



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

## FACULTAD DE INGENIERÍA

# MANEJO DE DATOS A TRAVÉS DEL WORLD WIDE WEB

## TESIS

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN

PRESENTAN

**JUAN MANUEL CASTILLEJOS REYES**  
**MILTON ARTURO GARCÍA LIMA**  
**HÉCTOR ZAVALA VEGA**

DIRECTOR DE TESIS

**ING. ALBERTO GONZÁLEZ GUIZAR**

CIUDAD UNIVERSITARIA

OCTUBRE DE 2004





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.

---

# Agradecimientos

**Por nuestra UNAM.**

**Dios, Patria y Vida**

A nuestros Padres, hermanos y amigos,

A nuestros profesores y compañeros,

A todos aquellos que creyeron y que comparten con nosotros este tiempo de vida.

Las guerras se ganan con soldados cansados...

*México, Ciudad Universitaria a 4 de Octubre de 2004*

---

---

---

# Índice

Pag.

Introducción .....	XI
<b>Capítulo 1: Antecedentes y conceptos básicos .....</b>	<b>1</b>
<b>1.1 Internet .....</b>	<b>1</b>
1.1.1 México e Internet .....	6
1.2 Lenguaje Estándar Generalizado de Marcado (SGML) .....	7
1.3 Lenguaje de Marcado de Hipertexto (HTML) .....	10
1.4 Servidores Web .....	13
1.5 Sistema administrador de bases de datos relacionales (RDBMS) .	15
1.6 Las Bibliotecas Digitales .....	16
1.7 El documento electrónico .....	18
<b>Capítulo 2: XML en el mundo .....</b>	<b>21</b>
<b>2.1 El nuevo estándar de Internet .....</b>	<b>23</b>
2.1.1 Características generales .....	25
<b>2.2 Documentos XML .....</b>	<b>28</b>
2.2.1 Estructura de los documentos XML .....	30
2.2.2 Los esquemas y el documento de tipo de Dato (DTD).....	36
<b>2.3 Negocios en la WEB .....</b>	<b>49</b>
2.3.1 Intercambio Electrónico de Datos (EDI) .....	49
2.3.2 Comercio Electrónico .....	49
2.3.3 Traducción .....	50
2.3.4 Ayuda de escritorio (soporte técnico) .....	50
<b>2.4 Perspectivas para XML .....</b>	<b>51</b>
2.4.1 Servicios Web en el futuro .....	51
<b>Capítulo 3: JAVA .....</b>	<b>53</b>
<b>3.1 La creación de Java .....</b>	<b>53</b>
<b>3.2 Características de Java .....</b>	<b>55</b>
3.2.1 Sencillo .....	55
3.2.2 Orientado a objetos .....	57
3.2.3 Distribuido .....	57

---

---

3.2.4	Robusto .....	58
3.2.5	Arquitectura neutral .....	59
3.2.6	Seguridad .....	60
3.2.7	Portabilidad .....	64
3.2.8	Interpretado .....	66
3.2.9	Multihilo .....	67
3.2.10	Dinámico .....	67
3.3	Herramientas de desarrollo .....	70
3.3.1	Integración con otras aplicaciones .....	71
3.4	Java Script .....	72
3.5	El Futuro de Java .....	72
Capítulo 4: Evaluación de Herramientas .....		77
4.1	La importancia de XML .....	78
4.1.1	Alternativas de desarrollo para la aplicación .....	78
4.1.2	Evaluación de XML .....	81
4.1.3	Ventajas de XML para servidores Web .....	82
4.1.4	Rendimiento máximo .....	83
4.1.5	Personalizable .....	84
4.2	El nuevo estándar en el intercambio de datos .....	85
4.3	¿Por qué utilizar XML? .....	85
4.4	¿Porqué JAVA? .....	86
Capítulo 5: Arquitectura y Diseño .....		89
5.1	Arquitecturas cliente-servidor .....	89
5.2	Diseño orientado a objetos .....	98
5.3	Objetos y clases de objetos .....	100

