual por el Internet realizando una transacción como si se tratara de un grupo, recibiendo el mismo valor tradicional de una organización que compra en gran volumen.

❑ **Distribuidor (*Distributor*)**

Una operación de tipo catálogo que está conectado a un gran número de productos fabricados para compradores al mayoreo y menudeo. Los corredores facilitan las transacciones entre distribuidores y la parte compradora. Para los compradores agiliza los tiempos de mercado al comprar en volumen así como reducir los costos. Para los distribuidores, este reduce los gastos de ventas, los procesos de orden del producto, transporte, etc.

❑ **Centro comercial virtual (*Virtual Mall*)**

Un sitio que ofrece varios mercados en línea. Un centro comercial común realiza cargos por acuerdo, y cuotas por transacciones. El modelo de centro comercial virtual puede ser el mas efectivo cuando se combina con un portal generalizado. Además, los centros

comerciales mas sofisticados proveen servicios de transacciones automatizadas y la oportunidad de relacionarse con otros mercados.

❑ **Metamediario (*Metamediary*)**

Un negocio que atrae compradores y comerciantes en línea al mismo tiempo y provee un servicio de transacciones tales como financiamiento y calidad de producto. Esto sería un centro comercial virtual, pero uno que realiza el proceso de transacciones, ordenes de envío y provee un servicio de cobro de cuentas. El *Metamediario* protege a los consumidores asegurando su satisfacción con el mercado. Los cargos del *Metamediario* son por cuotas por cada transacción realizada.

❑ **Corredores de subastas (*Auction Broker*)**

Un sitio que conduce vendedores por subasta. Los cargos del corredor son por venta, la cual es comúnmente en proporción al valor de la transacción. Los vendedores toman las grandes ofertas de los compradores arriba de su valor mínimo. Las subastas pueden variar sus términos de acuerdo a las reglas de oferta y demanda.

❑ **Subasta inversa (*Reverse Auction*)**

El modelo de negocio "*Ponga el precio*" también llamado "*Colección de demandas*" y "*Compras por pedido*". Los compradores prospecto hace una oferta final por un bien o un servicio específico y el corredor busca su cumplimiento. En algunos modelos, las ofertas de los corredores están entre el precio inicial y el precio esperado y quizás un precio mayor.

❑ **Clasificados (*Classifieds*)**

Una lista de objetos en compra o venta, comúnmente busca de un comprador o proveedor local. El precio puede o no estar especificado. Los cargos ocurren sin tener en cuenta si la transacción se completa.

❑ **Agente de búsqueda (*Search Agent*)**

Un agente es utilizado para buscar el mejor precio de un bien o servicio especificado por el comprador, o para localizar información difícil de encontrar. Una agencia de

empleos actua como agente de búsqueda, encontrando trabajo para los que busquen o encontrando gente para llenar una vacante de empleo.

- ❑ **Corredor de recompensas (*Bounty Broker*)**

El ofrecimiento de una recompensa por encontrar a una persona, objetos, ideas o alguna otra cosa, pero difícil de encontrar. El corredor establece los términos o un porcentaje de la recompensa si el objeto de búsqueda es finalmente encontrado.

Modelos de publicidad

Los modelos Web de publicidad es una extensión del tradicional modelo de difusión. Las difusoras, en este caso, un sitio Web, proveen contenido (común pero no necesariamente gratis) y servicios mezclados con mensajes publicitarios en forma de anuncios. Los anuncios pueden ser la mejor forma de obtener ingresos de una emisora. Una emisora crea su contenido o un distribuidor de contenido en otro lado. El modelo de publicidad solo funciona cuando el volumen del tráfico de observadores es amplio o altamente especializado.

- ❑ **Portal generalizado (*Generalized Portal*)**

Con un alto volumen de concurrencia, comúnmente 10 millones de visitas al mes, manejado por su genérica o diversidad de contenido o servicios. El gran volumen hace a la publicidad provechosa y permite ir mas allá de la diversificación de servicios del sitio, La competencia por la cantidad de visitantes ha ido hacia la distribución de contenidos y servicios gratuitos, tales como e-mail, tablas de mensajes, chat, noticias, información local, etc.

- ❑ **Portal personalizado (*Personalized Portal*)**

La naturaleza genérica de un portal de contenido múltiple es la lealtad de los usuarios. Esto se ha ido a la creación de portales que permitan la personalización de la interface y el contenido. Esto incrementa la lealtad a través de los mismos usuarios, cuando estos invierten tiempo en personalizar el sitio Las ganancias del portal son en base al volumen y al posible valor de la información que deriva de las preferencias del usuario. La personalización apoya al modelo de "Portal Especializado".

□ **Portal especializado (*Specilized Portal*)**

También llamado "Vortal" (i.e., Vertical Portal) aquí el volumen es menos importante. Es decir, un sitio que atrae solo a golosinas o constructores o nuevos padres, es altamente buscado después por ciertos publicistas que están buscando cierta audiencia en particular.

□ **Mercado de atención / Incentivo (*Attention/Incentive Marketing*)**

El modelo de "pago por atención", paga a los visitantes por ver el contenido y el llenado de formas, o boletas o más frecuentemente folletos. El mercado de atención hace su aprovechamiento de los usuarios con un complejo mensaje de productos de algunas compañías, el cual llega a ser del interés del cliente.

Modelo gratuito (*Free Model*)

Le da a los usuarios algo gratis: site hosting (almacenamiento), servicios Web, acceso a Internet, hardware gratis, tarjetas de agradecimiento electrónicas, Esto crea un gran volumen de oportunidades para la publicidad de este tipo de sitios. La viabilidad es difícil cuando la publicidad es el único medio de ingresos. Existe la posibilidad de mezclarlo con el modelo "Infomediary".

□ **Descuento de ofertas (*Bargain Discounter*)**

La venta de bienes comúnmente a un bajo costo y busca obtener sus ganancias a través de la publicidad.

□ **Informediary model (*Información de Segunda Mano*)**

La información acerca de los clientes y sus hábitos de compra son extremadamente valiosos. Especialmente cuando esa información es analizada cuidadosamente y enfocada hacia campañas de mercado. Algunas firmas fueron creadas con la única finalidad de coleccionar y vender información a otros negocios. Una *Informediary* puede ofrecer acceso Internet o hardware de manera gratuita a cambio de detallar información acerca de sus hábitos de navegación y compra. Este esta mas cerca al éxito que el que el uso Modelo de Publicidad.

❑ **Sistema recomendador (*Recommender System*)**

Es un sitio que permite a los usuarios intercambiar información acerca de la calidad de algún producto o servicio, o de los vendedores con quien ellos han tenido una experiencia de compra (buena o mala). El Sistema *Recomendador* toma ventaja del Modelo de Afiliación ofreciendo un mecanismo de aumento de ganancias por la venta de información de los clientes.

Modelo de registro (*Registration Model*)

El contenido del sitio es para ver libremente pero requiere que el usuario simplemente se registre (alguna otra información puede o no ser recolectada).

Modelo mercantil (*Merchant Model*)

Es típico para el vendedor al mayoreo y menudeo de bienes o servicios. Las ventas pueden estar hechas en base a un precio de lista o a través de una subasta. En algunos casos, el bien o servicio puede ser único en el Web.

Mercader virtual (*Virtual Merchant*).

Un negocio que opera solo en el Web y de manera tradicional un bien o servicio. El método de venta puede ser por medio de una lista de precios o una subasta.

❑ ***Bit vendor***

Este comerciante trata estrictamente con productos y servicios digitales en su forma más simple, maneja cualquiera de los dos tipos de ventas y distribución solamente por medio del Web.

Modelo de fabricante (*Manufacturer Model*)

Este modelo es, en el que esta basado el poder del Web para permitir fabricar y alcanzar a los compradores directamente. El modelo de fabricante esta basado en la eficiencia (ahorro en costos que puede o no ser transmitido al cliente), mejorando el servicio al cliente y un mejor entendimiento de las preferencias del cliente, los

productos perecederos tienen el beneficio de una distribución más rápida, se pueden demostrar las ventajas eliminando los intermediarios.

Modelo de Afiliación (*Model Affiliate*)

En contraste con el portal genérico, el cual busca manejar un gran volumen de tráfico en un solo sitio, el modelo de afiliación, provee oportunidades de compra a cualquier persona que pueda navegar. Esto se hace ofreciendo incentivos financieros para agregar a compañeros de sitio. A los afiliados se les otorgan "puntos de compra" en el mercado. Este es un modelo de "Pago por Rendimiento", si un afiliado no genera ventas, este no representa ningún costo al mercado. El modelo de afiliación es productivo, lo cual explica su popularidad. Sus variaciones incluyen, intercambio de anuncios, pago por click (pay-per-click), ganancias por compartir programas.

Modelo de comunidad (*Community Model*)

La viabilidad del Modelo de comunidad se basa en la lealtad de los usuarios (lo opuesto al alto volumen de tráfico). Los usuarios tienen una gran inversión tanto en tiempo como en diversión en el sitio. El modelo de comunidad puede también implicar una cuota de suscripción por obtener un mejor servicio.

Modelo de contribución voluntaria (*Voluntary Contributor Model*)

Muy parecido al tradicional modelo de emisoras públicas (Radio y TV), donde el radioescucha o el espectador contribuyen con la emisora. El modelo está dedicado a la creación de una comunidad de usuarios que apoyen el sitio a través de donaciones voluntarias. Las organizaciones sin fines de lucro pueden buscar su consolidación a partir de contribuciones de organizaciones de caridad y corporaciones que apoyen la misión de la organización.

Redes de conocimiento (*Knowledge Networks*)

También llamados "Sitio de Expertos", que provee una fuente de información basado en experiencias profesionales o de la experiencia de otros usuarios. Estos sitios son comúnmente manejados por foros de discusión donde las personas buscan información pudiendo hacer preguntas y recibir respuesta de alguien (que aparentemente) tiene conocimiento acerca del tema. Los expertos pueden ser empleados del sitio, un voluntario regular o en algunos casos, cualquiera persona en el Web que desea responder. Existen foros con temas mas especializados en los cuales es necesaria una cuota de inscripción, aunque por lo regular estos sitios son de uso comunitario para intercambiar información sin costo.

Modelo de suscripción (*Subscription Model*)

Los usuarios pagan para acceder al sitio. El gran valor del contenido es esencial. Esto ha probado ser viable para los puestos de periódicos, pero con poco éxito para el Modelo de Suscripción en el Web. En 1999 la *Jupiter Communications* encontró que el 46% de los usuarios de Internet no pagarían por ver el contenido de páginas Web. Algunos negocios lo han combinado con contenido gratuito, donde el contenido más importante solo es para los suscriptores.

Modelo de utilidad (*Utility Model*)

El Modelo de utilidad esta basado en cuanto al tiempo de su uso, pagando según se utilizó. El éxito depende de la habilidad de realizar los recargos por byte, incluyendo los micropagos.

1.3.2 Ventajas del comercio electrónico

- Permite a las empresas vender a consumidores que no pertenecen a su área de influencia. Abrir un sitio de comercio electrónico en Internet, pone a una empresa de inmediato en el mercado internacional. La existencia de avanzados

programas de software de traducción de lenguas hace accesible cualquier sitio a personas de diversas partes del mundo. Las compañías de mensajería ayudan a resolver complejos problemas de distribución internacional de comercio electrónico.

- ❑ Libertad de horarios. Esto se debe a que la limitante de los horarios de un comercio común esta sujeto a ciertos limites, esto se pierde en el Web, ya que ahí no se rige bajo ningún horario pues funciona las 24 horas del día los 365 días del año.
- ❑ Permite facilitar la compra a los clientes actuales logrando mayores ventas. La forma de pago mas común en los sitios de Internet es la tarjeta de crédito, evitando así el uso de dinero, todo se basa principalmente por medio de transacciones bancarias.
- ❑ Conocer mejor a los clientes y a recordar toda la información que den respecto a sus gustos, preferencias y hábitos de compra, mediante las bases de datos o sesiones.
- ❑ Permite ofrecer un servicio personalizado y a costos bajos.
- ❑ Permite competir con empresas grandes en igualdad de oportunidades, son compañías de todo tamaño las que están adoptando el comercio electrónico de negocio a negocio (B2B) para ahorrar cientos de millones de dólares por los costos más bajos y la reducción de inventario. El comercio electrónico de B2B abarca todos los aspectos de la cadena de suministros, desde la información del producto hasta el pedido, la factura, la requisición, el pago y el servicio al cliente. Es muy común percibir que las compañías de alta tecnología sean las primeras en adoptar el comercio electrónico.

Seguridad en Internet

En la actualidad existe la creciente duda de si las compras por Internet ofrecen la suficiente seguridad y prioridad para realizar esta actividad, se dice entonces que Internet es un canal seguro. La formas de pago son muy variadas, pero la más usual es el uso de la tarjeta de crédito, esto implica un flujo de información de suma importancia, ya que se manejan datos muy valiosos principalmente por la persona que

realiza el pago. Es por esto, que el proveedor debe garantizar todo intercambio de información donde se hace responsable de que los datos manejados serán tratados de manera confidencial. Los sistemas de seguridad que se implementan actualmente se pueden dividir en dos grupos: Canales Seguros de Comunicación y Sistemas de claves.

Canales Seguros de Comunicación

Son aquellos que agrupan un conjunto de protocolos que garantizan la confidencialidad y la integridad de las comunicaciones vía red. De entre los cuales destacan:

- a). HTTPS (HyperText Transfer Protocol Secure), que da soluciones de seguridad a las conexiones HTTP.
- b). El Protocolo SSL (Secure Socket Layer), diseñado e implementado por Netscape, que proporciona sesiones de comunicación encriptadas y autenticación del servidor.

□ HTTPS

Una vez que se entra a Internet para realizar compras, los navegadores regularmente o los sitios web envían un mensaje de que se va a entrar en un servidor seguro, (esto se comprueba cuando en la parte superior del navegador, la dirección empieza por https (Hipertext Transfer Protocol Secure), donde la "s" indica servidor seguro. A partir de ese momento, se ha ingresado a una página protegida por SSL o por SET.

Los protocolos SSL y SET son medios de encriptación de datos. Es decir: una vez que se ingresan los datos, nadie más los podrá interceptar, copiar o modificar. La mayoría de los bancos alojan sus pasarelas de pago en un "Servidor Seguro". Esto significa que la información confidencial (como un número de tarjeta de crédito) viajará encriptado a través de Internet, impidiendo la captación de ese dato por terceros ¹³.

□ SSL (Secure Sockets Layer)

Es un método de encriptación que tiene como fin establecer comunicaciones seguras. El encriptado de los datos, se realiza de forma automática por el navegador antes de

¹³ aece.org, Comercio Electrónico. [en línea] <http://www.aece.org/aece.asp> [10/02/2004]

que estos sean enviados al proveedor o en el caso de estar utilizando un TPV ¹⁴, a la Entidad Financiera. ¹⁵

❑ SET (Secure Electronic Transaction)

Conjunto de normas o especificaciones de seguridad desarrollado por VISA y MasterCard junto con empresas líderes en tecnología (Microsoft, Netscape, SAIC, GTE). Este protocolo asegura la confidencialidad e integridad de la información transmitida; identifica a las personas involucradas en la transacción (compradores, vendedores y Entidades Financieras); por último, ni el emisor ni el receptor pueden negar su participación en la transacción. Este servicio adjudica un certificado SET y almacena la información de la tarjeta, por lo que no es necesario teclear el número de la tarjeta en cada compra sino que el usuario sólo tiene que introducir una clave para obtener una protección SET completa en cada compra que realice. Las compras se realizan utilizando una tarjeta y es necesario disponer de un software específico para realizar las operaciones. ¹⁶

Este protocolo es un sistema de pago online: se necesita la autorización expresa del banco. Dicha autorización se transmite a través de una entidad intermediaria conocida como *pasarela*. Esta forma de pago está limitada a aquellas transacciones realizables con tarjetas de crédito.

La mayor parte de las pasarelas de pago evitan que el vendedor solicite directamente a sus clientes el número de tarjeta. El que lo solicita en realidad es el propio banco (a través de la aplicación de pasarela ubicada en los servidores seguros del banco). Lo único que recibe el vendedor es una notificación de si el pago ha sido o no correcto, pero no el número en sí.

En el pago SSL, el banco o caja del comprador puede requerir a la entidad que instala el Terminal Punto Venta Virtual el rechazo de la operación. Con este protocolo ninguna de las partes está obligada a aceptar la responsabilidad de pago, o de manera más simple, ambas partes aceptan la *irresponsabilidad* de aceptar un pago no firmado. Este es uno de los inconvenientes de SSL como forma de pago en Internet. La otra forma de pago virtual, SET, es un sistema de pago online: necesita la autorización expresa

¹⁴ Terminal Punto de Venta

¹⁵ aece.org, Comercio Electrónico. [en línea] <http://www.aece.org/aece.asp> [10/02/2004]

¹⁶ aece.org, Comercio Electrónico. [en línea] <http://www.aece.org/aece.asp> [10/02/2004]

del banco. Dicha autorización es tramitada a través de una entidad intermediaria (*pasarela*). Pero esta forma de pago está limitada a aquellas transacciones realizables solo con tarjetas de crédito.

El rechazo se puede llegar a dar un año después de haberse realizado la operación. En este caso, la entidad propietaria del TPV virtual cargará a su cliente los importes de estos rechazos. Depende del procedimiento que se use. En principio, siempre que la aplicación utilice una pasarela de pago de un banco, la transacción es segura. El servidor del banco se encarga de verificar que la tarjeta de crédito existe, que tiene fondos y realiza el ingreso correspondiente al importe de la transacción en el momento.

Sistemas de claves

Se trata de una de las aplicaciones más importantes ya que el usuario ejecuta una clave desde cualquier lugar de la red para obtener la seguridad en la transacción. Existen dos sistemas criptográficos que pueden ser: simétricos y asimétricos.

- ❑ **Encriptación simétrica:** obliga a los dos interlocutores (emisor y receptor) a utilizar la misma clave para encriptar y desencriptar el mismo código.
- ❑ **Encriptación asimétrica o criptografía** de claves públicas la cual está basada en el concepto de pares de claves, de forma tal que cada una de las claves encripta información que sólo la otra clave puede desencriptar. El par de claves se asocia a una sola persona, de forma que la clave privada solamente es conocida por su propietario mientras que la otra clave, (la pública) se publica ampliamente para que todos la conozcan. En este sentido cabe señalar que Estados Unidos ha permitido hasta hace poco la exportación de productos criptográficos que hicieran uso de claves de más de 40 bits, ahora sí permiten la exportación de dichos productos pero siempre que estos incluyan un sistema de recuperación de claves o de depósito de claves.

La incorporación por parte del protocolo SET parece ser la medida más segura hasta el momento. Es un sistema híbrido (utiliza ambos sistemas de encriptación) para evitar la lentitud de los sistemas de encriptación asimétricos y aprovechar la rapidez del sistema simétrico. Se utiliza además las firmas digitales y las entidades de certificación. Sin embargo esta siendo lenta su implantación

Como se puede ver, existen muchas formas de seguridad actualmente funcionando, quizás la más utilizada sea el SSL, que otorga una eficaz forma de encriptación de los números de la tarjeta. Sin embargo, la más eficaz es aquella que consigue la confidencialidad, autenticación, integridad y no el rechazo del mensaje, sólo conseguida por el momento por el protocolo SET.

Compras a Través del Móvil

El pago desde un móvil se realiza con cargo a una cuenta corriente o a la factura telefónica. La utilización de una forma u otra vendrá determinada por la localización de la aplicación del pago y por la entidad que gestione la operación más directamente.¹⁷

Las transacciones funcionan mediante autorización por código NIP (Número de Identificación Personal). Al realizarse la operación en la red telefónica, y no en Internet, la seguridad está garantizada. Funcionan con cualquier operadora de telefonía móvil.