---

<b>Capítulo 6: Análisis y diseño de un Sistema de publicación en línea. ....</b>	<b>103</b>
6.1 Manejo de datos aplicados a un modelo real. ....	103
6.2 Marco de referencia tecnológica y derecho de autor .....	104
6.3 Requerimientos del Sistema .....	107
6.4 Descripción general del Sistema .....	108
6.5 Análisis del proceso mediante casos de uso .....	111
6.6 Diagrama de casos de uso .....	128
6.7 Diagrama de clases .....	129
6.8 Diagramas de colaboración .....	132
6.9 Diagramas de secuencia .....	135
6.10 Requerimientos de hardware y software .....	142
6.11 Alcances .....	143
6.12 Exclusiones .....	143
6.13 Posibilidades de expansión .....	144
6.14 ¿Quién puede utilizar el sistema? .....	144
6.15 Base de datos .....	144
6.16 Diagrama del sitio .....	146
<b>Conclusiones .....</b>	<b>147</b>
<b>Manual de Usuario .....</b>	<b>151</b>
<b>Glosario de Términos .....</b>	<b>179</b>
<b>Referencias Bibliográficas .....</b>	<b>187</b>

---

# Introducción

El hombre desde que se concibe como tal, ha buscado la mejor forma de comunicarse con sus semejantes. En la prehistoria, los primeros hombres se comunicaron a través de señas, sonidos guturales, danzas y rituales; siendo las pinturas rupestres el medio gráfico más antiguo conocido en el que los primeros hombres lograron preservar sus conocimientos a través de las generaciones.

Conforme el ser humano empieza a evolucionar y a tener mayor necesidad de establecer una comunicación entre sus semejantes, se idearon medios que permitieran el establecimiento de una comunicación efectiva. Por tanto los primeros medios de comunicación con otros grupos humanos fueron los mensajeros que recorrían grandes distancias para entregar los mensajes de manera oral.

De esta manera poco a poco se desarrollaron formas de comunicación cada vez más sofisticadas, quedando al descubierto dos características importantes que definimos como el ciclo de la comunicación / cooperación. Sin la existencia de éstas, no sería posible el entendimiento humano:

**COOPERACIÓN ← → COMUNICACIÓN**

La evolución del hombre prosiguió y con ello, la llegada de grandes descubrimientos e invenciones como la manipulación del fuego, la recolección de comida que se convirtió en la agricultura, la cacería que además se derivó en la ganadería, la rueda, entre muchos más, pero el gigantesco salto de la civilización fué la aparición de la pintura y después de la escritura que no son mas que otro medio de comunicación en el que se depositan ideas y las abstracciones del mundo con el fin de ser almacenadas y preservadas para el futuro.

Desde su aparición, la escritura cobró una gran importancia ya que en los primeros edificios y tumbas se narra la vida de los altos gobernantes y jerarcas como testimonio de vida y perpetuidad. Poco después, sirve como medio de difusión de tecnología y con el nacimiento de las matemáticas, como medio de control económico y fiscal.

Ya mucho mas adelante con el establecimiento de ciudades, el nacimiento de las ciencias y el comercio, nacen las primeras bibliotecas que son recintos donde se almacenan ya sean las tablillas grabadas o papiros, la información valiosa de las primeras sociedades humanas.

La cultura griega y posteriormente la romana utilizaron la escritura como medio formal para imprimir el pensamiento filosófico, político, social y económico. De ahí el hecho de que se preserva hasta nuestros días el conocimiento por ejemplo de la democracia Griega y del derecho romano.

En la China antigua nace la tinta y el papel, en la India, además se inventa el ábaco que no es mas que un medio eficiente para hacer cuentas y así una serie de inventos que ayudaron a la abstracción de ideas y pensamientos con el fin de ser conservados.