Pasarela de Pagos

El procedimiento consiste en hacer una llamada a un archivo o URL proporcionada por el banco. Esa llamada se realiza con una serie de parámetros que permiten a la entidad identificar tanto al cliente como al pedido. Algunos de éstos parámetros son: identificador del comercio, referencia de la compra e importe total del pedido. Al finalizar la transacción, el banco informa al cliente (y al vendedor) del resultado de la transacción. Este proceso puede ser sencillo (información mostrada en pantalla al comprador) o algo más complejo (envío de correos electrónicos con el pedido, actualización de bases de datos de productos, etc.). Como cada aplicación de comercio electrónico tiene unas necesidades diferentes, los bancos suelen permitir al cliente personalizar el proceso de vuelta del banco.

¹⁷ aece.org, Comercio Electrónico. [en línea] <http://www.aece.org/aece.asp> [10/02/2004]

Terminal Punto de Venta Virtual (TPV)

Es un servicio financiero para negocios que, mediante la interconexión entre la página Web del comercio-cliente y el Servidor de pagos de cualquier banco, permite a los comerciantes realizar cobros por ventas pagadas con tarjetas de crédito. A pesar de la fiabilidad de la codificación y otras tecnologías, que son algunas veces superiores a las de otras redes de comunicaciones como las telefónicas, los medios no digitales dan una visión crítica sobre la seguridad de Internet.¹⁸

Debido a que el mercado electrónico sufre la falta de medidas de seguridad fundamentales disponibles en los mercados físicos. Cualquier participante en una transacción online nunca esta seguro sobre la identidad de la otra persona. Se necesitan tecnologías y legislación al respecto para corregir estos problemas

Sistemas de pago electrónico.

Existen diferentes sistemas de pago electrónico que se utilizan actualmente, de los cuales se pueden agrupar en tres grupos basándose en la información que se transmite online.

El primer tipo utiliza un tercero de confianza que mantiene toda la información sensible (como el número de cuenta y los números de la tarjeta de crédito) para sus clientes, lo que incluye tanto a vendedores como a compradores. Cuando se realiza una transacción, la información del pedido se transmite junto con la información sobre confirmación de pago, todo sin incluir información sensible. En realidad, no se realizan transacciones financieras online.

El segundo tipo es una extensión de la convencional transferencia de fondos. En las transacciones con tarjeta de crédito o cheques, la información sensible se intercambia, por ejemplo, proporciona la tarjeta de crédito a un comerciante, quien envía el número de tarjeta por la línea telefónica y espera confirmación. Los bancos mientras tanto reciben la misma información y ajustan las cuentas del vendedor y comprador. La información que se transmite online en este caso está encriptada por seguridad.

El tercer tipo incluye variantes del efectivo digital, dinero y monedas electrónicas. Lo que distingue a estos sistemas de los otros dos no es simplemente preservación de la

¹⁸ aece.org, Comercio Electrónico. [en línea] <http://www.aece.org/aece.asp> [10/02/2004]

identidad, sino el hecho de que lo que se transfiere es valor o dinero en si mismo. Con el segundo tipo descrito anteriormente, se podría cometer fraude espiando su mensaje (número de tarjeta de crédito) y haciendo compras con cargo a su tarjeta de crédito. Interceptar un mensaje es considerado como un robo de la propiedad, no sólo de información.

1.4 Páginas Web

Las páginas Web han evolucionado desde su concepción original, hoy en día han surgido nuevas necesidades y tecnologías para el continuo desarrollo de aplicaciones, con la finalidad de cumplir con los requerimientos que el mercado mundial ha estado demandando.

Una página Web es un documento escrito (generalmente) en lenguaje HTML ¹⁹, por medio de un navegador o *Browser* es posible visualizar este tipo de documentos. Un navegador es un programa que permite abrir páginas con contenido HTML.

Cada página Web tiene una dirección de Internet asociada a la misma, se le llama dirección electrónica o URL ²⁰. Una página Web puede incluir vínculos o ligas (*links*), estas ligas nos envían a otra dirección electrónica, otro determinado punto de la misma página, descargar archivos adjuntos, etc. Una página Web puede tener todo tipo de contenido multimedia (imágenes, audio, video, etc.), pero esto no quiere decir que esta información esta incluida dentro de la página, si no que cada archivo multimedia esta asociado a una dirección electrónica y cuando la página Web se esta cargando, va realizando las peticiones de cada archivo hasta que la página esta completamente cargada.

Las páginas se generan en el cliente o en el servidor, esto es que las páginas del cliente están escritas generalmente en lenguaje HTML (y sus diferentes variaciones) y se interpretan en el navegador o *browser* del cliente; las páginas de servidor están escritas en variados lenguajes de programación y estas al recibir una petición se ejecutan desde el servidor que es el que genera la página y la muestra en el

¹⁹ Hiper Text Marckup Languaje

²⁰ Uniform Resource Location

navegador del cliente. Las páginas Web según sus características, podrían catalogarse como páginas Web estáticas y páginas Web dinámicas (o de servidor).

1.4.1 Páginas Web estáticas

El término de páginas estáticas se emplea para definir el tipo de páginas Web cuyo contenido no puede ser cambiado por el usuario es decir que sólo está limitado a su visualización original. El contenido de la página permanece siendo el mismo, el usuario o cliente no puede interactuar con la página. Para que este tipo de páginas puedan variar, es necesario que el programador web realice los cambios. La tecnología para la creación de páginas Web por excelencia (tanto estáticas como dinámicas) desde los inicios del Internet es el HTML, cuya principal función es el formato de texto.

El HTML, fue el primer lenguaje que se creó para crear páginas Web. Una de sus principales ventajas es la simplicidad y sencillez de su funcionamiento, se basa en el uso de etiquetas para el formato de texto, es decir que el navegador Web lee las etiquetas y despliega el código interpretado en pantalla.

Un documento HTML se estructura en 3 partes:²¹

1. Una línea que contiene información de la versión del HTML.
2. Una declaración de cabecera delimitada por la etiqueta HEAD.
3. Un Cuerpo que lleva el contenido del documento. El cuerpo puede ser implementado con el elemento BODY o con el elemento FRAMESET (las cuales son etiquetas necesarias).

Ejemplo de un documento HTML:

²¹ Dave Raggett, Arnaud Le Hors, Ian Jacobs. HTML 4.01 Specification. [en línea], W3C Recommendation, 24 de diciembre 1999, disponible en WWW <http://www.w3.org/TR/REC-html40/>.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

<head><title> Mi Primer Documento HTML </title></head>

<body>

<p>Hola Mundo!

<br>



</body>

</html>
```

Este es un ejemplo de página web estática, donde se muestra una imagen y un mensaje, si de pronto se tuviera la necesidad de cambiar el mensaje desplegado o una imagen diferente, sería necesario modificar la página desde código. Los cambios de una página Web estática que no están contemplados, hacen que el desarrollador Web tenga la necesidad de realizar las modificaciones deseadas a nivel de código.

Un formulario es una sección de un documento que contiene elementos especiales llamados controles (checkboxes, radio buttons, menus, etc.) y etiquetas (Labels). Los usuarios generalmente "*Completan*" un formulario modificando sus controles (agregan texto, seleccionan opciones de menú, etc.) antes de enviar la información del formulario (a un servidor Web, a un correo electrónico, a una base de datos, etc.) y que ésta sea procesada. ²²

Si el formulario está escrito dentro de una página estrictamente en hecha con HTML, es muy común que los datos del formulario sean enviados una página de servidor o a una página Web dinámica para procesar la información enviada, estos datos son almacenados o sólo son procesados para un posterior. En ocasiones la información de un formulario es procesada dentro de la misma página HTML por un *script* escrito en

²² Dave Raggett, Arnaud Le Hors, Ian Jacobs. HTML 4.01 Specification. [en línea], W3C Recommendation, 24 de diciembre 1999, disponible en WWW <http://www.w3.org/TR/REC-html40/>.

algún otro lenguaje (muy comúnmente en JavaScript) desplegando los resultados en la misma página, es decir, sin ser enviados necesariamente a otra página.

Un formulario se compone, de lo que se llaman, "*Atributos*"²³. Los atributos son utilizados para definir el comportamiento de un formulario y para especificar el tipo de datos que se van a manejar. Los atributos de uso común son los siguientes:

action: Este atributo especifica donde se enviara y procesara la información del formulario. Este atributo especifica una dirección URL donde se encuentra ubicado el documento que recibirá los datos del formulario.

method: Este atributo especifica el método que será utilizado para enviar los datos del formulario. Los valores de uso común son el "*get*" (atributo por default) y el "*post*".

enctype: Este atributo especifica el tipo de contenido que será enviado en el formulario. El valor por default de este atributo es "*application/x-www-form-urlencoded*". El valor "*multipart/form-data*" debe ser usado en combinación con el elemento INPUT de tipo "*file*". Existe además el tipo "*text/plain*".

name: Este atributo sirve para asignarle al formulario un nombre como identificador.

Un formulario se escribe de la siguiente forma dentro del cuerpo (BODY) del HTML:

```
<form> .....</form>
```

```
<FORM> . ...</FORM>
```

```
<Form> ...</Form>
```

Se escriben tanto en mayúsculas, minúsculas o combinadas, siempre y cuando la etiqueta de apertura sea igual a la de cierre.

Los atributos se incluyen dentro de la etiqueta de apertura del formulario, Los atributos que tienen valores por *default* pueden omitirse sólo si realmente son estos los que se van a utilizar.

²³ Dave Raggett, Arnaud Le Hors, Ian Jacobs. HTML 4.01 Specification. [en línea], W3C Recommendation, 24 de diciembre 1999, disponible en WWW <http://www.w3.org/TR/REC-html40/>.


```
<form action="Registro.jsp" method="get" name="envios" enctype="application/x-www-form-urlencoded"
>
.....
</form>
```

En este ejemplo, el atributo **action** se refiere a una página Web dinámica JSP llamada "*Registro.jsp*", que será la que procese el contenido del formulario; el atributo **method** va a utilizar el método de envío "*get*"; el atributo **enctype** va a utilizar el tipo de datos como "*application/x-www-form-urlencoded*"; el nombre de nuestro formulario (**name**) se llamara "*envios*".

Cada uno de los controles del formulario tiene sus propios atributos al igual que un Formulario, los de uso común son:

name: el nombre que ayuda a identificar el componente.

value: el valor inicial del componente (excepto *textarea*).

Los controles de un formulario son objetos de los que se compone un formulario y son los siguientes:

Buttons

Existen 3 tipos de botones (**buttons**):

- ❑ **Submit buttons:** Cuando es presionado, todos los datos del formulario son enviados.
- ❑ **Reset buttons:** Cuando es presionado, todos los controles del formulario vuelven a su estado inicial.
- ❑ **Push buttons:** Este botón no tiene una acción por default. Cada *push button* tiene un evento asociado por el desarrollador, es decir es este último el que especifica la acción que se realizará al presionar ese tipo de botón (ej. puede existir una acción de botón para imprimir pantalla, abrir una ventana, etc.). La acción que se realiza, generalmente es atrapada por un script, el cual es el que realiza el evento.

```
<input type="submit" value="Enviar" name="smRegistro">
```

Checkboxes

Checkboxes y radio buttons son interruptores que son manipulados por el usuario. Un interruptor en "on" indica que el atributo esta seleccionado o "checked". Cuando un formulario es enviado, solo los elementos seleccionados o en "on" serán enviados. Varios Checkboxes pueden estar seleccionados a la vez.

```
<input type="checkbox" name="check1" value="1">
```

Radio buttons

El radio button es muy parecido a un Checkbox, a diferencia que en el radio solo se selecciona un solo elemento, son mutuamente excluyentes, es decir que si uno de ellos es seleccionado o puesto en "on", todos los demás son puestos en "off".

```
<input type="radio" name="sexo" value="m" >
```

```
<input type="radio" name="sexo" value="f" >
```

Menus

El menú ofrece varias opciones a los usuarios para escoger, El elemento SELECT crea el menú, en combinación con los elementos OPTGROUP y OPTION.

```
<select name="estadocivil">  
    <option value="1">Soltero</option>  
    <option value="2">Casado</option>  
    <option value="2">Divorciado</option>  
</select>
```

Text input

Se pueden crear dos tipos de controles que permitan al usuario ingresar texto. El elemento INPUT crea un control de entrada de texto de una sola línea, en cambio el elemento TEXTAREA crea un control de múltiples líneas.

```
<input type="text" name="nombre">
```

```
<textarea name="direccion"> </textarea>
```

File select

Este control, permite a los usuarios seleccionar archivos desde su propia plataforma que son enviados en conjunto con los datos del formulario.

```
<input type="file" name="archivo">
```

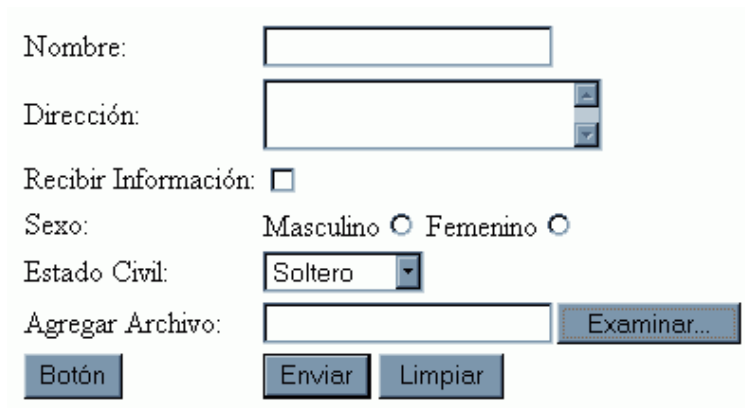
Hidden controls

Los desarrolladores crean controles que no son ingresados por los usuarios. Generalmente son utilizados para almacenar información entre el cliente/servidor que se pierden por la naturaleza del HTTP, es decir que pierde toda la información que se haya generado.

```
<input type="hidden" name="oculto" value="usuario">
```

Object Controls

Los desarrolladores pueden insertar objetos de uso general (imágenes, applets, etc.) en los formularios, de tal forma que dichos valores están asociados con otros controles.



A screenshot of a web form with the following elements:

- Nombre:
- Dirección:
- Recibir Información:
- Sexo: Masculino Femenino
- Estado Civil:
- Agregar Archivo:
- Botón:
- Enviar:
- Limpiar:

El objeto frente a "Nombre" es un *input* de tipo *text* y "Dirección" es de tipo *textarea*; "Recibir Información" es un *checkbox*; "Sexo" es un componente *radio button*; "Estado

Civil" es un objeto de tipo *select*, "Agregar Archivo" es un objeto *file select*, donde el boton "Examinar.." es parte de este componente, Los de tipo Buttons son tres "Botón", "Enviar y "Limpiar", el primero (de tipo *Button*) realiza una acción definida por el desarrollador, el segundo (de tipo *Submit*) envía los datos del formulario hacia una dirección definida en el formulario y el último (de tipo *Reset*) sirve para volver todos los componentes a su estado inicial.

Scripts

Un *script* es un programa que acompaña a un documento HTML o página Web que está incluido en el. El programa se ejecuta en la máquina del cliente cuando se carga el documento, o en algún otro momento, como por ejemplo cuando se activa un vínculo.²⁴

Los scripts ofrecen a los autores la posibilidad de extender los documentos HTML de maneras activas e interactivas:

- ❑ Pueden evaluarse los scripts a medida que se carga el documento para modificar los contenidos del documento dinámicamente.
- ❑ Los scripts pueden acompañar a un formulario para procesar los datos a medida que éstos se introducen. Los diseñadores pueden rellenar dinámicamente partes de un formulario según los valores de los otros campos, también pueden asegurarse de que los datos introducidos concuerden con rangos de valores predeterminados, de que los campos sean consistentes entre sí (también conocido como validación de los datos), etc.
- ❑ Los scripts pueden ser llamados por eventos que afecten al documento, como la carga, la descarga, el movimiento del foco sobre los elementos, los movimientos del ratón, etc.
- ❑ Los scripts pueden ser vinculados a controles de formulario (Ej., botones) para producir elementos gráficos para la interfaz del usuario.

Hay dos tipos de scripts que los autores asocian a un documento HTML:

²⁴ Dave Raggett, Arnaud Le Hors, Ian Jacobs. HTML 4.01 Specification. [en línea], W3C Recommendation, 24 de diciembre 1999, disponible en WWW <http://www.w3.org/TR/REC-html40/>.

- ❑ Aquellos que se ejecutan una sola vez cuando el agente de usuario carga el documento. Los scripts que aparecen dentro de un elemento SCRIPT se ejecutan cuando el elemento es cargado. Para los agentes de usuario que no puedan o que no vayan a ejecutar scripts, los autores pueden incluir contenido alternativo por medio del elemento NOSCRIPT.
- ❑ Aquellos que son ejecutados cada vez que ocurre un determinado evento. Estos scripts pueden ser asignados a varios elementos por medio de los atributos de eventos intrínsecos.

1.4.2 Páginas Web Dinámicas

Una página Web dinámica son generada de dos formas distintas.

- ❑ La página Web modifica su contenido o apariencia mediante funciones ya agregadas dentro del HTML. La página Web ya esta creada y almacenada en el servidor, esta página al ser solicitada, es enviada tal y como se encuentra en el servidor. Por ejemplo el cambiar el color del fondo de la página o el tipo de letra, crear calculadoras en línea o funciones que realizan ciertas operaciones matemáticas, etc. Cuando algún evento del usuario es atrapado, una función interna en el HTML se ejecuta y realiza la acción predeterminada. En este tipo de páginas dinámicas, es muy común utilizar la tecnología de JavaScript.
- ❑ La página Web es generada dentro del servidor, esto es, cuando una página Web es solicitada al servidor, corre la aplicación de ejecución de la página (se ejecuta el servicio que se encarga de compilar la página) y se compila y ejecuta cualquier sentencia que sólo reconoce el servidor y crea el HTML de la página Web. La página es generada en su totalidad con tecnología de Servidor (la aplicación genera todo el HTML) o estar dentro del HTML, es decir que hay tecnologías que se pueden insertar y mezclar dentro del HTML (existe código HTML junto con scripts escritos en otro lenguaje en el mismo documento), para que sólo una parte del código lo genere el servidor. Por ejemplo, una página Web de venta de libros, donde el usuario ingresa el titulo que desea adquirir, la página Web procesa la información del usuario y el servidor compila la página con los parámetros, creando así una página Web nueva a partir de la

información que fue ingresada. La página que se muestra, se crea a partir de la información que proporciona el usuario y dependiendo los parámetros que se registran, la página se va creando de acuerdo a la solicitud del cliente. En este tipo de páginas, es muy común verlas trabajar en conjunto con bases de datos (de muy diversas clases), donde generalmente (pero no siempre) obtienen los datos para crear el contenido dinámico. Este tipo de página dinámica se crea con diversas tecnologías de servidor, tales como ASP, PHP, JSP, etc. Aunque también pueden incluir JavaScript junto con los scripts de otro lenguaje. Al tratar de ver el código fuente de la página HTML no será posible encontrar código fuente del lenguaje que se utilizó para la creación de la página, ya que este lenguaje es compilado y se traduce a código enteramente HTML.

Existe una gran variedad de tecnologías para la creación de páginas Web dinámicas entre las cuales, las más importantes y destacadas son:

- ❑ JavaScript.
- ❑ CGI (Common Gateway Interface).
- ❑ ASP (Active Server Pages de Microsoft).
- ❑ PHP (Personal Home Pages).
- ❑ JSP (JavaServer Pages de Sun Microsystems).

1.4.3 JavaScript

JavaScript es un lenguaje pequeño y ligero; no es útil como lenguaje tipo *standalone*, pero fue diseñado para ser insertado dentro de otros productos y aplicaciones, tales como Navegadores Web o Browser. Dentro de un ambiente Host, JavaScript puede estar conectado a los objetos de su entorno para proporcionar un control programado sobre ellos. ²⁵

El núcleo de JavaScript (Core JavaScript) contiene un set de objetos, tales como arreglos, fechas, funciones matemáticas y un set de elementos como operadores (+, -, /, *), estructuras de control (if, else, for, while), y declaraciones. El núcleo de

²⁵ JavaScript Reference, [en línea], Netscape Communications Corporation, 12 de diciembre de 1997, disponible en WWW <http://www.lecad.uni-lj.si/documents/vaje/js-tecaj/js/jsref.pdf>

JavaScript se extiende a una variedad de propósitos complementándolo con objetos adicionales.

□ **JavaScript Client-Side**

De lado del cliente (Client-Side) JavaScript se extiende al núcleo del lenguaje proporcionando objetos para el control del Browser y el Document Object Model (DOM)²². Por ejemplo, las extensiones Client-Side permiten a una aplicación colocar elementos en un formulario HTML y responder a los eventos del usuario tales como botones del mouse, entrada del formulario, páginas de navegación, etc.

JavaScript Client-Side abarca núcleo (Core) del idioma además de objetos predefinidos solo relevantes en un navegador. JavaScript Server-Side abarca el mismo núcleo del lenguaje más los objetos extra y funciones que se ejecutan solo en el JavaScript de servidor.

JavaScript Client-Side es insertado directamente en una página HTML y es interpretado completamente por el browser en tiempo de ejecución. Por la facilidad en la producción de aplicaciones frecuentes tiene una gran popularidad por su rendimiento, las aplicaciones de JavaScript aprovechan sus capacidades Server-Side que son compiladas antes de ser desplegadas. Navegadores Web tales como Netscape Navigator e Internet Explorer en sus versiones más actuales (aunque hay variantes de JavaScript que sólo se ejecutan en un navegador en particular, es decir que existen funciones que sólo se ejecutan en Netscape y otras únicamente en Explorer, aunque comúnmente existen equivalencias de las mismas funciones pero con diferentes sentencias), puede interpretar JavaScript Client-Side insertado dentro de una página de HTML.

Las declaraciones JavaScript Client-Side insertadas en una página HTML pueden responder a eventos del usuario, tales como botones de mouse, entrada al formulario y navegación de la página. Por ejemplo, se escribe una función en JavaScript para verificar que los usuarios ingresan información válida al Server-Side dentro de un formulario que pide un número telefónico o un código postal, dado que la información es de tipo numérica, se puede validar que no se ingresen caracteres diferentes a números. Sin ninguna transmisión de la red, el JavaScript incluido en la página HTML puede verificar los datos insertados y desplegar un mensaje al usuario que ingresó información inválida.

□ JavaScript Server-Side

De lado del Servidor (Server-Side) JavaScript se extiende al núcleo del lenguaje proporcionando objetos para ejecutar JavaScript en un servidor. Por ejemplo, las extensiones del Server-Side le permiten a una aplicación comunicarse con una base de datos de diferentes distribuidores, compartir información con los usuarios a través de una aplicación, acceder a archivos del sistema en el servidor, proporcionar continuidad de información de una invocación a otra de la aplicación o realizar manipulaciones de archivos en el servidor. En el servidor, JavaScript es también incluido en páginas HTML. Las declaraciones Server-Side pueden conectarse a Bases de Datos relacionales de

En contraste con los Scripts JavaScript Client-Side, las aplicaciones JavaScript que se usan en el JavaScript Server-Side son compilados dentro de archivos ejecutables llamados bytecodes. Esas aplicaciones se ejecutan junto con el servidor Web que contiene el JavaScript en tiempo de ejecución. Por esta razón, la creación de aplicaciones JavaScript es un proceso de dos fases.

- En la primera fase, el desarrollador crea una página HTML (la cual contiene sentencias JavaScript Client-Side y Server-Side) y archivos de JavaScript. Se pueden compilar todos esos archivos dentro de un sólo ejecutable.
- En la segunda fase, una página en la aplicación es requerida por el browser de un cliente. El *engine* de tiempo de ejecución (esto es un servicio que se ejecuta en el Servidor que se encarga de generar las páginas dinámicas) utiliza la aplicación ejecutable para mostrar el código fuente y generar dinámicamente la página HTML. Esto ejecuta cualquier sentencia JavaScript Server-Side encontrada en la página. El resultado de todas las declaraciones pueden agregar nuevo HTML o las declaraciones JavaScript Client-Side a la página HTML. Esto entonces envía la página resultante por la red hacia el Navegador del cliente, el cual despliega los resultados.

Básicamente así es como funciona una página de servidor, donde es el servidor el que crea la página Web según es solicitada (como se mencionó anteriormente).

Netscape creó JavaScript, el cual fue usado inicialmente en sus navegadores.

El código JavaScript se incluye entre etiquetas especiales de comienzo y final:

```
<script language="JavaScript"> ....código... </script>
```

ó

```
<script> ....código... </script>
```

Para ejecutar un Script en JavaScript sólo se necesita de un navegador. JavaScript no tiene capacidades para la manipulación de gráficos, aplicaciones para redes, capacidad para lectura / escritura de archivos en el lado del cliente (excepto cookies), se usa por lo regular para hacer validaciones de datos previas al envío o procesamiento de los mismos.

□ **Document Object Model (DOM)**

El Document Object Model (DOM) se define como una plataforma y lenguaje neutral, una interface que permite a los programas y Scripts acceder y actualizar dinámicamente el contenido y estructura de un documento. El DOM fue creado para tener acceso al núcleo de las interfaces y manipular la estructura y el contenido de un documento Web.²⁶

El DOM es un API (Application Programming Interface) para validar documentos HTML. Esto define la estructura lógica del documento en la forma en que el documento es accesado y manipulado. Con el DOM, los programadores crean documentos, estructuras de navegación, agregar, modificar o eliminar elementos y contenido. Cualquier cosa que se encuentre en un documento HTML puede ser accesado, cambiado, borrado o agregado utilizando el Document Object Model.

En un modelo abstracto de datos, el modelo se centra alrededor del dato, en los lenguajes de programación orientados a objetos, el dato en si es encapsulado en un objeto donde se esconde el dato, protegiéndolo de cualquier manipulación directa externa. Las funciones asociadas con esos objetos, determinan como los objetos pueden ser manipulados y así ellos forman parte del modelo de objetos. El DOM se origina como unas especificaciones que permiten a los programas en JavaScript y Java

²⁶ Arnaud Le Hors, Philippe Le Hégarret, Lauren Wood, Gavin Nicol, Jonathan Robie, Mike Champion, Steve Byrne. Document Object Model (DOM) Level 3 Core Specification. [en línea] W3C Recommendation, 7 Abril 2004. <http://www.w3.org/TR/DOM-Level-3-Core/>

ser portables entre navegadores Web. El HTML dinámico fue antecesor del DOM y fue pensado originalmente en términos para navegadores, es decir que estos no tengan problemas de incompatibilidad con estos lenguajes.

1.4.4 Common Gateway Interface (CGI)

El Common Gateway Interface (CGI) no es estrictamente un lenguaje sino un método para la creación de páginas dinámicas, de hecho se puede crear un CGI con varios lenguajes de programación (C, C++ y Perl). CGI fue la primera respuesta al problema de generar contenido dinámico a las páginas Web, permitió a los desarrolladores a utilizar pequeños programas que literalmente creaban nuevas páginas Web al instante. Con este avance, podía existir del todo una página estática ya que la escritura del CGI permitía generar HTML dinámico.

Los Scripts CGI necesitaban ser escritos en un lenguaje de programación y los más populares fueron C, C++ y Perl. Los servidores Web podían pasar a los programas toda la información disponible incluyendo las variables de entorno o entradas estándar. El programa genera la petición (*response*) en HTML y los datos puedan ser incluidos dentro del código.

Las conexiones con las bases de datos se hacían agregando SQL en los programas de C++ o bibliotecas para los Scripts del lenguaje. Estos programas tenían algo muy similar a aplicaciones con SQL agregado, a excepción de que esos resultados debían ser en HTML, en charts o reportes. Las ventajas del CGI es que era una tecnología ampliamente soportada, además de que todos los servidores Web soportaban CGI y casi todas las bases de datos soportaban SQL insertado en programas de C o C++. La desventaja del sistema CGI es que el sistema no era del todo portable, los programas tenían que ser modificados para ser movidos o traspasados a otra maquina o si otra base de datos era usada. El problema con estos lenguajes es que no eran del todo sencillos, en ocasiones era demasiado frustrante incluso para imprimir una fecha. El sistema CGI tampoco tomaba las ventajas sobre el servidor Web. Finalmente cada llamada a un programa CGI era un proceso distinto, por lo cual incremento el número de procesos que se ejecutaban en el servidor, lo cual hacia disminuir el rendimiento.

Uno de los inconvenientes del CGI, es que cada vez que se hace una petición HTTP, se realiza un nuevo proceso, si hay n peticiones del programa CGI, el código del programa se carga la misma cantidad de veces en memoria. No tiene portabilidad, si se desea trasportar un CGI a otra máquina o cambiar la base de datos, el programa puede no funcionar o trabajar incorrectamente. Un CGI esta construido en un lenguaje de programación que puede no estar estandarizado, lo que provoca incompatibilidades con el sistema.

1.4.5 Active Server Pages (ASP)

En 1996 surgió de Microsoft un lenguaje de programación para la creación de páginas Web dinámicas llamado Denali (el nombre de un parque nacional de Alaska), hecho exclusivamente para generar páginas dinámicas. Esta nueva tecnología se basó en el lenguaje de Microsoft Visual Basic Script (VBScript) en lugar de C o Perl.

Utilizando el Visual Basic en lugar de otro lenguaje, Microsoft esperó introducirse para que, de una manera más amigable, obtener contenido dinámico, que, a su vez, permitiría a más personas usar y crear páginas Web dinámicas. Denali también vino con Internet Information Server (IIS), un producto de Microsoft para Servidores Web, sin ningún costo al cliente. Esto estaba en contraste con otras líneas del producto compitiendo con ColdFusion de Macromedia, donde Microsoft esperaba obtener mas alcance al no poner un costo a su producto.

Los usuarios en general aceptaron bastante bien a Denali y a finales de 1996 fue cambiado y mejorado para ser nombrado como Active Server Pages. El Active Server Pages es un lenguaje interpretado, es decir no se compila y se puede combinar HTML, scripts y componentes ActiveX reutilizables del servidor para crear aplicaciones dinámicas. Las Active Server Pages permiten Scripts de tipo Server-Side mediante el Internet Information Server (IIS) con apoyo nativo de VBScript y JScript.

Las Active Server Pages (ASP's) son páginas Web que contienen Scripts de tipo Server-Side además de la usual mezcla de texto y etiquetas HTML. Los scripts tipo Server-Side son comandos especiales que se insertan en páginas Web que son procesadas antes de que las páginas sean enviadas por el servidor al Browser de algún usuario

que está visitando el website. Cuando se hace la petición (request) de un archivo HTML normal, el Browser lo despliega como una combinación de texto, imágenes, y sonidos.

Antes de que el servidor envíe la Active Server Page al Browser, se ejecutan todos los scripts Server-Side contenidos en la página. El código ASP se incluye entre etiquetas especiales de comienzo y final:

```
<% ...código... %>
```

Se crea un archivo con extensión **".asp"** y el servidor reconocerá la aplicación y la compilará para crear un archivo con código HTML. Los Scripts creados en ASP están basados en el lenguaje VBScript (Visual Basic Script), JScript o JavaScript, una página creada en tecnología ASP tiene el principal inconveniente de que sólo se ejecuta en el sistema operativo de Windows (NT, 2000, XP). Hay que pagar el costo de una licencia por el *kit* de herramientas de desarrollo de esta tecnología.

1.4.6 PHP (Personal Home Pages)

PHP es un lenguaje de script que permite escribir páginas Web que contienen código insertado que se ejecuta cuando se accede la página y que genera contenido dinámico para ser incluido como parte de la salida enviada al navegador Web del cliente. 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 sólo 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.²⁷

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" (PHP), el "Intérprete de Formularios" (FI), añadió soporte para base de

²⁷ Stig Sether Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski, Andrei Zmievski, Jouni Ahto, Rafael Martínez. Manual de PHP. Grupo de documentación de PHP. redistribuido bajo los términos de la "GNU General Public License". 15 de abril de 2001.

datos como mSQL. PHP/FI creció rápidamente y la gente empezó a contribuir en el código.

A mediados de 1997 el desarrollo del proyecto sufrió un 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 reutilizado para PHP3 y otra gran cantidad fue escrito completamente desde cero.

PHP es un lenguaje interpretado de alto nivel insertado en páginas HTML y ejecutado por el servidor. La mayoría de su sintaxis es similar a C, Java y Perl, con algunas características específicas propias de PHP.

No es lo mismo un script CGI escrito en su totalidad con otro lenguaje de programación como Perl o C, en vez de escribir un programa con muchos comandos para crear una salida en HTML, escribimos el código HTML con cierto código PHP insertado en el mismo, que producirá una salida.

El código PHP es ejecutado en el servidor. El cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor Web es incluso configurado para que procese todos los ficheros HTML con PHP.

Los nombres de los archivos para los Scripts PHP están escritos con una extensión que permite al servidor Web reconocer y ejecutar al interprete PHP para procesarlo, la extensión es **".php"** (anteriormente se manejaban **".php3"** y **".phtml"**).

El código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP. Existen 4 formas distintas:

- ◆ **<?php ...código... ?>**
- ◆ **<? ...código...?>**
- ◆ **<% ...código...%>** (compatible con ASP)
- ◆ **<script language="php">...código... </script>**

Este lenguaje esta basado en la inserción de Scripts dentro del HTML, es decir que es lenguaje de programación por Scripts.

2. JavaServer Pages (JSP)

2.1 Java

Para comprender mejor el funcionamiento de una página JSP es conveniente tratar de entenderlo desde su origen, que es Java. Se explicarán los conceptos básicos del lenguaje, así como todas sus características, beneficios, ventajas, limitantes, etc.

Para explicar el por que JSP es una tecnología que ofrece todo lo necesario para solucionar cualquier problema que implique la necesidad de crear páginas Web dinámicas, es indispensable conocer las bases del lenguaje y exportarlo hacia el Web.

Historia de Java

Sun Microsystems financió un proyecto de investigación corporativo interno con el nombre código de Green en 1991. Como resultado del proyecto fue la creación de un lenguaje basado en C y C++, que su creador, James Gosling llamo Oak, inspirado por un roble (oak en ingles) que crecía fuera de su ventana en Sun. Más tarde se descubrió que ya existía un lenguaje de programación llamado Oak. Cuando un grupo de empleados de Sun, visitó una cafetería local, alguien sugirió el nombre Java (como se le conoce también en ingles al café) y así fue como quedo nombrado.¹

El proyecto inicialmente estaba orientado a la creación de un lenguaje de programación para dispositivos electrónicos, pero este mercado no estaba teniendo el éxito esperado en contraste a con la popularidad de la World Wide Web que hizo explosión en 1993 y la gente de Sun detecto el potencial inmediato de Java para la creación de páginas Web con "contenido dinámico".

Sun anunció formalmente Java en una presentación en mayo de 1995. Java hizo surgir un interés inmediato en la comunidad de los negocios en vista de la magnitud del interés comercial por la World Wide Web. Java no es un lenguaje académico como Pascal o un lenguaje diseñado por una sola persona o un grupo reducido para su propio uso local como C o C++. Más bien Java había sido diseñado con motivos comerciales y genero un interés avasallador en la comunidad de los negocios a causa de otro avance relacionado con Internet.

¹ Deitel, H.M., Deitel, P.J. (1998). Como programar en Java. México: Pearson Educación. Segunda ed.

2.1.1 Clases y Métodos

Java es un lenguaje orientado a objetos, para poder desarrollar una aplicación, es necesario construir una Clase. Como un objeto de la realidad, este tipo de objeto posee una serie de características propias que los diferencian de los demás Objetos y Clases. Por ejemplo, podemos definir una Clase Automóvil; nuestro Automóvil se puede dividir en Compactos, Deportivos, Carreras, Exóticos, etc. Cada uno de nuestros Autos puede ser de velocidades tipo Estándar o Automático. Nuestra Clase Auto se compone de Métodos, donde cada Método tiene una característica única dentro de la Clase, (Fig 2.1.1).

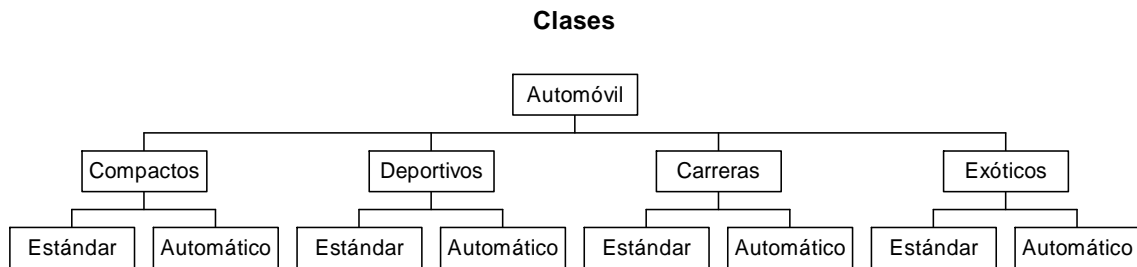


Fig 2.1.1

Nuestra Clase Automóvil posee los siguientes métodos que son: Compactos, Deportivos, Carreras, Exóticos; donde los valores de nuestros Métodos pueden ser Estándar o Automático.

En la programación en Java, se dice que una Clase está compuesta de Métodos y estos a su vez, contienen valores. Las clases y los métodos son Públicos o Privados. Cuando un Método o una Variable es declarada como Privada, ningún Método externo puede acceder a ella, a diferencia de la declaración Pública. Java posee ciertas características que lo hacen un lenguaje de programación muy útil para el desarrollo de aplicaciones para el Web y para resolver problemas del mundo real.

Entre las características y ventajas de Java se encuentran: Encapsulación, Herencia, Polimorfismo, Portabilidad, Seguridad, Estabilidad, Memoria, Multihilos.

2.1.2 Encapsulación

Cuando una clase es declarada como Privada (`private`), todos los Métodos y Variables se protegen o encapsulan para que ninguna clase o Método externo accese a ellos.

2.1.3 Herencia

La herencia es una forma de reutilización del Software, cuando una Clase *hereda* de otra (llamada *Superclase*), la primera obtiene todas las características, propiedades y métodos de la *Superclase*. Esto es muy útil, cuando se necesita la implementación de un método ya existente, simplemente se hereda de él y evita reescritura del código. Ej. Cuando ya existe una clase para obtener el área de un triángulo y deseamos reutilizar esta clase, simplemente la heredamos construyendo un objeto a partir de esta.

El Objeto *triangulo* ya tiene todos los métodos y propiedades del objeto *Triangulo*. Evitando así la reescritura de código. De aquí que la gran cantidad de clases que es posible implementar.

2.1.4 Polimorfismo

El Polimorfismo es la escritura de Métodos con nombre identificador igual, pero con la recepción de variables y/o la cantidad de ellas es distinta. Ej, se crean 2 métodos Suma, ambos Métodos reciben 2 variables, pero la diferencia es que una recibe enteros y la otra flotantes. Esta característica se implementó por primera vez en el lenguaje de programación C++. El sistema identifica de manera automática que método aplicar reconociendo el tipo de variables que entran siendo enviadas.

2.1.5 Portabilidad

Una de las características más notables del lenguaje es su lema "Write Once, Run Anywhere", esto quiere decir, que después de compilar una aplicación de Java (se crea una clase con métodos), esta se ejecuta en cualquier plataforma (Windows, Unix, OS, etc.). Cuando se necesita exportar una aplicación de Java de una plataforma a otra, no se necesita hacer modificación alguna a la aplicación ni se necesitan instalaciones de software extra.

2.1.6 Seguridad

Java es uno de los lenguajes de programación mas seguros, por ejemplo un applet² no tiene acceso a los recursos (en caso de que la aplicación sea un applet) del usuario o cliente, así que cuando se ejecuta un applet del lado del cliente, se tiene la certeza de que no tendrá contenido dañino para el sistema.

2.1.7 Estabilidad

Cuando se ejecuta una aplicación, sea de cualquier lenguaje, cabe la posibilidad que el programa tenga un error en tiempo de ejecución (ej. división sobre cero, valores nulos, etc.), terminando prematuramente con la ejecución del sistema. Java permite atrapar este error y continuar con el funcionamiento del programa, a esto se le conoce como manejo o control de Excepciones.

2.1.8 Memoria

Java se ejecuta en la JVM (Java Virtual Machine) donde son interpretados los bytecodes, aquí se asigna la memoria para correr las aplicaciones. Java elimina la utilización de punteros, (como su nombre lo indica, un puntero o apuntador, apunta a un segmento de la memoria) evitando así un posible problema de corrupción de memoria, ya que el apuntador sigue utilizando memoria si esta no es liberada. Para esto existe el recolector de basura de Java, que libera automáticamente los recursos de la memoria que ya no se utilizan. Además que una aplicación en Java ocupa un solo segmento de la memoria al ejecutarse en la JVM.

2.1.9 Multihilos

Cuando múltiples aplicaciones se ejecutan al mismo tiempo, estas aplicaciones se ejecutan una tras de otra, es decir al inicio se ejecuta la primera, después la segunda y así sucesivamente hasta terminar de ejecutar todas las aplicaciones. A esto se le llama *ejecución de aplicaciones de un solo hilo*. Java posee la implementación de Multihilos (*Multithreads*). Esto es, que se pueden crear aplicaciones que se ejecuten simultáneamente.

² 2.2 Applets

Ej. Si se ejecutan tres aplicaciones al mismo tiempo: una aplicación de calculadora, después una aplicación para la administración de correo electrónico y al final una aplicación para editar texto. Normalmente se ejecutarían en el mismo orden en que fueron llamadas, pero con la implementación de Multihilos, se ejecutan primero las aplicaciones de mayor prioridad o importancia, (primero el administrador de correo electrónico, después el editor de texto y al final la calculadora). Esto funciona de la siguiente manera, cada aplicación puede tener una prioridad definida, la aplicación con mayor prioridad se ejecuta primero que las de menor prioridad, pero no impidiendo su ejecución, sino que se van ejecutando simultáneamente, es decir que la aplicación de mayor prioridad utiliza la mayor parte de los recursos de sistema mientras se va ejecutando, dejando recursos mas limitados a las aplicaciones de menor prioridad. Cuando la aplicación de mayor prioridad se termina de ejecutar, libera los recursos para terminar de ejecutar las demás aplicaciones.

Aunque Java es portátil, ciertas partes del lenguaje están ligadas a la plataforma, en el sistema operativo Windows por ejemplo, los hilos se manejan por un cierto tiempo de uso ya definido en el sistema operativo, por lo tanto el sistema de multihilos se comporta de maneras diferentes dependiendo la plataforma en que se esté ejecutando la aplicación. ¹⁰

2.2 Applets

Los applets son aplicaciones o programas de Java que se ejecutan en un navegador Web. Los applets se cargan dentro de una página Web que se ejecuta en la JVM (la JVM debe estar habilitada en el navegador Web para poder ejecutar el applet).

Ciclo de Vida de un Applet

Un applet tiene un ciclo de vida que consiste en los siguientes métodos: `init`, `start`, `paint`, `stop` y `destroy`.

init .- Este método se utiliza siempre que un applet necesita inicializarse (volver a sus valores iniciales). Este método se invoca automáticamente cuando el applet se ejecuta por primera vez.

start .- Este método se ejecuta después de que el método `init` es invocado. Todas las operaciones iniciales se ejecutan dentro de este método. Este método se ejecuta cada vez que se regresa a la página HTML que contiene el applet.

paint .- Este método se ejecuta cada vez que la clase `Graphics` es invocada. Cada vez que la página HTML pasa a primer plano, este método se encarga de volver a pintar o dibujar el applet, cada vez que lo necesite (por ejemplo si la página Web pasa a segundo plano y luego se pasa a primer plano, el método *paint* se vuelve a ejecutar).

stop .- Este método se invoca cuando el usuario abandona la página Web que contiene el applet. Este método suspende todas las operaciones que se estén ejecutando dentro del applet.

destroy .- Este método se ejecuta cuando el navegador que contiene la página Web con el applet se cierra, se encarga de terminar todas las operaciones realizadas y liberar todos los recursos que se hayan destinado para el applet.

El applet fue de las primeras tecnologías para crear aplicaciones para el Web. Entre sus beneficios se encuentran:

- ❑ Son seguros, un applet no tiene acceso a los recursos del cliente o usuario.
- ❑ Una gran colección de herramientas para poder enriquecer el contenido Web: manipulación y visualización de imágenes, reproducir audio y video, gráficos, eventos del usuario, etc.
- ❑ Su ejecución solo requiere de un navegador Web.
- ❑ La capacidad de poder crear programas reales (aunque limitados) para el Web.

Los applets tienen una serie de limitaciones:

- ❑ El tiempo de ejecución es lento, más aun si se cuenta con un equipo de cómputo con recursos limitados.
- ❑ El tiempo de descarga ocasionalmente es prolongado si no se cuenta con una conexión de alto desempeño.

- ❑ Crear aplicaciones que tengan accesos a bases de datos o algún otro recurso del servidor, es laborioso.

Para cargar un applet dentro de un documento HTML se insertan las siguientes etiquetas:

- ❑ Etiqueta desaprobadada³ (Puede no ejecutarse en todos los navegadores Web):