Las culturas prehispánicas tuvieron avances increíblemente notables. Además del comercio y el derecho, descubrieron las ciencias exactas y gracias a la astronomía calcularon los movimientos de los astros, construyendo calendarios precisos y exactos. Para mantener una buena comunicación la sociedad azteca por ejemplo implementó un sistema complejo de logística basado en mensajeros que recorrían grandes distancias en poco tiempo.

En la cultura occidental, es en el medioevo en donde surgen los libros escritos. Esta ardua tarea era asignada solo a los religiosos quienes mantenían el privilegio de ser los únicos con derecho a educación. Los religiosos por ejemplo lograron



traducir el pensamiento Griego al latín, así como el pensamiento matemático de la cultura árabe entre otros.

Las guerras e invasiones siempre han ayudado a expandir no solo el dominio de los pueblos, sino también el pensamiento y la tecnología de los mismos. Es hasta la invención de la imprenta de Gutenberg cuando se incrementa en gran medida la manera de elaborar y preservar información escrita. La comunicación escrita fue en gran medida la que logró la conquista de las tierras más lejanas a través de cartas cartográficas y mapas, así como la preservación del pensamiento humano en todas sus expresiones y matices a través de los siglos.

Ya comenzado el siglo XIX con la revolución industrial y los avances tecnológicos, comenzaron los estudios formales de la electricidad que agregaron el cúmulo de innovaciones tecnológicas más importantes de la época propiciando el ambiente perfecto para la creación de inventos como el telégrafo en el año de 1844, el teléfono en 1876, las primeras máquinas mecanógrafas en el año de 1886.

En el año de 1925 se desarrolla la válvula de filamento que da origen al primer radio y posteriormente a las máquinas de cálculo que ocuparon los primeros bulbos y que derivaron en lo que ahora conocemos como las computadoras. Todos estos inventos y otros más conforman la base de todos aquellos adelantos tecnológicos con los que contamos hoy en día en cuanto a comunicación se trata.

En el siglo pasado, las dos guerras mundiales prometieron grandes avances en los sistemas de comunicación. Así nacen las primeras redes que sirvieron como punta de estrategia ofensiva. Los ejércitos alemanes, británicos, americanos y japoneses, desarrollaron sistemas de comunicación cifrada o de coordinación radiofónica entre sus infanterías, aviación, buques y submarinos de guerra.

La preservación del conocimiento humano ha trascendido de generación en generación que va desde los primeros sonidos guturales, el nacimiento de la

escritura hasta la llegada de la era digital. Parece ser que uno de los requisitos esenciales para hacerse llamar civilización, es la preservación de los conocimientos. El fin no ha cambiado, pero el medio evolucionó desde las tablillas de arcilla hasta los documentos electrónicos muy sofisticados de hoy.

Desde nuestro contexto temporal y social, nosotros hemos desarrollado éste trabajo de tesis que pretende detallar algunas consideraciones para el manejo y manipulación de la información desde el punto de vista tecnológico en lo que aparenta ser la nueva imprenta: Internet, sus aplicaciones y como una de sus consecuencias, el futuro de los libros.

---

# Capítulo 1

## ANTECEDENTES Y CONCEPTOS BÁSICOS

### 1.1 Internet

En 1947 se construyó en la universidad de Pennsylvania la ENIAC que fue la primera computadora electrónica, donde los ingenieros John Mauchly y John Eckert encabezaron el equipo de diseño, esta máquina ocupaba todo un sótano de la universidad, tenía más de 18,000 tubos de vacío, consumía 200 KW de energía eléctrica y requería todo un sistema de aire acondicionado, pero tenía la capacidad de realizar cinco mil operaciones aritméticas en un segundo.

El proyecto, auspiciado por el departamento de defensa de los Estados Unidos, culminó dos años después, cuando se integró a ese equipo el ingeniero y matemático húngaro John Von Neumann (1903 - 1957). Las ideas de Von Neumann resultaron tan innovadoras para su desarrollo, que es considerado el padre de las computadoras. La EDVAC tenía aproximadamente cuatro mil bulbos y usaba un tipo de memoria basado en tubos llenos de mercurio por donde circulaban señales eléctricas sujetas a retardos.

Durante la segunda guerra mundial, un hombre llamado Vannevar Bush facilitó las relaciones entre el gobierno federal de los Estados Unidos, la comunidad científica americana y los empresarios. Después de la guerra, ayudó a institucionalizar esta relación, como resultado fueron creadas asociaciones tales como la Fundación Nacional de la Ciencia "NSF", y la Agencia de Proyectos Avanzados de Investigación *ARPA*, en ésta última empezó Internet.

La Agencia de Proyectos avanzados de Investigación *ARPA* fue creada por el presidente Dwight Eisenhower después de que los soviéticos lanzaran el satélite *Sputnik* en octubre de 1957.

En 1962, J.C.R. Licklider fue a trabajar para ARPA. Éste era un psicólogo e informático, creía que las computadoras se podrían utilizar para aumentar el pensamiento humano y sugirió que fuera establecida una red de computadoras para permitir a los investigadores de ARPA comunicarse entre ellos de modo eficiente. Licklider no construyó realmente la red que propuso, pero su idea siguió viva cuando dejó ARPA en 1964.

El análisis para la implantación de este tipo de redes comenzó en 1967 en un simposium realizado en Inglaterra sobre principios operativos, organizado por ACM, como resultado de este tipo de estudios en el año de 1968 se implementó la primer red de este tipo en Gran Bretaña utilizando un mainframe IBM.

En 1969 ARPA y la agencia del Pentágono, comienzan un proyecto que desarrollarían RAND, MIT y UCLA para crear una red con las características que tanto se deseaban alcanzar; de tal manera, que el primer nodo de esta red fue implantado en UCLA y para diciembre del mismo año existían ya cuatro nodos más en ARPANET, en 1971 quince nodos constituían ARPANET y para el año de 1972 ya existían treinta y siete nodos que conformaban la red, todos ellos con el fin común de establecer la comunicación entre proyectos y trabajos entre investigadores; también permitió el intercambio de mensajes por correo electrónico, creando los primeros grupos de noticias que conocemos actualmente.

Los años 70 transcurrieron con instituciones que se conectaban directa o indirectamente las primeras redes como ARPANET. Los responsables de esta tecnología desarrollaron estándares y protocolos, como Telnet, la especificación de transferencia de archivos o el protocolo de voz en redes *NVP*. Bob Metcalfe inventó Ethernet, y Douglas Englebart, inventó el mouse entre otras cosas. Otras redes de computadoras como la hawaiana ALOHANET y la red enlazada de

satélites, SATNET, empezaron a crearse. Pronto había muchas redes diferentes alrededor del mundo, pero no podían comunicarse con otras porque utilizaban protocolos o estándares para transmisión de datos diferentes.

ARPANET demostró ampliamente sus características en 1973, al llevar a cabo la primera conferencia internacional de ésta, conectando 40 máquinas en algunas partes del mundo permitiendo intercambio de información sin ninguna falla.

Para el año de 1974 el NCP fue reemplazado por un nuevo estándar o norma, el TCP/IP (Transfer Communication Protocol/ Internet Protocol); TCP tiene la capacidad de partir los mensajes en tramas con una dirección destino y su dirección de origen, e IP es el encargado del direccionamiento de dichos paquetes a través de los nodos de la red.

En 1979 ARPA crea la primera comisión de control de la configuración de Internet y tras varios años de trabajo, en 1981 se termina de definir el protocolo TCP/IP y ARPANET lo adopta como estándar en 1982, sustituyendo a NCP. Son las primeras referencias a Internet, como *"una serie de redes conectadas entre sí, específicamente aquellas que utilizan el protocolo TCP/IP"*. Internet es la abreviatura de *Interconnected Networks*, es decir, Redes Interconectadas, o red de redes. Además en estos años se fundan Microsoft (1975) y Apple (1976).