```
<applet code="Clase.class" width="100" height="50">  
</applet>
```

- ❑ Etiqueta recomendada:

```
<OBJECT codetype="application/Java" classid="Clase.class"  
        width="100" height="50">  
</OBJECT>
```

Donde "Clase.class" es nuestra aplicación, "width" es el ancho y "height" la altura que se va a definir para el applet.

Años mas tarde comenzaron a surgir nuevas tecnologías para la creación de páginas Web dinámicas más eficientes, rápidas y con acceso rápido a recursos del servidor, los applets no podían competir con esto, así que se creó una nueva tecnología de servidor para la creación de páginas Web dinámicas: Los Servlets.

2.3 Servlets

Los Servlets se crearon en 1997, implementando Java (junto con todas sus ventajas) para la creación de páginas Web dinámicas. Esto resulto ser un gran avance, ya que se podía crear contenido dinámico en Internet con el respaldo Java.

³ Desaprobada o *deprecated* indica que existe una versión nueva de la etiqueta y que es probable que no se reconozca en versiones superiores de navegadores.

Los applets tenían una desventaja, ya que se ejecutan en la JVM del lado del cliente, esto daba origen a ciertos problemas de incompatibilidad en los navegadores, los Servlets corrigen estas dificultades al ejecutarse en la maquina virtual del Servidor, esto quiere decir que el servidor ejecuta la aplicación y la despliega en el navegador web del cliente, olvidándose de la incompatibilidad con los navegadores incluso de versiones antiguas.¹²

Un Servlet es una aplicación de Java, tiene acceso a todas las clases creadas por este lenguaje, y por consiguiente todos sus beneficios tales como acceso a redes, bases de datos, etc., esto es un punto fundamental, ya que se respalda por un lenguaje de programación de alto nivel.

Ciclo de Vida de un Servlet

Dado que un applet también posee un ciclo de vida, los Servlets también poseen uno, el cual se define así: *init*, *service* y *destroy*.

init .- Cuando el motor de Servlets recibe una petición (*request*), comprueba si el Servlet esta cargado, el método *init* se ejecuta cuando se crea el Servlet y no vuelve a ejecutarse en cada petición del usuario. Se utiliza para establecer los valores iniciales. El Servlet se crea cuando el cliente invoca una URL que corresponda con el Servlet. Dentro del método *init*, el Servlet realiza cualquier tarea de arranque que se le haya asignado (se utiliza para inicializar valores).

service .- Este método se ejecuta cuando el servidor recibe una petición (*request*). Este método verifica que tipo de petición HTTP se ha hecho (*get*, *post*, *delete*, *put*, etc.). Existe una subclase que se encarga de atender cada petición HTTP dependiendo el método que se utilice, El método *service* transforma los objetos de petición y respuestas en sus equivalentes específicos HTTP (cada tipo de petición tiene un método asociado para atenderla, Ej. si se utiliza el método POST, se utiliza *doPost()*, para GET se utiliza *doGet()* y así sucesivamente), que examina la petición y llama al método apropiado (*doGet*, *doPost*, etc)

destroy .- El servidor elimina la instancia de un Servlet por el administrador o tal vez por permanecer inactivo durante un largo período. Este método se encarga de liberar

todos los recursos asignados durante la ejecución del método *init*,¹² (cierra conexiones con bases de datos, variables, etc.), muy similar al método *destroy* de un applet.⁴

A pesar de que el Servlet posee una gran cantidad de beneficios (por incorporar todas las características de Java), tiene un inconveniente, todo el contenido HTML es generado con sentencias de Servlet, es decir (como en JavaScript) se tiene que generar mediante la llamada de métodos (marcados en negritas, Fig. 2.3).

Este código de Servlet (Fig. 2.3) genera el texto "Hello WWW" y es visible que para generar el HTML, es necesaria la instrucción *out.println()*. Esto hace que la generación, modificación y actualización del código HTML dinámico y estático sea complicado (a diferencia de las otras tecnologías que se combinan con HTML estático a excepción de JavaScript), ya que el Servlet genera el HTML dinámico y estático. Como solución a este problema se creó otra tecnología alternativa que resuelve este inconveniente, se crearon las JavaServer Pages (JSP).

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWWW extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String docType =
            "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 "+
            ""TRANSITIONAL//EN">\n";
        out.println(docType +
            "<HTML>\n" +
            "<HEAD><TITLE>Hello WWW</TITLE></HEAD>\n" +
            "<BODY>\n" +
            "<H1>Hello WWW</H1>\n" +
            "</BODY></HTML>");
    }
}
```

Fig. 2.3

2.4 JavaServer Pages (JSP)

Las JavaServer Pages (JSP) son tecnología para generar páginas Web con contenido dinámico, donde el código de Java va insertado y mezclado junto con el HTML estático,

⁴ Hanna, Phil. (2002). JSP Manual de Referencia. España: Mc Graw Hill.

(a diferencia de los Servlets, que eran estos los que generaban el HTML tanto dinámico como estático).

Las JSP ofrecen ciertas ventajas sobre los Servlets (aunque una página JSP se convierte en un Servlet Fig 2.4 a)⁵ entre ellas están ⁶:

- ❑ El código de Java va mezclado con HTML estático (como asp y php) en Scripts llamados Scriplets.
- ❑ Se vuelven a compilar de manera automáticamente cuando es requerido.
- ❑ Es más fácil la modificación del código.
- ❑ Mayor compatibilidad con entornos de desarrollo.

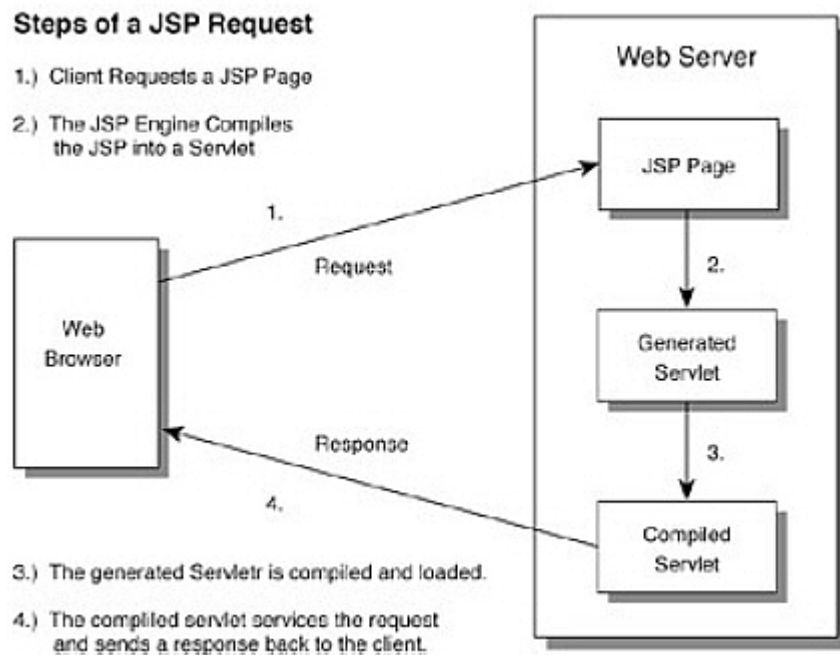


Fig 2.4a

Las páginas JSP pasan por 3 etapas ⁷:

⁵ Mastering Jakarta Struts, James Goodwill, Wiley Publishing, 2002 pag 27

⁶ Hanna, Phil. (2002). JSP Manual de Referencia. España: Mc Graw Hill.

1. **Código fuente JSP:** Es el código que se escribe dentro de un archivo con extensión **".jsp"** junto con el código HTML, que incluye instrucciones en Java, directivas JSP y diversas acciones para dar respuestas a peticiones en el Web.
2. **Código fuente Java:** El conector de JSP traduce el código fuente de un archivo JSP a código fuente Java de un Servlet.
3. **Clase Java Compilada:** Como cualquier otra clase Java compilada, se compila en código de bytes en un archivo **.class**, preparado para ser cargado y ejecutado.

El código de un documento **.jsp** (Fig 2.4b) se compila y genera un archivo **.java** y (sólo si no genera errores, Fig 2.4c) y un archivo **.class**. Es decir que en segundo plano un archivo JSP se convierte en un Servlet.

```
<% @ page contentType="text/html; charset=iso-8859-1"%>
<html>
<head>
<title>Ejemplo de un JSP</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body><%= "Hola JSP !" %></body>
</html>
```

Fig 2.4b

```
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import org.apache.jasper.runtime.*;

public class HolaJSP_jsp extends HttpJspBase {
    private static java.util.Vector _jspx_includes;
    public java.util.List getIncludes() {
        return _jspx_includes;
    }
    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {
        JspFactory _jspxFactory = null;
        javax.servlet.jsp.PageContext pageContext = null;
```

⁷ Hanna, Phil. (2002). JSP Manual de Referencia. España: Mc Graw Hill

```

HttpSession session = null;
ServletContext application = null;
ServletConfig config = null;
JspWriter out = null;
Object page = this;
JspWriter _jspx_out = null;
try {
    _jspxFactory = JspFactory.getDefaultFactory();
    response.setContentType("text/html; charset=iso-8859-1");
    pageContext = _jspxFactory.getPageContext(this, request, response, "", true, 8192, true);
    application = pageContext.getServletContext();
    config = pageContext.getServletConfig();
    session = pageContext.getSession();
    out = pageContext.getOut();
    _jspx_out = out;
    out.write("<html>\r\n");
    out.write("<head>\r\n");
    out.write("<title>Ejemplo de un JSP");
    out.write("</title>\r\n");
    out.write("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">\r\n");
    out.write("</head>\r\n");
    out.write("<body>\r\n");
    out.print("Hola JSP !" );
    out.write("</body>\r\n");
    out.write("</html>\r\n");
}
catch (Throwable t) {
    out = _jspx_out;
    if (out != null && out.getBufferSize() != 0)
        out.clearBuffer();
    if (pageContext != null) pageContext.handlePageException(t);
} finally {
    if (_jspxFactory != null) _jspxFactory.releasePageContext(pageContext);
}
}
}

```

Fig 2.4c

El archivo HolaJSP.jsp (Fig 2.4b) es traducido para generar un documento **.java** (Fig 2.4c), y si no hay generación de errores, se compila y crea un archivo **.class** (Fig 2.4d) que es el que se despliega en el navegador como resultado final:

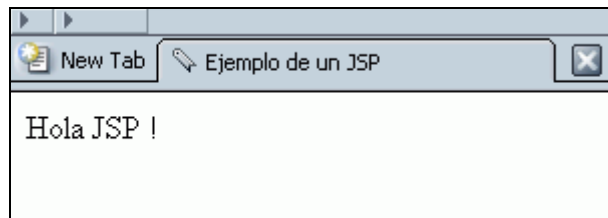


Fig 2.4d

2.4.1 Tag Library

Una de las implementaciones más importantes que ofrecen las JavaServer Pages es la creación de etiquetas personalizadas o mejor conocidas como *Tag Library*. Las Tag Libraries ofrecen algo más de lo que aportan las etiquetas comunes de HTML, permite crear etiquetas personalizadas para el desarrollo de aplicaciones más complejas. Un ejemplo de las características de las *Tag Library* es la asociación de tablas con campos específicos de una base de datos, es decir, se crean una tabla con los resultados obtenidos de una base de datos, donde cada celda es un campo específico de la consulta a la Base de Datos, donde la invocación de dicha celda, ya sabe que campo le corresponde y sabe que campo es el que se mostrara sin necesidad de realizar un barrido por separado del proceso.

Componentes que constituyen una *Tag Library*.

- ❑ Clase controladora de la etiqueta: esta clase, se encarga de realizar los procesos internos y la definición de acciones con respecto a las etiquetas y atributos que contendrá.
- ❑ Archivo descriptor de la etiqueta: un archivo descriptor de la etiqueta es un documento XML (TLD Tag Library Definition) el que se encarga de especificar los nombres y los atributos que contendrá dicha etiqueta.
- ❑ Archivo JSP que implementará la *Tag Library*. El archivo JSP que contendrá el llamado a las etiquetas personalizadas.

Se recomienda la utilización de este tipo de *framework* para proyectos que requieran manejar una gran cantidad de información, dada la complejidad que puede ocasionar la creación de las etiquetas personalizadas.

2.5 JDBC

El API (Application Programming Interface) JDBC provee un acceso a bases de datos relacionales desde Java. Utilizando el API JDBC, ejecutando aplicaciones escritas en Java ejecuta comandos SQL, también es utilizado para interactuar con diversas fuentes de datos en un entorno heterogéneo y distribuido.¹³ El API JDBC nos permite crear aplicaciones que pueden interactuar con bases de datos de las más diversas distribuciones (Microsoft ODBC, Oracle, Informix, MySQL, etc.)

2.5.1 Drivers

Un Driver (*controlador*) es una clase Java que es proporcionada por la empresa distribuidora de la Base de Datos. La principal función del Driver es el permitir la conexión de una aplicación Java a una Base de Datos obteniendo así acceso a la información almacenada.⁸

Las especificaciones JDBC clasifican a los Drivers en 4 tipos⁹, dependiendo su estructura.

Tipo 1: Este Driver es llamado "*bridge*" (puente) por que provee de una vía JDBC a un driver ODBC. Este tipo de controlados es muy limitado en términos de efectividad en Internet¹⁰. Este driver es considerado como experimental y recomendado solo para aplicaciones locales o cuando no existe otro controlador disponible.

Tipo 2: El Driver de tipo 2 convierte llamadas del API JDBC a llamadas específicas al API de la Base de Datos, para que estas llamadas sean ejecutadas, debe existir un software específico instalado en el cliente para poder funcionar apropiadamente. El controlador está construido parcialmente en lenguaje de Java y código nativo del cliente (C/C++).

⁸ Hanna, Phil. (2002). JSP Manual de Referencia. España: Mc Graw Hill.

⁹ D. Speegle, Gregory. (2002). JDBC: Practical Guide For Java Programmers. USA: Morgan Kaufmann Publishers

¹⁰ Reese, George. (2000). Database Programming with JDBC & Java. USA: O'Reilly. Segunda edición

Tipo 3: El Driver de tipo 3 esta construido totalmente en código de Java y se comunica con un servidor intermedio utilizando un protocolo de Base de Datos independiente y no se requiere de ningún software especial implementado en el cliente.¹¹

Tipo 4: El Driver de tipo 4 esta construido totalmente en código de Java e implementa un protocolo de red para una específica fuente de datos.¹² El cliente se conecta directamente a la Base de Datos. Los Drivers de este tipo llaman directamente al protocolo nativo empleando el sistema de administración de la base de datos.¹³

Como se observa, un Driver es una clase que al ser implementada en una aplicación construida en Java, permite un acceso de comunicación a cualquier tipo de Base de Datos, ya sea de uso comercial (Oracle, Informix, Sybase, Microsoft SQL Server, etc.) o de uso publico (MySQL, Postgres, mSQL, etc.), lo cual ayuda a la creación de proyectos a gran escala (empresas de desarrollo, de investigación, etc.) como proyectos de recursos limitados (Instituciones Educativas, Instituciones Gubernamentales, etc.).