En 1983 ARPANET se separa de la red militar que la originó, de modo que ya sin fines militares se puede considerar esta fecha como el nacimiento de Internet. Es el momento en que el primer nodo, militar, se desliga dejando abierto el paso para todas las empresas, universidades y demás instituciones que ya por esa época poblaban la joven red. En este año se anuncia la primera versión del Windows de Microsoft, que saldría dos años más tarde.

En los años 80, la expansión es enorme, cada vez se conectan más máquinas a la red, y se van mejorando los servicios por ejemplo, el servidor desarrollado en

la universidad de Wisconsin ya no requiere que el usuario tenga que conocer la ruta exacta para acceder a otros sistemas. En 1985, quince años después de la primera propuesta, se termina el desarrollo del aún vigente protocolo para la transmisión de archivos en Internet FTP, basado en la filosofía de cliente-servidor. Un punto fundamental en el éxito fue el hecho de que ARPA distribuyera a bajo costo los protocolos, que fueron adoptados por el UNIX de BSD, muy difundido entre las universidades. De esta forma se crearon una gran cantidad de servicios y se provocó un importante avance en el desarrollo de la red. Por esta época se crea el Domain Name Server (DNS).

A partir de 1987 empezó la gran expansión, en parte debida a que el año anterior se creó la NSFNET, que estableció cinco centros de supercomputadoras para proveer un alto poder de proceso. Es ahora cuando se incorporan a Internet diversas redes de Europa. También en ese año encontramos la primera aplicación informática de hipertexto.

En los Estados Unidos el gran aumento de usuarios provocó en 1990 la retirada de la agencia ARPA, y su red pasó a estar a cargo de la NSF. Internet comenzó a saturarse y, para evitar el colapso, se restringieron los accesos. Eran años de incertidumbre ya que nadie había ideado la red para los fines y las dimensiones que se estaban alcanzando, y los responsables se veían abrumados, por tal motivo usando hipertexto, Tim Berners-Lee creó una nueva manera de interactuar con Internet en 1990: El WWW (*World Wide Web*) es un servicio de transmisión y presentación de documentos de hipertexto multimedia.

El WWW se desarrolló y por ende, existió la necesidad de crear nuevas tecnologías de software y hardware para convertirlo en una herramienta funcional. Por ejemplo, Marc Andreessen creó un nuevo navegador llamado *Mosaic* en 1993 y después dirigió al equipo que creó *Netscape Navigator*. Además, Berners-Lee, creó las bases del protocolo de transmisión HTTP, el lenguaje de documentos HTML y el concepto URL.

En septiembre de 1993 se inició el primer servidor Web en español. En estos momentos se aumenta la potencia de las redes troncales en los Estados Unidos, y en 1994 se eliminan las restricciones de uso comercial de la red y el gobierno de EE.UU. deja de controlar la información de Internet. En el año de 1995 es el gran "boom" de Internet, puede ser considerado como el nacimiento del Internet comercial. Desde ese momento el crecimiento de la red ha superado todas las expectativas.

Este hecho se produce porque es en este año cuando la WWW supera a *ftp-data* transformándose en el servicio más popular de la red, después de que el año anterior superase a *telnet*.

Es por ello que Internet resulta ser la infraestructura informática de comunicación e intercambio de información por excelencia más grande del mundo hoy día, su empleo se incrementa a diario y en las más diversas aplicaciones tal como la obtención de información de manera remota y el comercio electrónico, han sido las más explotadas actualmente.

Hoy día Internet es empleado en instituciones tanto educativas como gubernamentales, empresas privadas y por personas de todo el mundo, entre las que se llevan a cabo constantes intercambios de información, por ello gracias a Internet podemos recibir información de manera casi instantánea de cualquier parte del mundo, agilizando y facilitando el proceso de comunicación a distancia.

### **1.1.1 México e Internet**

#### **El primer nodo de Internet en México**

La historia del Internet en México empieza en el año de 1989 con la conexión del Instituto Tecnológico y de Estudios Superiores de Monterrey, en el Campus Monterrey, ITESM hacia la Universidad de Texas en San Antonio (UTSA), específicamente a la escuela de Medicina. Una línea privada analógica de 4 hilos a 9600 bits por segundo fue el enlace.

#### **Conexiones a BITNET en México**

Sin embargo, antes de que el ITESM se conectara a Internet, casi a final de los 80s, recibía el tráfico de BITNET por la misma línea privada. El ITESM era participante de BITNET desde 1986. Las conexiones se hacían a través de líneas conmutadas. La conexión permanente de esta institución se logró hasta el 15 de Junio de 1987 (a BITNET y posteriormente a INTERNET). La UNAM se conectó a BITNET en Octubre de 1987.

En noviembre de 1988 se cambia la conexión permanente que interconectaba equipo IBM con RSCS, a equipos DEC utilizando DECNET. Al cambiar el protocolo se tenía la posibilidad de encapsular tráfico de TCP/IP en DECNET y por lo tanto formar parte de Internet. Al siguiente año, en 1989, se cambió de una a tres líneas. Con ello, se cambió el equipo de interconexión y se incorporaron los equipos de ruteo CISCO.

En su mayoría las instituciones educativas fueron las más primeras a integrarse a Internet. A finales de 1995 se crea el Centro de Información de Redes de México (NIC – México) el cual es encargado de la coordinar y administrar los recursos de Internet asignados al país (todos aquellos dominios bajo “mx”).



Ya para el año de 1997 en México comenzaron a aparecer los ISP, de los cuales el más importante para esa fecha era TELMEX con 17 enlaces para uso privado. A finales del 1996, la apertura en materia de empresas de telecomunicaciones y concesiones de telefonía de larga distancia provoca un auge momentáneo en las conexiones a Internet. Empresas como AVANTEL y Alestra AT&T ahora compiten con TELMEX.

En 1997 existen más de 150 proveedores de acceso a Internet (ISPs) que brindan sus servicios en el territorio mexicano, ubicados en los principales centros urbanos: ciudad de México, Guadalajara, Monterrey, Chihuahua, Tijuana, Puebla, Mérida, Nuevo Laredo, Saltillo, Oaxaca, por mencionar sólo algunos.

Con Internet el ámbito comercial México tuvo un giro impresionante al igual que en el resto del mundo, con lo cual se inicia una nueva era de desarrollo para nuestro país que ha beneficiado a empresas e instituciones que han participado en proyectos que utilizan a Internet como base de su éxito en nuestro país.

## **1.2 Lenguaje Estándar Generalizado de Marcado (SGML)**

Hoy en día casi cualquier documento que leemos en nuestra vida diaria tiene algún tipo de estructura. La estructura de un texto la deducimos de forma implícita a partir de la distribución explícita del texto en la página. De esta manera nuestro cerebro asocia la apariencia de un texto con su estructura.

En ocasiones, es necesario hacer dichas diferencias un poco más explícitas, agregando información contextual al documento como son el “subrayado” de una palabra o un párrafo para denotar su importancia, haciéndolo más legible y comprensible a la vez un documento. Una versión electrónica de un documento, sin anotaciones de ningún tipo, es una simple cadena de caracteres. Sin embargo las computadoras no pueden extraer información estructural de dichos documentos y por lo tanto la versión impresa sería una copia idéntica de la cadena original sin

información, y por tal motivo serían equivalentes a las que se producirían con una máquina de escribir con un único tipo de letra.

Para ello los procesadores de texto agregan multitud de anotaciones a un documento que especifican las características de cada versión en papel ya que la función de un procesador de texto es asistir en la creación de documentos impresos. En este sentido la palabra *anotación* define todo aquello que forma parte de un documento electrónico pero que no es imprimible.