2.6 Administrador de servicios Web

Los servicios Web, en términos generales, son servicios ofrecidos vía Web. En un escenario típico Web, una aplicación de negocios envía una petición a un servicio para obtener una URL utilizando un protocolo vía http, el servicio recibe la petición, la procesa y devuelve una respuesta.¹⁴

Un administrador de servicios Web es aquel que provee de un servicio específico Web, pero no un servicio como los mencionados en el e-commerce, sino un servicio *para* Web. Existe una gran cantidad de servicios Web, con distintos propósitos, entre los

¹¹ Database Programming with JDBC & Java, George Reese. O'Reilly, 2nd Edition

¹² Jon Ellis, Linda Ho, Maydene Fisher. JDBC 3.0 Specification. [en línea], Sun Microsystems Final Release 1 de diciembre 2001, disponible en WWW <http://Java.sun.com/products/jdbc/download.html>

¹³ JDBC: Practical Guide For Java Programmers, Gregory D. Speegle. Morgan Kaufmann Publishers, 1st Edition

¹⁴ Eric Armstrong, Stephanie Bodoff, Debbie Carson, Maydene Fisher, Scott Fordin, Dale Green, Kim Haase, Eric Jendrock. The Java Web Services Tutorial. [en línea] Sun Microsystems, Febrero 2003. <http://Java.sun.com/webservices/tutorial.html>

cuales están, servicios de correo electrónico, servicios de bases de datos, servicios de ejecución de aplicaciones, servicios para envío de archivos, etc.

Por ejemplo, cuando se visita un catálogo virtual de una tienda de discos, detrás de la página existen diversos servicios Web activados. Si la página esta hecha con algún tipo de tecnología para páginas dinámicas (sea asp, php, jsp entre otras), indica que existe un servicio Web activado para visualizar este tipo de páginas; si realizamos una consulta para visualizar algún tipo de música en especial, se mostrara una lista de títulos disponibles gracias a un servicio Web de bases de datos; en caso de no encontrar el titulo que se desea, se puede enviar un correo electrónico para preguntar por él, el correo electrónico llegará gracias a un servicio Web de correo electrónico.

Cabe señalar la necesidad de un administrador de servicios Web para poder visualizar páginas con contenido dinámico. Existe una variedad de servicios Web para poder ejecutar páginas Web dinámicas, Microsoft tiene entre sus herramienta el Personal Web Server (PWS) y el Internet Information Server (IIS), Con estas herramientas es posible ejecutar y mostrar páginas Web dinámicas con tecnología **asp**, además ser configuradas (con el software apropiado) para ejecutar páginas Web dinámica con tecnología **php**. Microsoft Personal Web Server es un servidor Web personal que se utiliza para alojar un sitio Web en su equipo personal de casa o como parte de la intranet corporativa o para crear y probar un sitio Web antes de alojarlo en un Proveedor de servicios Internet (ISP).¹⁵

Existen varias herramientas que contienen las implementaciones para Servlets y JavaServer Pages. Entre las más conocidas esta Macromedia Jrun ¹⁶, WebSphere de IBM ¹⁷, Jakarta Tomcat ¹⁸ y el Java Web Service Development Pack (JWSDP) ¹⁹ aunque viene con la distribución de Tomcat. La gran variedad herramientas orientadas a la ejecución de aplicaciones para la construcción de páginas Web dinámicas es muy

¹⁵ Microsoft Corporation, Internet Information Services, [en línea] <http://www.microsoft.com/WindowsServer2003/iis/default.msp>

¹⁶ Adobe Systems Incorporated..Macromedia JRun 4 [en línea] <http://www.macromedia.com/software/jrun/>

¹⁷ IBM Corporate. WebSphere Software [en línea] <http://www-306.ibm.com/software/websphere/>

¹⁸The Apache Software Foundation. Apache Tomcat. [en línea] <http://tomcat.apache.org/>

¹⁹ Copyright 1994-2006 Sun Microsystems, Inc. Java Web Services Developer Pack . [en línea] <http://Java.sun.com/webservices/downloads/webservicespack.html>

variada y de distribución pública (existen distribuciones que requieren de la compra de una licencia del producto así como de distribución gratuita).

3 Aplicación

3.1 Creación de un sistema para la administración de recursos de una biblioteca usando tecnología JSP

Se hará uso de la tecnología Java (JSP, JDBC, etc.), para evaluar si resuelve los requerimientos necesarios para la creación de un sistema capaz de administrar recursos de una biblioteca, a continuación se detallarán los requerimientos necesarios para su evaluación.

Una Universidad tiene la necesidad de ofrecer un catálogo en línea de los libros que tiene en biblioteca para su consulta, el sistema deberá tener la capacidad de poder ser accedido por Internet o por una intranet. El préstamo de dichos libros se llevará a cabo dentro de las instalaciones de la escuela. Se debe contar además con un sistema administrador libros de la biblioteca siendo este configurable dependiendo las diferentes necesidades de los campus de la Institución, sólo los alumnos de la escuela y personal docente podrán acceder a la aplicación (dado que sólo a ellos se les permite el servicio de préstamo y consulta de libros). Se deben diferenciar los perfiles de acceso al sistema, ya que las características de los servicios pueden variar de un usuario a otro, solamente un usuario especial (Administrador) podrá ser capaz de configurar el sistema según sean las necesidades de la institución y además tendrá acceso a todos los módulos del sistema. El sistema tendrá la característica principal, de ser compatible con diferentes plataformas (Windows y/o Linux) dado que no se cuenta con un estándar de la misma dentro de la Institución, además de tener la capacidad de poderse desplegar en cualquier tipo de navegador, ya que los usuarios al consultarlo de manera remota, no les será posible mantener un control del tipo de navegador que se tiene, siendo que cada uno utiliza un navegador distinto al de la plataforma local (MIE/Netscape).

Se detallará el proceso teórico acompañado con pantallas sugeridas para el funcionamiento del sistema, además se explicará el funcionamiento del sistema.

Uno de los requerimientos indispensables, es que la aplicación deberá tener la capacidad de ser vista en Intranet (la parte referente a los procesos) y también por

medio de Internet (la parte referente a las consultas), Java posee tecnologías que permiten la ejecución de aplicaciones mediante el Web (Servlets, JSP, etc.), es por ello, que se tomará esta tecnología como opción para la creación de este sistema.

Para el desarrollo del Sistema se emplearán las siguientes herramientas:

- ❑ Sistema Operativo: Windows XP Home Edition service pack 2, Linux Mandrake 10.
- ❑ Lenguajes de Programación: Java, JDK versión 1.4.2_05.
- ❑ IDE: Sun ONE Studio 5 Standar Edition.
- ❑ Application Server: Sun ONE Application Server 7.
- ❑ Base de Datos: MySQL 4.0.20.
- ❑ Navegador Web: Netscape 7.2.
- ❑ Otros: MySQL-Front version 3.0, JavaScript, HTML 4.01.

El Sistema estará dividido en 2 módulos, que corresponden a Modulo de Usuarios, que es el que se encargará del registro de usuarios así como la configuración de los perfiles de cada uno de ellos. El modulo de administración de la biblioteca se encargará de registrar los libros así como la configuración de submodulos para el correcto funcionamiento del sistema. Para fines prácticos, solo se mostrara el modulo de Administración de la biblioteca, ya que es el más completo y cumplirá con la función de visualizar los aspectos mas importantes que se desean demostrar.

Estructuración de los procesos

Una buena distribución de procesos ayuda a repartir las actividades y mejorar la administración de las tareas.

Se dividieron los procesos en base a la tecnología que ofrece Java: JSP, Servlets y Clases u Objetos Java.

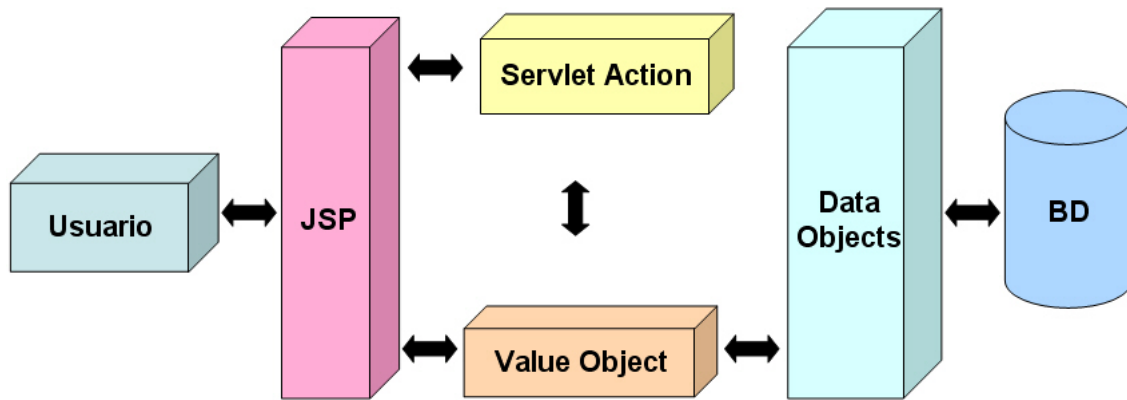


Fig. 3.1a

Usuario: el Usuario es la persona que interactúa con el sistema, también nombrado como "*cliente*".

JSP: Corresponde al *Front*, es decir la interfase que es visualizada por el cliente. La pantalla con la cual se muestran los datos procesados o el resultado de la interacción del cliente.

Servlet Action: Servlet encargado de recibir todas las acciones generadas en el JSP, el cual dependiendo la acción solicitada, se encarga de canalizar el resultados.

Value Object: Clases u objetos de Java que contienen el datos procesado o por procesar de la base de datos.

Data Objects: Clases u objetos de Java que se encargan de realizar las conexiones y consultas a la base de datos, llenando los Value Objects como resultado del proceso.

La estructura se define de la siguiente manera:

El usuario al acceder al sistema se despliega una página JSP, que al interactuar con la misma, el JSP se comunica con el Servlet Action, donde este a su vez se comunica con los Value Objects que son llenados con la información solicitada, los Data Objects se encargan de hacer las conexiones con la base de datos y llenar los Value Objects que son enviados ya sea al ServletAction o al JSP. Este diseño de estructura evita que los usuarios tenga acceso directo a los datos del sistema.

Con el diseño de estructura anterior se generó una administración de clases como se muestra en la Fig. 3.1b.

Portabilidad

El sistema inicialmente se instalará en sistema operativo Windows, pero si el sistema llegara a necesitar de un sistema operativo más robusto, éste podrá ser exportado sin ningún tipo de contratiempo a otra plataforma como Unix o Linux, ya que el sistema contará con la capacidad de ser compatible con cualquier sistema operativo existente.

Base de datos

Se utilizará como base de datos a MySQL por ser de carácter gratuito. En caso de que exista la posibilidad de utilizar una base de datos más robusta, será posible realizar la migración de la misma con una serie de modificaciones mínimas, donde la posibilidad de volver a crear o compilar las clases de Java sea muy baja y si existe el caso sólo se modificarán algunos Data Objects y unos parámetros para el acceso a la base de datos.

Se propone el siguiente diagrama de Entidad Relación DER, (Fig 3.1c).

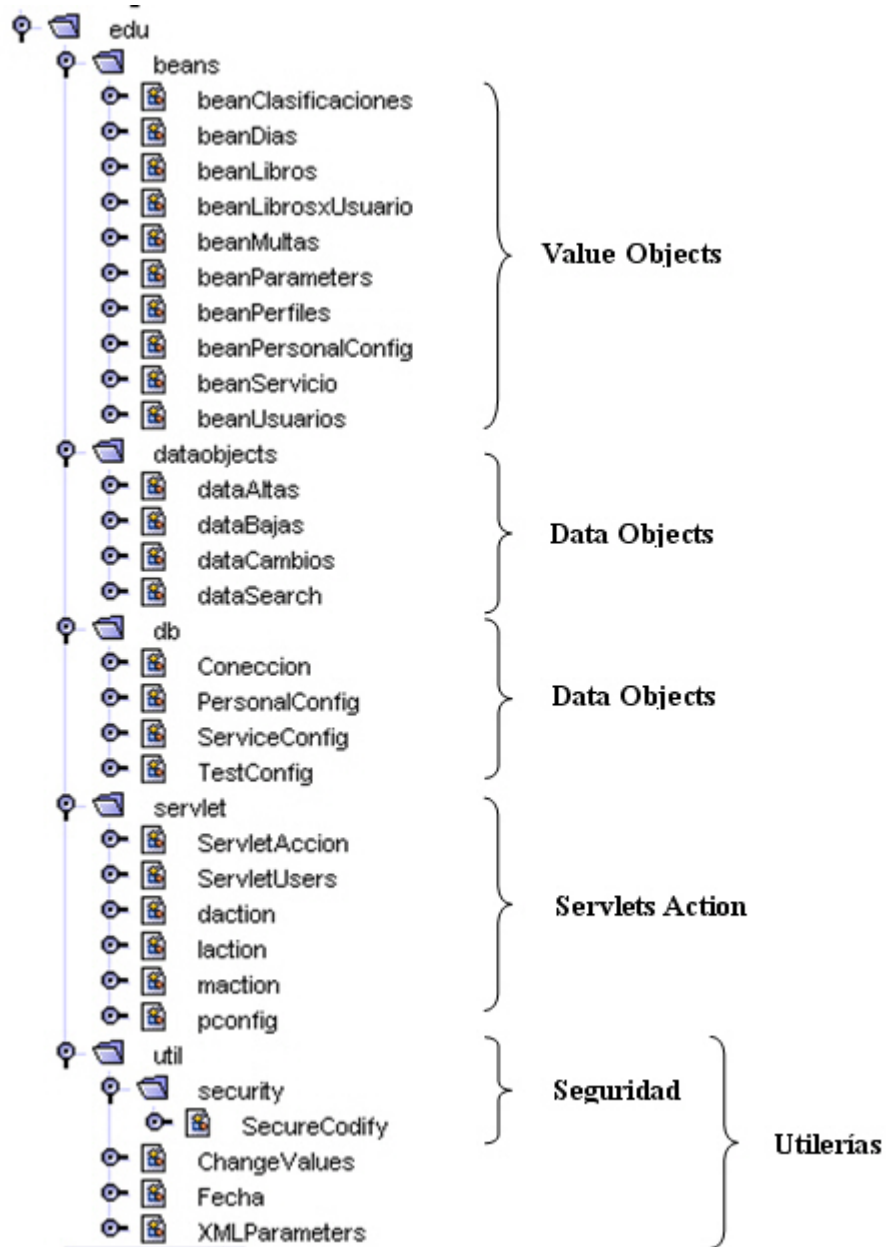


Fig. 3.1b

Seguridad

En aspectos de seguridad, sólo se utilizó un algoritmo numérico para la codificación del password de cada usuario, es decir que si se accesa al password de cada usuario de manera directa, será visible una serie de números, que para decodificarlos, será necesaria la clase de seguridad que se implementó.

Costos

Dado que el sistema esta orientado a una institución educativa, los gastos son un aspecto importante, por lo que es necesario reducirlos al máximo, así que se ve la posibilidad de utilizar software de uso gratuito para evaluar si se cubre las necesidades de desempeño y económicas.

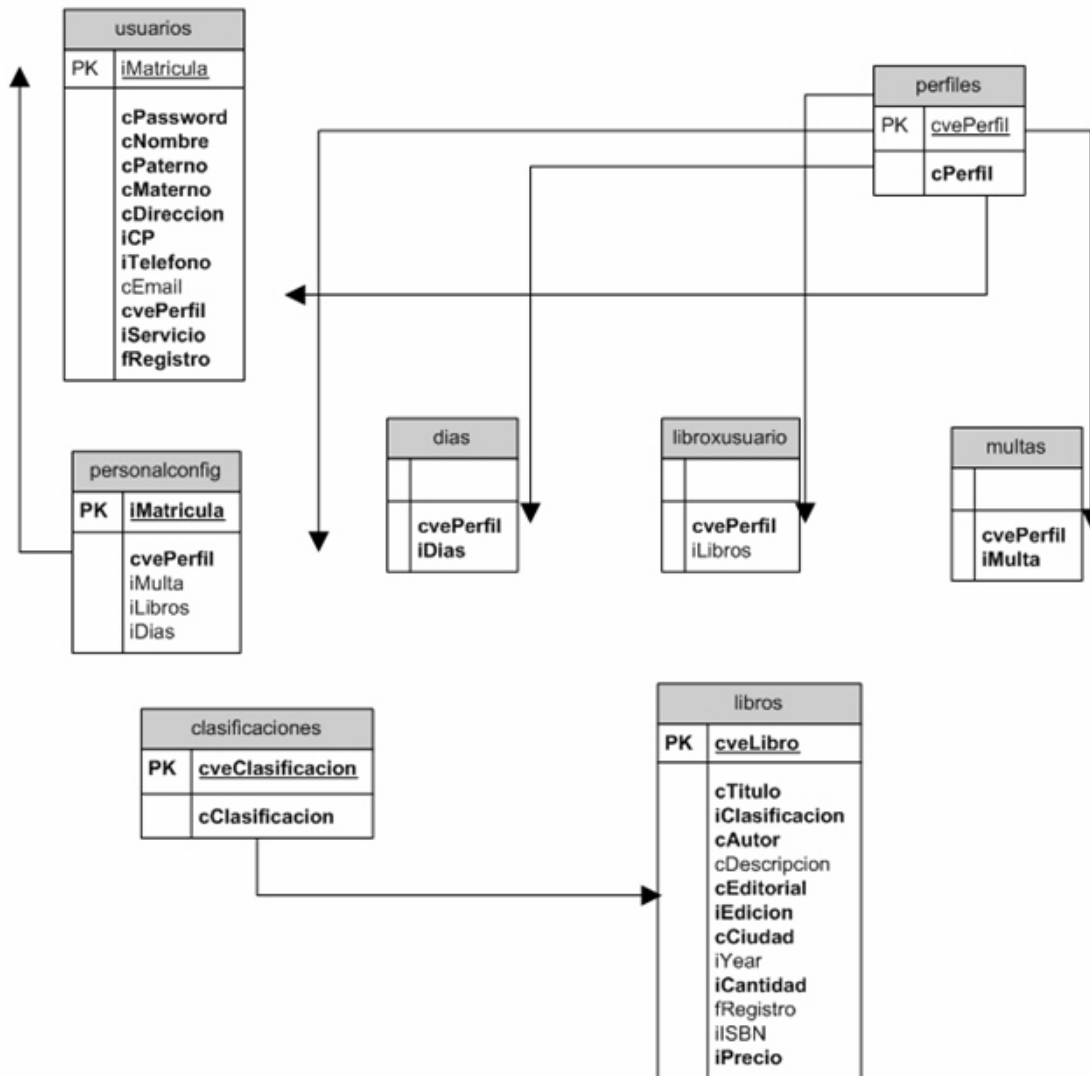


Fig. 3.1c

Rendimiento.

Se sabe que Java es un lenguaje interpretado, y por consiguiente el tiempo de ejecución es mas lento que el de un programa normal, pero al ver los resultados del rendimiento del sistema ya en ejecución, se verá su verdadera capacidad.

3.2 Lista de datos

La Lista de Datos consiste en la enumeración y descripción de los datos que se introducirán al sistema (módulo Altas y Cambios) para su registro, es necesario tener clara esta información ya que el llenado con datos correctos, ayudará al sistema a tener un mejor desarrollo y desempeño en cuanto a su funcionamiento. Los datos se enumerarán a continuación y se dará la descripción de cada uno de ellos. Para conocer la longitud de cada uno de los campos, será necesario consultar el *Diccionario De Datos*. Para la captura de los datos que alimentarán el sistema no se deberán introducir caracteres especiales, tales como comillas ("), apostrofes (') o acentos(`), el sistema omitirá este tipo de caracteres de manera automática, pero para obtener un mejor rendimiento, es necesario evitar la introducción de este tipo de símbolos así como de otros caracteres especiales donde no sean requeridos. Para evitar mensajes de error o advertencias se recomienda ingresar solo el tipo de dato que esta permitido.

Llenado de la información

Se llenarán todos los campos correspondientes con carácter de obligatorios los que estén indicados con un asterisco color rojo (*), los demás campos son opcionales, es decir que no es necesario su llenado para su registro. Si se introdujeron datos inválidos dentro de algún campo, el sistema enviará un mensaje listando detalladamente el(los) campo(s) que se escribieron de manera errónea así como el tipo de dato que le corresponde.

Clave

El campo Clave es de carácter obligatorio, es el identificador principal con el cual se registrará cada uno de los libros en el sistema, la Clave es de tipo numérico con longitud de 10 dígitos. La clave ingresada debe ser única, no deberá existir ningún duplicado de la misma (el sistema no lo permite), en caso de que el Usuario¹ intente registrar un libro con una clave ya existente, el sistema enviará un mensaje de error y no continuará con el proceso (solamente en módulo de Altas).

Autor

El campo Autor es de carácter obligatorio, este campo define el(los) autor(es) de cada libro, el tipo de dato es solo caracteres con longitud de 150 dígitos. En caso de que el

¹ Usuario es toda persona que haga uso del sistema.

ejemplar cuenta con más de un autor, se podrán introducir todos los autores, separándolos con comas (,).

Título

El campo Título es de carácter obligatorio, este campo define el título del libro que se va a registrar, es alfanumérico con longitud de 80 dígitos, es decir que se pueden combinar números con letras.

Clasificación

El campo Clasificación se obtiene de un catalogo, ayuda a tener una mejor clasificación de los libros registrados, este campo se selecciona sobre información ya cargada por el Administrador en el sistema (ver módulo de Clasificaciones) no existiendo en el ningún tipo de validación especial.

Descripción

El campo Descripción es de carácter opcional, en este campo se da una breve descripción o sinopsis del libro con la finalidad de informar un brevemente acerca del contenido del libro, es alfanumérico es decir que se pueden combinar números con letras.

Editorial

El campo Editorial es de carácter obligatorio, da el nombre de la editorial encargada de imprimir el libro, es alfanumérico es decir que se pueden combinar números con letras con longitud de 30 dígitos.

Edición

El campo Edición es de carácter obligatorio, describe el número de edición del libro, es de tipo numérico con longitud de 2 dígitos.

Ciudad

El campo Ciudad es de carácter opcional, describe la ciudad de origen donde se imprimió el libro, el tipo de dato es solo caracteres con longitud de 20 dígitos.

Año

El campo Año es de carácter opcional, describe al año en que se imprimió el libro, es de tipo numérico con longitud de 4 dígitos.

Cantidad

El campo Cantidad es de carácter obligatorio, describe la cantidad de libros que se tienen en existencia física en la biblioteca, es de tipo numérico con longitud de 2 dígitos.

Fecha de Registro

El campo Fecha de Registro, es un campo que llena el sistema de manera automática. Este campo es la fecha en que se registró el libro en el sistema.

ISBN

El campo ISBN es de carácter opcional, describe código internacional del libro (no todos los libros poseen este código), es de tipo numérico con longitud de 11 dígitos.

Precio

El campo Precio es de carácter obligatorio, describe el costo del libro de manera individual, es de tipo numérico con longitud de 5 dígitos.

3.3 Módulo altas

Dentro del sistema, si el usuario tiene el perfil de Administrador, podrá visualizar esta opción en la pantalla, se presiona el botón **Altas**, se enviará a una página de registro (Fig. 1.1).

Al momento de terminar de llenar los campos necesarios (Fig. 1.2), para continuar con el registro, se deberá presionar el botón de **Registrar** o si se desea volver a llenar los campos, se presiona el botón **Limpiar**, para borrar todos los campos que hallan sido completados, regresando al estado inicial. Después de presionar el botón de **Registrar**.

The screenshot shows a web browser window with the address bar displaying "http://localhost/jspfiles/modulos/10100.jsp". The page title is "Bienvenidos !!!". At the top, there are five navigation buttons: "ALTAS", "BAJAS", "CAMBIOS", "CONSULTAS", and "CONFIGURAR". The main heading is "Biblioteca Escolar". Below the heading is a registration form with the following fields and labels:

- Clave:** * [text input]
- Autor:** * [text input]
- Título:** * [text input]
- Clasificación:** * [dropdown menu with "Informatica" selected]
- Descripción:** [text area]
- Editorial:** * [text input]
- Ciudad:** [text input]
- Cantidad:** * [text input]
- ISBN:** [text input]
- Edición:** * [text input]
- Año:** [text input]
- Fecha de Registro:** 13/2/2005
- Precio:** * [text input]

At the bottom of the form are two buttons: "Registrar" and "Limpiar". Below the buttons is a note: "NOTA. Los datos con asterisco (*) no pueden quedar en blanco." The browser's status bar at the bottom shows the URL "http://localhost/jspfiles/modulos/10100.jsp".

Fig. 1.1

Después de presionar el botón de **Registrar**, aparecerá una pantalla con la vista preeliminar de todos los datos que se introdujeron (Fig. 1.3), para confirmar o preguntar si los datos llenados son los correctos, la pantalla contendrá 2 botones, **Continuar** y **Volver**. Si en la vista preeliminar se observa algún error en el llenado de los campos, (ortográfico, dato equivocado, etc.) se presiona el botón **Volver**, para regresar a la pantalla anterior, donde los datos aparecerán en la forma en que fueron llenados, permitiendo así la modificación de cualquier campo sin tener que llenar todos los datos nuevamente.

Bienvenidos !!!

ALTAS **BAJAS** **CAMBIOS** **CONSULTAS** **CONFIGURAR**

Biblioteca Escolar

Clave: * 98746512
Autor: * Kathy Sierra, Bert Bates
Título: * Java 2: Sun Certified Programmer & Developer For...
Clasificación: * Informatica
Descripción: Libro de estudio para exámenes de certificación Java, nivel Programmer y Developer
Editorial: * Mc Graw Hill
Ciudad: E.U.
Cantidad: * 5
ISBN: 129873994
Edición: * 1
Año: 2003
Fecha de Registro: 13/2/2005
Precio: * 580

NOTA. Los datos con asterisco (*) no pueden quedar en blanco.

http://localhost/jspfiles/modulos/10100.jsp

Fig. 1.2

Bienvenidos !!!

ALTAS **BAJAS** **CAMBIOS** **CONSULTAS** **CONFIGURAR**

Biblioteca Escolar

Si los datos son correctos presione Continuar

Clave: * 98746512
Autor: * Kathy Sierra, Bert Bates
Título: * Java 2: Sun Certified Programmer & Developer For Java 2
Clasificación: * Informatica
Descripción: Libro de estudio para exámenes de certificación Java, nivel Programmer y Developer
Editorial: * Mc Graw Hill
Ciudad: E.U.
Cantidad: * 5
ISBN: 129873994
Edición: * 1
Año: 2003
Fecha de Registro: 13/2/2005
Precio: * \$ 580.00 MN

Document: Done (1.735 secs)

Fig. 1.3

Se presiona el botón de **Continuar**, cuando los datos en la vista preliminar son los que se desean realmente Ingresar. Posteriormente aparecerá una pantalla con un mensaje de estatus donde indicará si el proceso de registro fue exitoso o si ocurrió algún error. La pantalla con el mensaje de estatus (éxito o error al momento de

intentar agregar cualquier registro) contiene un botón de retorno, el cual nos enviará a la pantalla de búsqueda de este módulo en especial.

3.4 Módulo bajas

Dentro del sistema, si el usuario tiene el perfil de Administrador, podrá visualizar esta opción en la pantalla, si presiona el botón **Bajas**, se enviará a una página de búsqueda (Fig. 2.1), donde por medio de 3 diferentes tipos de búsqueda (Clave de Libro, Título, Autor), se podrá localizar el ejemplar que se desea eliminar.

La pantalla de **Búsqueda** posee ciertas restricciones en su búsqueda. Se debe seleccionar un modo de búsqueda y escribir la palabra mediante la cual se buscará el libro, (ej. Título). En caso de que existan coincidencias con la palabra utilizada aparecerá uno o más registros en existencia (Fig. 2.2) con una breve descripción del título dependiendo del tipo de búsqueda seleccionada, en caso contrario aparecerá un mensaje (Fig. 2.3).




Fig. 2.1



Fig. 2.2



Fig. 2.3

Después de ser desplegados los resultados de búsqueda, se presiona el botón que se encuentra junto al título , donde se enviará a una página donde aparecerá la información completa del título seleccionado (Fig. 2.4).

La página donde se muestra la información del título seleccionado, mostrará 2 botones con la leyenda **Eliminar** y **Regresar** respectivamente. El botón de **Regresar**, nos posiciona en la página anterior con los resultados de búsqueda que se habían obtenido; el botón de **Eliminar** es para borrar el registro completo dentro del sistema. Al presionar el botón **Eliminar**, aparecerá un mensaje que tiene como finalidad preguntar a usuario si realmente desea borrar el registro, si el usuario presiona cancelar, el proceso se interrumpirá, si presiona **Aceptar**, se eliminará el registro del sistema y se desplegará una página que informará el estatus del proceso, esto es, si se realizó el borrado de manera exitosa o si existió un error en el proceso de borrado.




Fig. 2.4

La pantalla con el mensaje de estatus (éxito o error al momento de intentar eliminar cualquier registro) contiene un botón de retorno, el cual nos enviará a la pantalla de búsqueda de este módulo en especial.

3.5 Módulo cambios

Si el usuario tiene el perfil de Administrador dentro del sistema podrá visualizar esta opción en la pantalla, se presiona el botón **Cambios** y se enviará a una página de búsqueda (similar a la Fig. 2.1 de Bajas), donde, por medio de 3 diferentes tipos de búsqueda (Clave de Libro, Título, Autor), se podrá localizar el ejemplar que se desea cambiar o modificar.

La pantalla de **Búsqueda** posee ciertas restricciones (ver detalles en la descripción de esta sección). Se debe seleccionar un modo de búsqueda y escribir la palabra mediante la cual se buscará el libro, (ej. Título). Dependiendo el tipo de búsqueda seleccionada, aparecerá, si existen coincidencias con la palabra utilizada, uno o más registros en existencia (similar a la Fig. 2.2 de Bajas) con una breve descripción del título, en caso contrario aparecerá un mensaje (similar a la Fig. 2.3 de Bajas).

Después de ser desplegados los resultados de búsqueda, se presiona el botón que se encuentra junto al título  , donde se enviará a una página donde aparecerá la información completa del título seleccionado (Fig. 3.4) con todos los campos listos para agregar o modificar cualquier dato que se desee, a excepción de la clave del libro y fecha del registro, que son los únicos datos que no se permiten modificar, (en el caso de que se desee cambiar la clave de libro, se tendrá que dar de baja el registro y agregar uno nuevo con los datos correspondientes). Existen 3 botones en dicha página que son **Actualizar**, **Regresar** y **Limpiar**. El botón de **Actualizar**, permite que los cambios realizados, se registren en el sistema, el Botón de **Regresar**, permite volver a la página de búsqueda, con los resultados de búsqueda anteriores, el botón de **Limpiar**, vuelve al estado anterior los datos que se mostraban en pantalla originalmente.

Después de presionar el botón **Actualizar**, continuará en una pantalla con un mensaje de estatus donde aparecerá un mensaje, dependiendo si se realizó la actualización de manera exitosa o si existió un error en el proceso de borrado. La pantalla con el mensaje de estatus (éxito o error al momento de intentar actualizar cualquier registro) contiene un botón de retorno, el cual nos enviará a la pantalla de búsqueda de este módulo en especial.



Biblioteca Escolar

Clave: 98746512

Autor: * Kathy Sierra, Bert Bates

Título: * Java 2: Sun Certified Programmer & Developer For ...

Clasificación: * Informatica

Descripción: Libro de estudio para exámenes de certificación Java, nivel Programmer y Developer

Editorial: * Mc Graw Hill

Edición: * 1

Ciudad: E.U.

Año: 2003

Cantidad: * 5

Fecha de Registro: 13/2/2005

ISBN: 129873994

Precio: * 580

Actualizar Regresar Limpiar


NOTA. Los datos con asterisco (*) no pueden quedar en blanco.

Fig. 3.4

3.6 Módulo consultas

Dentro del sistema, cualquier perfil de usuario (Administrador, Alumno, Profesor), podrá visualizar esta opción en la pantalla, siempre y cuando el usuario no este en el estatus de Suspensión de Servicio (solo perfiles de Alumno, Profesor), se presiona el botón **Consultas**, se enviará a una página de búsqueda (similar a la Fig. 2.1 de Bajas), donde, por medio de 3 diferentes tipos de búsqueda (Clave de Libro, Título, Autor), se podrá localizar el ejemplar que se desea consultar.

La pantalla de **Búsqueda** posee ciertas restricciones en su funcionamiento (ver detalles en la descripción de esta sección). Se debe seleccionar un modo de búsqueda y escribir la palabra mediante la cual se buscará el libro, (ej. Título). Dependiendo el tipo de búsqueda seleccionada, aparecerá, si existen coincidencias con la palabra utilizada, uno o más registros en existencia (similar a la Fig. 2.2 de Bajas) con una breve descripción del título, en caso contrario aparecerá un mensaje (similar a la Fig. 2.3 de Bajas).

Después de ser desplegados los resultados de búsqueda, se presiona el botón que se encuentra junto al título  , donde se enviará a una página donde aparecerá la información completa del título seleccionado (similar a la Fig. 2.4 de Bajas). Existen un botón en dicha página que es **Regresar**. El Botón de **Regresar**, permite volver a la página de búsqueda, con los resultados de búsqueda anteriores.

3.7 Módulo configurar

Si el usuario tiene el perfil de Administrador dentro del sistema podrá visualizar esta opción en la pantalla, al presionar el botón **Configurar**, se enviará a una página de opciones (Fig. 5.1), donde se podrán configurar 5 diferentes módulos: Multas, Días, Límite de Libros, Suspensión del Servicio y Agregar Clasificaciones.

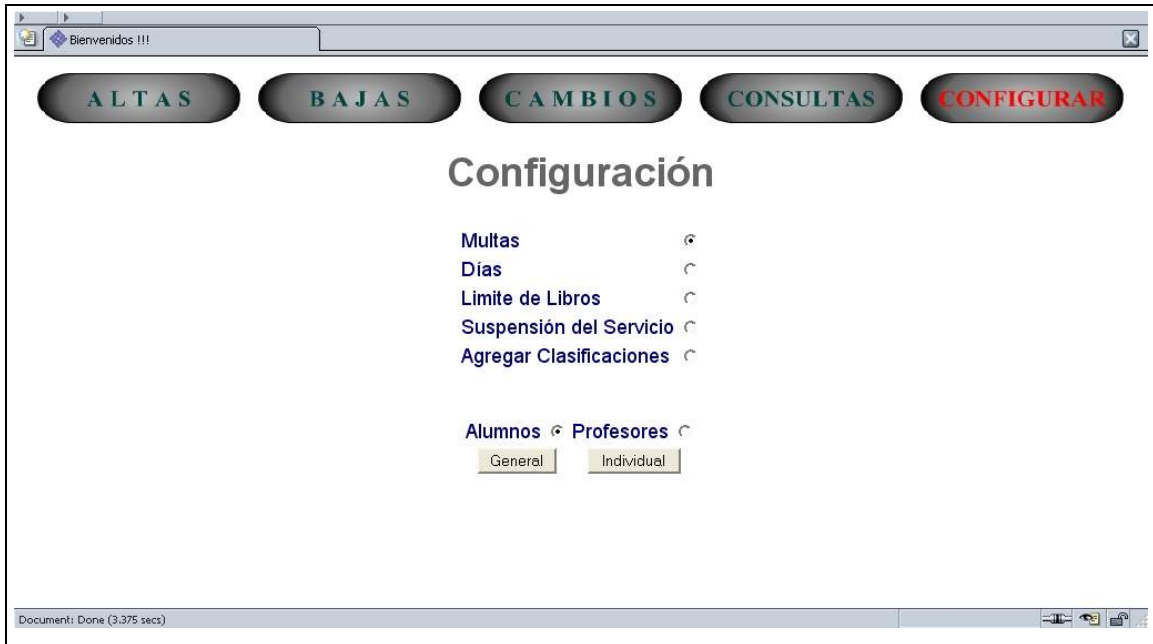


Fig. 5.1

La configuración se aplica en dos modalidades; General e Individual (por perfil de Alumno y Profesor).

General: quiere decir que la configuración se aplicará a todos los usuarios del perfil seleccionado (sea Alumno o Profesor).

Individual: quiere decir que la configuración se aplicará de manera individual por el perfil seleccionado (sea Alumno o Profesor). Esta modalidad se aplica solo en casos especiales.

3.7.1 Descripción de los módulos de configuración

Multas: Es el costo que se cobrará al usuario por cada día de retraso a la fecha de devolución de cualquier libro prestado (el costo es por libro y en pesos).

Días: El número de días que se prestará un libro, rebasando esta cantidad de días, se penalizará al usuario con una multa por cada día de retraso, dependiendo la cantidad asignada en el módulo de Multas.

Libros: Es la cantidad de libros que se le prestará al usuario.

Suspensión de Servicio: Este concepto se utilizará en el caso de que al usuario, por alguna razón se le suspenda el servicio de préstamos.

Agregar Clasificaciones: El Administrador se encargará de crear, modificar o eliminar el catálogo de clasificaciones, por medio de los cuales, cada libro será asignado.

Multas (General)

Dentro del módulo **Configurar** se selecciona Multas, se elige el tipo de usuario (Alumno o Profesor) al cual se le asignará la cantidad y se presiona el botón de **General**. Aparecerá un pantalla (Fig. 5.2) donde se mostrará el título del módulo (Multas), el tipo de configuración (General) y el perfil que se va a configurar (Alumno o Profesor). Existen dos botones (Aceptar y Regresar) y dos campos de texto, el primero que tiene la leyenda Valor Actual, es la multa que se está aplicando actualmente (si es la primera vez que se ejecuta esta configuración es posible que aparezca el valor de cero) y el campo Valor Nuevo, que es donde se le asignará la nueva cantidad. Para cambiar la cantidad existente, se inserta un nuevo valor dentro del campo de Valor Nuevo y se presiona Aceptar (el sistema no actualizara ningún carácter que no sea numérico, enviando un mensaje de error si esto llegara a suceder) o Regresar si no se desea configurar el módulo. Si se presionó el botón de Aceptar, aparecerá una pantalla con un mensaje de error o éxito junto con un botón de volver, que nos enviará a la página inicial del módulo de **Configurar**.

Multas (Individual)

Dentro del módulo **Configurar** se selecciona Multas, se elige el tipo de usuario (Alumno o Profesor) al cual se le asignará la cantidad a pagar por la multa y se presiona el botón de **Individual**. Aparecerá un pantalla de búsqueda (Fig. 5.3) donde, por medio de 3 diferentes tipos de búsqueda (Clave de Usuario, Nombre, Apellido Paterno), se podrá localizar el usuario que se desea configurar, se mostrará además el título del módulo (Multas), el tipo de configuración (Individual) y el perfil que se va a configurar (Alumno o Profesor).