El Lenguaje Estándar Generalizado para Anotaciones [ISO8879] fue establecido por la ISO. En los años 60 IBM intentó resolver sus problemas asociados al manejo de datos en diferentes plataformas con GML. El principal problema era que cada aplicación utilizaba sus propias marcas para describir la información. Estas marcas, son códigos para indicar a un programa cómo debe tratar su contenido. La forma que IBM creó para solventar esto se basaba en tratar las marcas como texto accesible desde cualquier sistema, texto plano, código ASCII. Y la norma se denominó GML, que pasó a manos de ISO y se convirtió en el Lenguaje Generalizado de Marcado Estándar (SGML, ISO 8879). Esta norma es la que se aplica desde entonces a todos los lenguajes de marcas, cuyos ejemplos más conocidos son el HTML y el RTF.

SGML, es un metalenguaje con el cual se pueden definir lenguajes de anotación con los que se puede almacenar y procesar texto. Fue diseñado para permitir el intercambio de información entre distintas plataformas, soportes físicos, lógicos y diferentes sistemas de almacenamiento y presentación como las aplicaciones y sistemas que ocupan bases de datos, comercio electrónico, edición de documentos digitales, entre otros. SGML no es sólo un conjunto de códigos normalizados, sino que se convirtió en un lenguaje con el que se podía crear una definición del tipo de documento "DTD", mediante la cual se definen con precisión aquellos elementos que son necesarios en la elaboración de un documento o un grupo de documentos estructurados de manera similar.

SGML logró la independencia de la plataforma-procesador de texto que utiliza por lo general una organización. La transición de un procesador de textos de un fabricante a otro es una tarea que requiere que cada uno de los documentos sea convertido al nuevo formato si no se quiere tener el riesgo de que en el futuro dicho documento sea ilegible, algo en lo que SGML no tiene ningún problema.

### Características de SGML:

- ❖ Desde un punto de vista se puede ver que nos permite independencia del conjunto de representación de caracteres del equipo que se utiliza, por lo que es transparente migrar de una máquina que utilice ASCII extendido a otra que utilice UNICODE o ISO 8859-1.
- ❖ Hace énfasis en las anotaciones estructurales en lugar de las procedurales, es decir no incluye información sobre como deben ser interpretadas las anotaciones. Su principal objetivo es describir las unidades lógicas de un documento, más no como deben ser tipografiadas.
- ❖ Informalmente, siempre hemos asociado tipos a los documentos que nos encontramos: memorándums, cartas, reportes, artículos, libros, etc.; SGML formaliza esta idea al crear la noción de DTD, el cual describe una clase o familia de documentos con ciertas características comunes.
- ❖ SGML evita la dependencia de los conjuntos de caracteres, tales como ASCII y EBCDIC, específicos a fabricantes de equipo. SGML utiliza caracteres normalmente encontrados en casi todos los sistemas y define mecanismos para emplear caracteres especiales o poco comunes.

Ejemplo de código SGML:

```
<LibroDigital>
  <titulo>Tesis Manejo de Datos a través del WWW</titulo>
    <autores> Héctor Zavala Vega</autores>
    <autores> Milton A. Lima</autores>
    <autores> Juan M. Castillejos</autores>
  <disponibilidad> S </disponibilidad>
</LibroDigital>
```

### 1.3 Lenguaje de Marcado de Hipertexto (HTML)

Conforme las necesidades de tener lenguajes más robustos para crear y soportar herramientas más completas pero al mismo tiempo de uso sencillo para el uso de Internet se creó el lenguaje de programación HTML, Lenguaje de Marcado de Hipertexto, es un lenguaje simple utilizado para crear documentos de hipertexto para su publicación en Internet. No es un lenguaje de descripción de página como Postscript; HTML nos permite definir de forma estricta la apariencia de una página, aunque una utilización algo desviada hace que se utilice en ocasiones como un lenguaje de presentación. Además, la visualización de la página es muy dependiente del navegador utilizado, el mismo documento no produce el mismo resultado en la pantalla si se observa en los diferentes navegadores, Mosaic, Internet Explorer o Netscape, es decir, HTML se enfoca principalmente a describir la estructura y el contenido de un documento, y no tanto a definir el formato de la página y su apariencia.