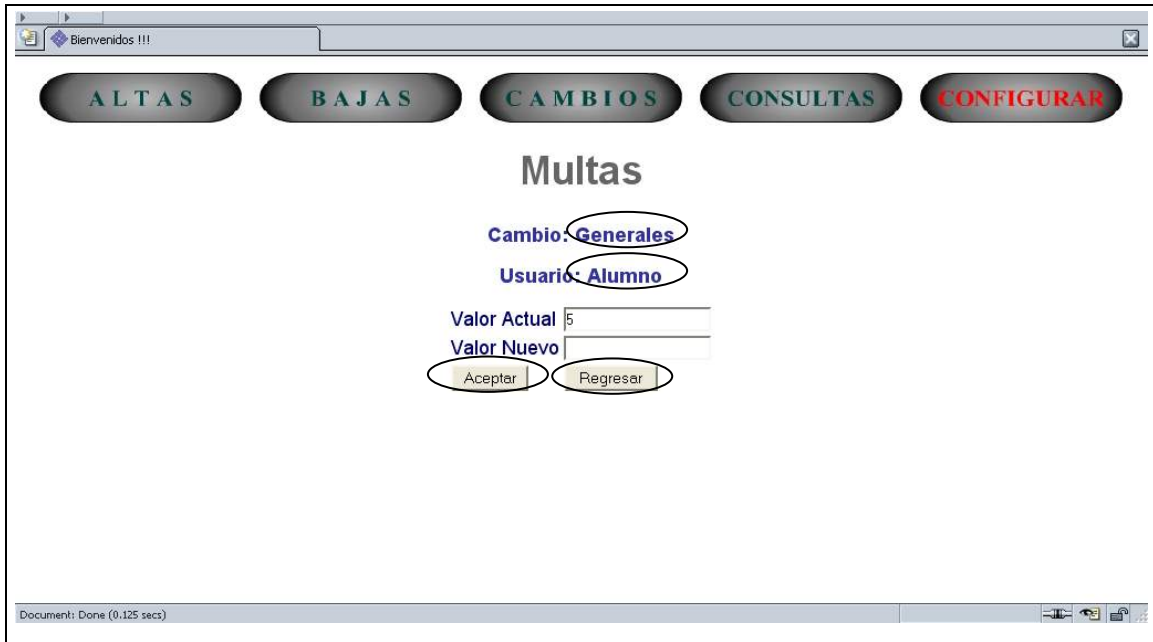


Fig. 5.2



Fig. 5.3

La pantalla de Búsqueda (Módulo Configuración) posee ciertas restricciones en su búsqueda (ver detalles en la descripción de esta sección). Se debe seleccionar un modo de búsqueda y escribir la palabra mediante la cual se buscará al Usuario, (ej. Nombre). Dependiendo el tipo de búsqueda seleccionada, aparecerán uno o más registros en existencia con una breve descripción, si existen coincidencias con la palabra utilizada, (Fig. 5.4), en caso contrario aparecerá un mensaje de aviso (Fig.


5.5). Después de ser desplegados los resultados de búsqueda, se presiona el botón que se encuentra junto al título , donde se enviará a una página donde aparecerá la configuración actual de la opción seleccionada (Fig. 5.6), sean Multas, Días o Limite de Libros o un Cero en caso de no contar con dicha configuración, Existen una serie de botones con diferentes propósitos que son **Regresar**, **Guardar**, **Borrar** y **Optimizar**, El Botón de **Regresar**, permite volver a la página de búsqueda, con los resultados de búsqueda anteriores. El botón de **Guardar** se encarga de guardar el cambio en la configuración del usuario seleccionado. El botón de **Borrar** se encarga de eliminar la configuración especial del usuario seleccionado, es decir que volverá a contar con la configuración general de todos lo demás usuarios de su perfil. El botón e **Optimizar**, se encarga de eliminar registros que ya no se usarán más, optimizando así el rendimiento del sistema, se recomienda que después de realizar la acción de **Guardar** o **Borrar**, se utilice dicha acción.



Fig. 5.4



Fig. 5.5



Fig. 5.6


Días (General)

Dentro del módulo **Configurar** se selecciona Días, se elige el tipo de usuario (Alumno o Profesor) al cual se le asignará el número de días y se presiona el botón de **General**. Aparecerá un pantalla (similar a la Fig. 5.2 de Multas) donde se mostrará el título del módulo (Límite de Días), el tipo de configuración (General) y el perfil que se va a configurar (Alumno o Profesor). Existen dos botones (Aceptar y Regresar) y dos campos de texto, el primero que tiene la leyenda Valor Actual, es la multa que se esta aplicando actualmente (si es la primera vez que se ejecuta esta configuración es posible que aparezca el valor de cero) y el campo Valor Nuevo, que es donde se le asignará el nuevo valor. Para cambiar la cantidad existente, se inserta un nuevo valor dentro del campo de Valor Nuevo y se presiona Aceptar (el sistema no actualizará

ningún carácter que no sea numérico, enviando un mensaje de error si esto llegara a suceder) o Regresar si no se desea configurar el módulo. Si se presionó el botón de Aceptar, aparecerá una pantalla con un mensaje de error o éxito junto con un botón de volver, que nos enviará a la página inicial del módulo de **Configurar**.

Días (Individual)

Dentro del módulo **Configurar** se selecciona Días, se elige el tipo de usuario (Alumno o Profesor) al cual se le asignará la cantidad y se presiona el botón de **Individual**. Aparecerá un pantalla de búsqueda (similar a la Fig. 5.3 de Multas) donde, por medio de 3 diferentes tipos de búsqueda (Clave de Usuario, Nombre, Apellido Paterno), se podrá localizar el usuario que se desea configurar, se mostrará además el título del módulo (Limite de Días), el tipo de configuración (Individual) y el perfil que se va a configurar (Alumno o Profesor).

La pantalla de **Búsqueda** (Módulo Configuración) posee ciertas restricciones en su búsqueda (ver detalles en la descripción de esta sección). Se debe seleccionar un modo de búsqueda y escribir la palabra mediante la cual se buscará al usuario, (ej. Nombre). Dependiendo el tipo de búsqueda seleccionada, aparecerá, si existen coincidencias con la palabra utilizada, uno o más registros en existencia (similar a la Fig. 5.4 de Multas) con una breve descripción del usuario, en caso contrario aparecerá un mensaje (similar a la Fig. 5.5 de Multas). Después de ser desplegados los resultados de búsqueda, se presiona el botón que se encuentra junto al título  , donde se enviará a una página donde aparecerá la configuración actual de la opción seleccionada (similar a la Fig. 5.6 de Multas), sean Multas, Días o Límite de Libros o un Cero en caso de no contar con dicha configuración, Existen una serie de botones con diferentes propósitos que son **Regresar**, **Guardar**, **Borrar** y **Optimizar**, El Botón de **Regresar**, permite volver a la página de búsqueda, con los resultados de búsqueda anteriores. El botón de **Guardar** se encarga de guardar el cambio en la configuración del usuario seleccionado. El botón de **Borrar** se encarga de eliminar la configuración especial del usuario seleccionado, es decir que volverá a contar con la configuración general de todos lo demás usuarios de su perfil. El botón **Optimizar**, se encarga de eliminar registros que ya no se usarán más, optimizando así el rendimiento del sistema, se recomienda que después de realizar la acción de **Guardar** o **Borrar**, se utilice dicha acción.


Límite de libros (General)

Dentro del módulo **Configurar** se selecciona Límite de Libros, se elige el tipo de usuario (Alumno o Profesor) al cual se le asignará el número de Libros y se presiona el botón de **General**. Aparecerá un pantalla (similar a la Fig. 5.2 de Multas) donde se mostrará el título del módulo (Límite de Libros Prestados), el tipo de configuración (General) y el perfil que se va a configurar (Alumno o Profesor). Existen dos botones (Aceptar y Regresar) y dos campos de texto, el primero que tiene la leyenda Valor Actual, es el límite de libros que se está aplicando actualmente (si es la primera vez que se ejecuta esta configuración es posible que aparezca el valor de cero) y el campo Valor Nuevo, que es donde se le asignará el nuevo valor. Para cambiar la cantidad existente, se inserta un nuevo valor dentro del campo de Valor Nuevo y se presiona Aceptar (el sistema no actualizará ningún carácter que no sea numérico, enviando un mensaje de error si esto llegara a suceder) o Regresar si no se desea configurar el módulo. Si se presionó el botón de Aceptar, aparecerá una pantalla con un mensaje de error o éxito junto con un botón de volver, que nos enviará a la página inicial del módulo de **Configurar**.

Límite de libros (Individual)

Dentro del módulo **Configurar** se selecciona Límite de Libros, se elige el tipo de usuario (Alumno o Profesor) al cual se le asignará la cantidad y se presiona el botón de Individual. Aparecerá un pantalla de búsqueda (similar a la Fig. 5.3 de Multas) donde, por medio de 3 diferentes tipos de búsqueda (Clave de Usuario, Nombre, Apellido Paterno), se podrá localizar el usuario que se desea configurar, se mostrará además el título del módulo (Límite de Libros), el tipo de configuración (Individual) y el perfil que se va a configurar (Alumno o Profesor).

La pantalla de Búsqueda (Módulo Configuración) posee ciertas restricciones en su búsqueda (ver detalles en la descripción de esta sección). Se debe seleccionar un modo de búsqueda y escribir la palabra mediante la cual se buscará al Usuario, (ej. Nombre). Dependiendo el tipo de búsqueda seleccionada, aparecerá, si existen coincidencias con la palabra utilizada, uno o más registros en existencia (similar a la Fig. 5.4 de Multas) con una breve descripción del usuario, en caso contrario aparecerá un mensaje (similar a la Fig. 5.5 de Multas). Después de ser desplegados los

resultados de búsqueda, se presiona el botón que se encuentra junto al título , donde se enviará a una página en la que aparecerá la configuración actual de la opción seleccionada (similar a la Fig. 5.6 de Multas), sean Multas, Días o Límite de Libros o un Cero en caso de no contar con dicha configuración, Existen una serie de botones con diferentes propósitos que son **Regresar**, **Guardar**, **Borrar** y **Optimizar**, El Botón de **Regresar**, permite volver a la página de búsqueda, con los resultados de búsqueda anteriores. El botón de **Guardar** se encarga de guardar el cambio en la configuración del usuario seleccionado. El botón de **Borrar** se encarga de eliminar la configuración especial del usuario seleccionado, es decir que volverá a contar con la configuración general de todos lo demás usuarios de su perfil. El botón e **Optimizar**, se encarga de eliminar registros que ya no se usarán más, optimizando así el rendimiento del sistema, se recomienda que después de realizar la acción de **Guardar** o **Borrar**, se utilice dicha acción.

Suspensión del servicio

Dentro del módulo Configurar se selecciona Suspensión del Servicio, se elige el tipo de usuario (Alumno o Profesor) al cual se le suspenderá el servicio de biblioteca. En éste módulo no existe la característica General, ya que la suspensión se aplica única y exclusivamente a un usuario en particular por operación.

Aparecerá un pantalla de búsqueda (similar a la Fig. 5.3 de Multas) donde, por medio de 3 diferentes tipos de búsqueda (Clave de Usuario, Nombre, Apellido Paterno), se podrá localizar el usuario que se desea configurar, se mostrará además el título del módulo (Suspensión del Servicio) y el perfil del usuario que se va a suspender (Alumno o Profesor).

La pantalla de Búsqueda posee las mismas restricciones en su búsqueda (ver detalles en la descripción de esta sección). Se debe seleccionar un modo de búsqueda y escribir la palabra mediante la cual se buscará al Usuario, (ej. Nombre). Dependiendo el tipo de búsqueda seleccionada, aparecerá, si existen coincidencias con la palabra utilizada, uno o más registros en existencia (similar a la Fig. 5.4 de Multas) con una breve descripción del usuario, en caso contrario aparecerá un mensaje (similar a la Fig. 5.5 de Multas). Después de ser desplegados los resultados de búsqueda, se presiona el

botón que se encuentra junto al título  , donde se enviará a una página donde aparecerá el estatus actual del usuario seleccionado (Fig. 5.7).

En la siguiente página aparecerá una breve descripción de los datos personales del usuario, junto al Apellido Materno aparece como título el estado actual del usuario (Activo o Inactivo) y un checkbox (señalado en un óvalo) donde indica el estado inicial. Cada vez que se actualice el estado del usuario, la barra de título cambiará dependiendo si se esta o no seleccionado, en caso de que se encuentre seleccionado, indicará que el usuario tiene actualmente el servicio suspendido. Para cambiar la configuración, dependiendo si se desea Habilitar o Inhabilitar el servicio, se presiona el botón **Guardar** y el estado del título cambiará dependiendo la operación que se haya seleccionado. El Botón de **Regresar**, permite volver a la página de búsqueda, con los resultados de búsqueda anteriores.



Fig. 5.7

Agregar clasificaciones

Dentro del módulo Configurar se selecciona Agregar Clasificaciones, (en esta opción no aplican la opción de ningún tipo de perfil ni las opciones de Individual y General, ya que este tipo de configuración no repercute en ningún perfil de usuario) y se apreciará que las opciones de Perfiles se deshabilitarán y sólo estará disponible el botón de General, que su función es enviar al módulo correspondiente.

En la siguiente página (Fig. 5.8) aparecerá el título del módulo, debajo de este las tres operaciones básicas para el mantenimiento del módulo (Altas, Bajas y Cambios) con un *radio button* debajo de cada uno de ellos. La página inicial (antes de seleccionar alguna opción) aparece con el subtítulo de consulta, lo cual se indica cuando no se ha ejecutado ninguna operación.

Alta de clasificaciones

Se selecciona el *radio button* que se encuentra bajo el subtítulo *Agregar* y se presiona el botón de *Aceptar*. La pantalla se actualizará (Fig. 5.9) y aparecerá un campo de texto donde se introducirá la nueva clasificación que se agregará al sistema y así mismo, el botón con el título de Registrar se deberá presionar para actualizar el registro. Si el registro fue exitoso o erróneo mostrará un mensaje en pantalla dependiendo el estatus de la operación.

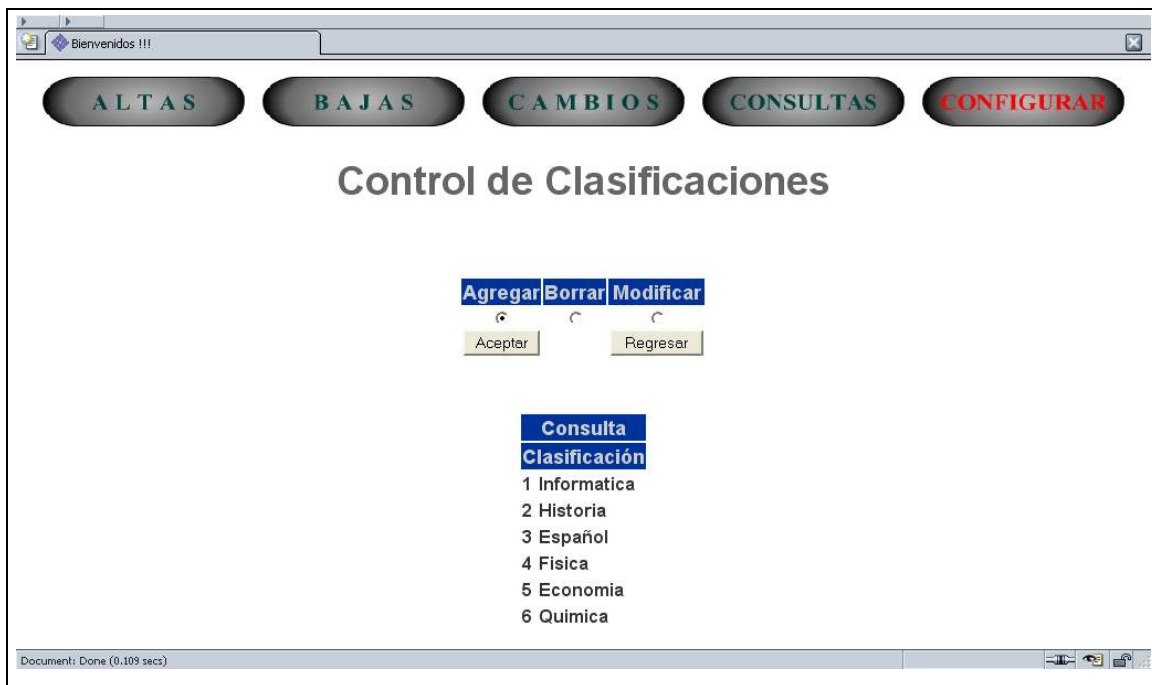


Fig. 5.8

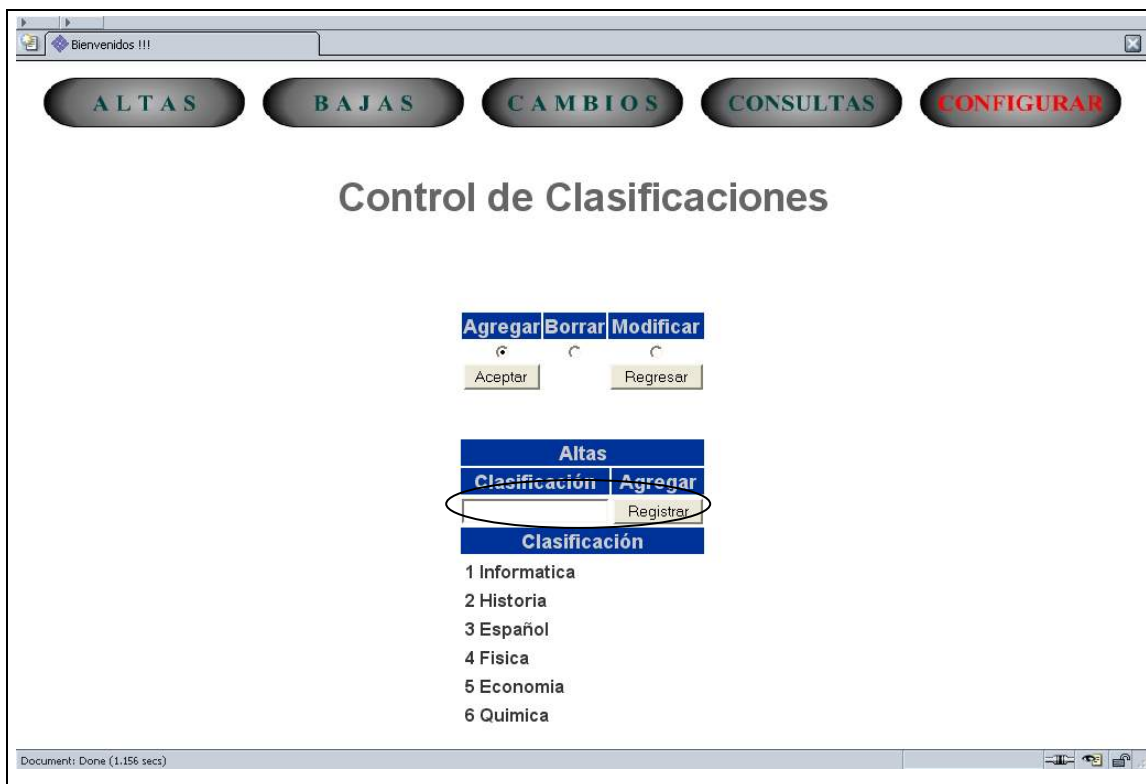


Fig. 5.9

Baja de Clasificaciones

Se selecciona el *radio button* que se encuentra bajo el subtítulo *Borrar* y se presiona el botón de *Aceptar*. La pantalla se actualizará (Fig. 5.10) y aparecerá la lista completa de todas las clasificaciones disponibles, con un botón del lado derecho de los registros, para elimina cualquier registro, se deberá presionar el botón que se encuentra junto a cada uno de los registros y aparecerá un mensaje de confirmación, sí realmente se desea eliminar el registro, se presiona *Aceptar* o *Cancelar* dependiendo la opción que se desee, sí se presiona *Aceptar*, aparecerá un mensaje dentro de la página donde señalara si la operación fue exitosa o errónea.

Cambio de Clasificaciones

Se selecciona el *radio button* que se encuentra bajo el subtítulo *Modificar* y se presiona el botón de *Aceptar*. La pantalla se actualizará (Fig. 5.11) y aparecerá la lista completa de todas las clasificaciones disponibles dentro de un campo de texto para su modificación, además de botón del lado derecho de los registros. Se modificará el texto dentro del campo de texto (*text field*) y al presionar el botón junto al registro aparecerá un mensaje de confirmación, donde el sistema preguntara si realmente se

desea modificar el registro, se presiona *Aceptar* o *Cancelar* dependiendo lo que se desea, si se presiona *Aceptar*, aparecerá un mensaje dentro de la página si fue exitoso o erróneo dependiendo el estatus de la operación. En la parte inferior de la pantalla aparece un botón con el título *Deshacer*, que cuando se presiona, vuelve al estado inicial todas las modificaciones (es decir al texto inicial previo).



Fig. 5.10



Fig. 5.11

3.8 Búsqueda

La sección de Búsqueda no es estrictamente un módulo, es un submódulo compartido por 3 módulos (Bajas, Cambios y Consultas). Existe además un submódulo de búsqueda dentro del módulo de Configuración.

Este submódulo está compuesto por un cuadro de texto, 2 botones (Buscar y Limpiar) y 3 diferentes formatos o tipos de búsqueda (Clave, Título y Autor o Clave de Usuario, Nombre y Apellido Paterno, dentro del Módulo de Configuración). Uno de los tipos de búsqueda es seleccionado, se deben insertar palabras o claves válidas para tener un mejor funcionamiento, dado que son diferentes tipos de búsquedas, se necesitan diferentes tipos de validaciones. Cuando se desee buscar por la Clave de Libro o Clave de Usuario, que es de tipo numérico, no se podrán insertar caracteres distintos a éste, (letras o símbolos), ya que el sistema enviará un mensaje de error y no procederá la búsqueda; así mismo en los otros dos formatos de búsqueda (Título y Autor o Nombre y Apellido Paterno, dentro del Módulo de Configuración). Se podrá insertar cualquier tipo de carácter, el sistema omitirá los acentos al momento de su registro, es por ello que cuando se busque una palabra con acento, el sistema no encontrará coincidencias.

en el registro, así como otros caracteres especiales. Se presiona el botón de **Buscar** después de introducir una palabra clave, si se presiona con espacio en blanco, el sistema detendrá la búsqueda y enviará un mensaje de error. Si se desea limpiar el contenido de la pantalla se debe presionar el botón **Limpiar**, el cual borrará los resultados de búsqueda (si existen desplegados en pantalla) y eliminará cualquier palabra escrita en el cuadro de texto.

4 Conclusiones

Resultados obtenidos

A continuación se hace un recuento general de los resultados que se obtuvieron en el presente proyecto. Tras implementar las herramientas de java (Servlets, JDBC) para la creación de páginas Web dinámicas (JSP), la mayoría de los requerimientos establecidos se cumplieron. Del mismo modo, se facilitó la administración de los procesos obteniendo así una mejor distribución de tareas reutilizando el código fuente de una manera más efectiva. Los costos se redujeron al eliminar el pago de licenciamiento de software al utilizar código abierto u *Open Source* como la base de datos, el lenguaje de programación, el IDE y el Web Application. Finalmente se creó un producto portable, independiente del sistema operativo, al utilizar clases compiladas de Java. En contraste, la base de datos, el IDE y el Web Application necesitan de una instalación totalmente separada ya que son generados para plataformas y sistemas operativos específico. El Sistema creado es portable de un sistema operativo a otro, pero en dicho sistema operativo se debe preparar el ambiente para la ejecución del sistema desarrollado.

Mejoras para el futuro.

Una de las mayores áreas de oportunidad para el óptimo sistema desarrollado es el tiempo de ejecución de los componentes de Java. El lenguaje de programación es lento al ejecutarse, tiene la característica de *compile once run everywhere* es un costo beneficio, esta situación podrá ser corregida en el corto o mediano plazo mediante un mejor hardware. Ello disminuirá el tiempo de respuesta de Java o cualquier otra aplicación.

Nuevas líneas de investigación.

Como se ha mencionado en el resto del documento, los JSP's son útiles para la creación de páginas Web dinámicas. sin embargo, en ocasiones el desarrollo basado en esta tecnología es largo, para mitigar esto existen alternativas disponibles en el mercado como la de crear etiquetas personalizadas (Custom Tags) que facilitan el desarrollo.

Comentarios finales.

Un aspecto importante referente al uso de Java es que a pesar de que no es un lenguaje de programación fácil de aprender, sí ofrece una amplia gama de enfoques para el uso de su tecnología, (redes, multimedia, bases de datos, etc.), los cuales no es posible comprenderlos con facilidad si no se conocen los conceptos fundamentales del lenguaje. Si no se cuentan con dichos conocimientos, la creación de cualquier sistema o aplicación con Java, puede representar un gran problema, ya que el paso inicial consistiría en familiarizarse con los conceptos básicos del lenguaje.

Java es un lenguaje *Open Source* muy popular en la actualidad. Se le da soporte continuo por una amplia base de desarrolladores, lo que garantiza cierta estabilidad en la calidad de sus productos. Al utilizar java, se tiene el respaldo de un lenguaje robusto, muy capaz de adaptarse a la mayoría de las necesidades.

Existen herramientas disponibles para facilitar el desarrollo y empleo de Java, tales como IDE's, Web Application's, Web Services, etc., capaces de cumplir con diversos tipos de requerimiento (sistema de cualquier dimensión) y necesidades (costos, compatibilidad, portabilidad, etc.). Ello debido a que se puede confiar en el soporte que se le da al lenguaje de manera constante.

Con el uso de Java para desarrollar nuestra aplicación, se logró demostrar que con la tecnología JSP's se cumplieron las necesidades del proyecto propuesto. A pesar de que los requerimientos no representaban una gran complejidad, se cumplieron aspectos muy importantes que Java pudo cubrir sin mayor dificultad.

Java tiene interactividad con una amplia gama de bases de datos. Los proveedores de estas Bases de datos ofrecen de manera publica un Driver (clases hechas con Java) que se encargan realizar las interactividad entre ambas partes. Esto en turno brinda a Java la capacidad de conexión con casi cualquier base de datos de uso comercial u *Open Source*, lo cual puede facilitar la adopción de la tecnología de una base de datos de alguna marca a otra de mayor capacidad o desempeño y vice versa. Lo anterior no implicaría un cambio sustancial en la aplicación, solo la configuración de los parámetros de conexión. En nuestra aplicación se opto por utilizar como base de datos la distribución de MySQL, por ser de tipo gratuito, haciendo disponible el Driver que

realiza la conectividad con la misma. Si surgiera la necesidad de hacer un migración a una base de datos distinta a la utilizada, solo sería necesario cambiar el Driver que se utilizar para la conexión, configurar al nueva conexión y sería suficiente sin la necesidad de hacer un recompilado completo del sistema, esta es una gran ventaja que Java ofrece.

A pesar de que existan tecnologías que presenten la misma funcionalidad los JSP utilizan un lenguaje orientado a objetos, no de Scripts, es decir que exportan todas las funcionalidades de Java para una página Web.

Java por si solo es un lenguaje de programación versátil y robusto en cuanto a las características que ofrece. Si aunamos todo esto a un sistema basado en generación de páginas dinámicas que implementen esta tecnología, se obtiene un producto estable y escalable. Esto hace que el tiempo de ejecución adicional para sistemas que no son de misión crítica pueda ser aceptable en primera etapa y superados mejorando la calidad de hardware en segunda etapa. Con lo anterior en mente, el uso de la tecnología Java para la creación de páginas Web dinámicas es una opción recomendable, razón por la cual esta tecnología es la de mayor demanda en el mercado.

ANEXO A

Acrónimos

ARPANET	Advanced Research Project Agency
API	Application Programming Interface
ARPA	Advanced Research Project Agency
ASP	Active Server Pages
ATM	Assincronus Transfer Mode
B2B	Business to Business
B2C	Business to Consumer
B2G	Business to Government
C2B	Consumer to Business
C2C	Consumer to Consumer
C2G	Consumer to Government
CCIRN	Coordinating Committee for Intercontinental Networks
CGI	Common Gateway Interface
CREN	Corporation for Research and Educational Networking
DARPA	Defense Advanced Research Project Agency).
DNS	Domain Name System
DOM	Document Object Model
EDI	Electronic Data Interchange
EFT	Electronic Fonts Transfer
FNC	Federal Networking Council
FTP	File Transfer Protocol
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	HiperText Transfer Protocol Secure
IAB	Internet Activities Board
IANA	Internet Assigned Numbers Authority
ICCC	International Computer Communication Conference
IETF	Internet Engineering Task Force
IIS	Internet Information Server
IMP	Interface Message Processors
ING	Internet New Generation
IP	Internet Protocol
IR	Internet Registry

ISP	Internet Service Provider
JSP	Java Server Pages
JVM	Java Virtual Machine
JWSDP	Java Web Service Development Pack
LAN	Local Area Network
LDAP	Light Directory Access Protocol
MIT	Massachusetts Institute of Technology
NAC	Network Analysis Corporation
NC	Network Computers
NCP	Network Control Protocol
NIC	Network Information Center
NMT	Network Measurement Center
NPL	National Physic Lab
NSF	National Science Foundation
ODBC	Open Data Base Connectivity
OSI	Open Systems Interconnection
P2P	Peer to Peer
PHP	Personal Home Pages
PWS	Personal Web Server
RARE	Reseaux Associees pour la Recherche Europeenne
RFC	Requests For Comment
SET	Secure Electronic Transaction
SRI	Stanford Research Institute
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLD	Top Level Domains
TPV	Terminal Punto de Venta
UCSB	UC Santa Barbara
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAN	Weight Area Network
WAP	Wireless Application Protocol
WML	Wireless Markup Language
WWW	World Wide Web

ANEXO B

Código fuente seleccionado.

```
/**
 * Propiedades.java
 * Created on March 28, 2005, 5:39 PM
 */
package edu.util;

import java.util.*;

/**
 * @author Sandor Trujillo Q.
 */
public class Propiedades{
    private HashMap hmProps;
    private String cKey;
    private int iSize;

    /**
     * Creates a new instance of Propiedades
     * @param cPropsFile
     */
    public Propiedades(String cPropsFile) {
        try{
            ResourceBundle rb = ResourceBundle.getBundle(cPropsFile);
            if(rb != null ){
                Enumeration en = rb.getKeys();
                HashMap htKeys = new HashMap();
                while(en.hasMoreElements()){
                    String cProp = en.nextElement().toString();
                    htKeys.put(cProp, rb.getString(cProp));
                }
                this.setISize(htKeys.size());
                this.setHmProps(htKeys);
            }
        } catch(Exception ex){
            ex.printStackTrace();
        }
    }
}
```

Código fuente perteneciente a la clase que se encarga de extraer propiedades de un archivo *“.properties”* que sirven para configurar el acceso a la base de datos y otras funciones.

```

/**
 * ChangeValues.java
 * Created on 19 de diciembre de 2004, 11:04 AM
 */
package edu.util;
/**
 * @author Sandor Trujillo Q.
 */
public class ChangeValues {
    private String cValue;
    /** Creates a new instance of changeValue */
    public ChangeValues(String s){
        try{
            if(s != null && s.trim().length() > 0 ){
                s = s.replace('á', 'a');
                s = s.replace('é', 'e');
                s = s.replace('í', 'i');
                s = s.replace('ó', 'o');
                s = s.replace('ú', 'u');
                s = s.replace('Á', 'A');
                s = s.replace('É', 'E');
                s = s.replace('Í', 'I');
                s = s.replace('Ó', 'O');
                s = s.replace('Ú', 'U');
                s = s.replace('ü', 'u');
                s = s.replace('Û', 'U');
                s = s.replace('\\', '\\');
                cValue = s;
            }
            else
                cValue = "";
        }
        catch (NullPointerException npe){
            npe.printStackTrace();
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }
    /**
     * Getter for property cValue.
     * @return Value of property cValue.
     */
    public String getCValue() {
        return cValue;
    }
}

```

Código que se encarga de reemplazar caracteres no permitidos en la base de datos.

```

/**
 * SecureCodify.java
 * Created on 19 de diciembre de 2004, 01:50 PM
 */

package edu.util.security;

/**
 * @author Sandor Trujillo Q.
 */
public class SecureCodify {

    private String cValue;
    private String a;
    private String b;
    private int i;
    private int j;

    public String Codify(String s) {
        try{
            a = s.substring(0,5);
            b = s.substring(5,s.length());
            i = (int)Long.parseLong(a,36);
            j = (int)Long.parseLong(b,36);
        }
        catch(Exception ex){
            i = 0;
            j = 0;
            ex.printStackTrace();
        }
        finally{
            return this.cValue = ""+i+j;
        }
    }

    public String Decodify(String s){
        a = s.substring(0,8);
        b = s.substring(8,s.length());
        i = Integer.parseInt(a);
        j = Integer.parseInt(b);
        return this.cValue = ""+Integer.toString(i,36)+Integer.toString(j,36);
    }
}

```

Código fuente perteneciente a la clase codificadora y decodificadora de Password, con un principio de conversión numérica en base 36.

```
#####  
##### propsMySQL.properties #####  
#####  
#parámetro del Driver de la base de datos  
cDriver=com.mysql.jdbc.Driver  
#nombre del servidor de la base de datos  
cServerName=localhost  
#nombre de la base de datos  
cDataBase=biblioteca  
#número del puerto de la base de datos  
cPort=3306  
#login de la base de datos  
cLogin=root  
#password de la base de datos  
cPassword=nemesis  
#parametro con la clave del perfil de Administrador  
Administrator=1
```

Archivo de propiedades con datos que ayudan a la configuración de la base de datos.

Referencias de Información.

Referencia Bibliográfica.

[1] Akif, Mohammad y colaboradores. (2002) .

Java y XML, Referencia Para Programadores.

España: Anaya Multimedia.

[2] Bonson, Enrique. (1995).

World-Wide-Web y el ciberespacio económico-financiero.

Madrid: Ra-Ma.

[3] Bonson, Enrique. J. Sierra, Guillermo. (1997).

Internet empresarial.

Madrid: Ra-Ma.

[4] C. Laudon Kenneth. Guercio Traver, Carol. (2002).

E-commerce: business, technology, society.

Boston: Addison Wesley.

[5] Carpintier Santana, Rodolfo. (1997).

Internet hoy : como hacer negocios en la red.

Madrid : Ra-Ma.

[6] Coupey, Eloise. (2001).

Marketing and the internet.

New Jersey: Prentice Hall.

[7] D. Monnier, Philippe. (1999).

Cybermarketing: a guide for managers in developing countries.

Geneva: International Trade Centre.

[8] D. Speegle, Gregory. (2002).

JDBC: Practical Guide For Java Programmers.

USA: Morgan Kaufmann Publishers.

- [9] Deitel, H.M., Deitel, P.J. (1998).
Como programar en Java.
México: Pearson Educación. Segunda edición.
- [10] Deitel, H.M., Deitel, P.J. (1999).
Java 2: How To Program.
USA: Prentice Hall. Tercera edición.
- [11] DuBois, Paul. (2001).
MySQL: Edición Especial.
Madrid: Pearson.
- [12] Goodman, Danny. (2001).
JavaScript Bible, Gold Edition.
USA: Hungry Minds, Inc.
- [13] Hair McDaniel, Lamb. (2002).
Marketing,
Thomson. Sexta edición.
- [14] Hall, Marty. (2001).
Servlets y JavaSever Pages, Guía Practica.
México: Prentice Hall.
- [15] Hanna, Phil. (2002).
JSP Manual de Referencia.
España: Mc Graw Hill.
- [16] Hostman, Cay S., Gary Cornell, (2003).
Java 2 Volumen 1, Fundamentos.
Madrid: Prentice Hall.
- [17] Houlette, Forrest. (2002).
Resolución de Problemas en SQL.
España: Mc Graw Hill-Osborne Media.

- [18] Kalakota, Ravi. B. Whinston, Andrew. (1996).
Frontiers of the electronic commerce.
Massachusetts: Addison-Wesley.
- [19] Reese, George. (2000).
Database Programming with JDBC & Java.
USA: O'Reilly. Segunda edición.
- [20] S. Janal, Daniel. (1996).
101 Businesses You Can Start on the Internet.
Van Nostrand Reinhold Company.
- [21] Schildt, Herbert. (2001).
Java: Manual de Referencia.
España: Mc Graw Hill. Cuarta edición.
- [22] Sierra, Kathy. Bates, Bert. (2003).
Java 2, SUN Certified Programmer & Developer.
USA: Mc Graw Hill-Osborne.
- [23] Silberschatz, Abraham. (1998).
Fundamentos de Bases de Datos.
Mc Graw Hill, tercera edición.
- [24] V. Lesikar, Raymond. E. Flatley, Marie. (2002).
Basic business communication skills for empowering the internet generation.
Boston: McGraw-Hill/Irwin.

Referencia Electrónicas

- [1] Arnaud Le Hors, Philippe Le Hégarret, Lauren Wood, Gavin Nicol, Jonathan Robie, Mike Champion, Steve Byrne. **Document Object Model (DOM) Level 3 Core Specification.** [en línea] W3C Recommendation, 7 Abril 2004.
<http://www.w3.org/TR/DOM-Level-3-Core/>

- [2] Danny Coward, Yutaka Yoshida. **Java Servlet Specification Version 2.4**. [en línea], Sun Microsystems Proposed Final Draft, 5 agosto de 2002, disponible en WWW <http://www.jcp.org/aboutJava/communityprocess/final/jsr154/index.html>
- [3] Dave Raggett, Arnaud Le Hors, Ian Jacobs. **HTML 4.01 Specification**. [en línea], W3C Recommendation, 24 de diciembre 1999, disponible en WWW <http://www.w3.org/TR/REC-html40/>.
- [4] Eric Armstrong, Stephanie Bodoff, Debbie Carson, Maydene Fisher, Scott Fordin, Dale Green, Kim Haase, Eric Jendrock. **The Java Web Services Tutorial**. [en línea] Sun Microsystems, Febrero 2003. <http://Java.sun.com/webservices/tutorial.html>
- [5] **JavaScript Reference**, [en línea], Netscape Communications Corporation, 12 de diciembre de 1997, disponible en WWW <http://www.lecad.uni-lj.si/documents/vaje/js-tecaj/js/jsref.pdf>
- [6] Jon Ellis, Linda Ho, Maydene Fisher. **JDBC 3.0 Specification**. [en línea], Sun Microsystems Final Release 1 de diciembre 2001, disponible en WWW <http://Java.sun.com/products/jdbc/download.html>
- [7] Mark Roth, Eduardo Pelegrí-Llopart. **JavaServer Pages Specification Version 2.0**. [en línea], Sun Microsystems proposed final draft 3, 11 de abril 2003, disponible en WWW <http://Java.sun.com/products/jsp/>
- [8] **MySQL Reference Manual**, [en línea], MySQL com, disponible en WWW <http://dev.mysql.com/doc/>
- [9] Stig Sether Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski, Andrei Zmievski, Jouni Ahto, Rafael Martínez. **Manual de PHP**. Grupo de documentación de PHP. redistribuido bajo los términos de la "GNU General Public License". 15 de abril de 2001.

Referencia de Internet (ISO 690-2).

- [1] <http://www.webmasterbase.com/article.php> [18/02/2004]
- [2] Copyright 1994-2006 Sun Microsystems, Inc., JavaServer Pages Technology, [en línea] <http://Java.sun.com/products/jsp/>
- [3] Copyright 1994-2006 Sun Microsystems, Inc., Java Servlet Technology, [en línea] <http://Java.sun.com/products/servlet/>
- [4] Copyright 1994-2006 Sun Microsystems, Inc. Java Web Services Developer Pack . [en línea] <http://Java.sun.com/webservices/downloads/webservicespack.html>
- [5] 1995-2006 MySQL AB. MySQL Database Products, [en línea] <http://www.mysql.com/products/>
- [6] Microsoft Corporation, Internet Information Services, [en línea] <http://www.microsoft.com/WindowsServer2003/iis/default.msp>
- [7] Microsoft Corporation. ASP [en línea] <http://www.asp.net/Default.aspx?tabindex=0&tabid=1> [12/04/2004]
- [8] <http://ecommerce.about.com/cs/ecommerce/f/FAQ001.htm> [10/02/2004]
- [9] FastWebServer.com, Domain Registration [en línea] http://www.fastwebserver.com/e_commerce_faq.htm [10/02/ 2004]
- [10] aece.org, Comercio Electrónico. [en línea] <http://www.aece.org/aece.asp> [10/02/2004]
- [11] Internet Society. Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, Stephen Wolff. A Brief History of the Internet. [en línea] <http://www.isoc.org/internet/history/index.shtml> [29/02/2004]
- [12] Adobe Systems Incorporated..Macromedia JRun 4 [en línea] <http://www.macromedia.com/software/jrun/>
- [13] World Wide Web Consortium. HTML [en línea] <http://www.w3c.com/>

- [14] Asoc. Española de Comercio Electrónico-CommerceNet Español. Comercio Electrónico [en línea] <http://www.commercenet.org> [10/02/2004]
- [15] IBM Corporate. WebSphere Software [en línea] <http://www-306.ibm.com/software/websphere/>
- [16] The Apache Software Foundation. Apache Tomcat. [en línea] <http://tomcat.apache.org/>