Una de las claves del éxito de Internet, a parte de lo atractivo de su presentación es sin duda, su organización y coherencia. Todos los documentos para la publicación en Internet comparten un mismo aspecto y una única interfaz, lo que facilita enormemente su manejo por parte de cualquier persona.

Esto es posible porque el lenguaje en que están escritos los documentos, no solo permite establecer hiperenlaces entre diferentes documentos, sino que es un "lenguaje de descripción de página" independiente de la plataforma en que se utilice. Un documento HTML contiene toda la información necesaria sobre su aspecto y su interacción con el usuario, y en ocasiones dependiendo del navegador que utilizemos va a ser el responsable de asegurar que el documento tenga un aspecto coherente.

Por tanto, es un lenguaje muy sencillo que nos permite preparar documentos web insertando en el texto de los mismos una serie de marcas (comúnmente llamadas *tags*) que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos. Para escribirlo lo único que se necesita es un editor de texto ASCII, Edit del MS-DOS o el Bloc de notas de Windows, vi o pico de Unix, entre otros. Las marcas o tags que controlan el comportamiento del documento son fragmentos de texto encerrados entre los signos "mayor que" y "menor que" (<marca>).

Existen diferentes tipos de marcas: algunas controlan simplemente la presentación del texto del documento; otras la forma en que se incluirán en él imágenes y finalmente, los hiperenlaces con documentos o con diferentes partes del mismo documento.

Existen una serie de programas que nos ayudan en la elaboración de documentos HTML, como HTMLLED (shareware) o HTML Assistant, ambos para Windows, pero no son imprescindibles para escribir el código, ya que con crear un archivo ASCII con el código HTML es suficiente. Lo que si es necesario para visualizarlo es un programa cliente de Internet, tal como Mosaic, Netscape o Internet Explorer.

Las marcas funcionan muchas veces por parejas, una para indicar el inicio de enlace o formato, y otra para señalar el final. La marca de inicio consiste en una letra o una palabra. Ejemplo: <TITLE>.

El HTML estableció la estructura del Hipertexto, permitiendo flexibilidad al escribir un documento y no obligando al usuario a seguir reglas estrictas.

### **Características:**

- ❖ Asociabilidad: Los conceptos se interrelacionan gracias al HTML, permitiendo ir de uno a otro con facilidad
- ❖ Simplicidad: El HTML pretende ser lo más fácil posible, tanto en aprendizaje como en uso
- ❖ Portabilidad: Es un lenguaje independiente de plataforma

Con éstas características, HTML brinda un acceso controlado a la información, facilitando su transmisión. Pese a tener cierta discriminación, sabe lo que es un párrafo o una imagen, pero no puede saber sobre que trata el párrafo o la imagen.

HTML es un lenguaje de definición algo primitivo que abrió grandes caminos, actualmente lo que fueron sus grandes virtudes se están convirtiendo en sus mayores defectos, ya que no ha podido evolucionar a las nuevas necesidades aunque intenta emplear herramientas que le permitan seguir activo (cgi, applets y activeX) pero todo esto no parece ser suficiente para las necesidades actuales y lo que no evoluciona está condenado a desaparecer.

### **HTML y su evolución hacia XML**

Durante los últimos años hemos apreciado y vivido un cambio profundo en muchos aspectos, pero referente a la información publicada de manera electrónica, cada día existen nuevas definiciones y términos tecnológicos.

El protocolo HTML, requiere evolucionar para sobrevivir a los cambios constantes de la informática. Debido a esto, mejora hacia HTML más avanzados (HTML 4.0 actualmente) o intenta crear herramientas que le permitan sobrevivir a pesar de éstos esfuerzos, es necesario cambiar hacia hipertextos más evolucionados, los diseñadores de HTML han intentado salvar el desarrollo con la utilización de otros productos que han salido al mercado, tales como: SGML, DHTML, pero no parecen ser lo suficientemente buenos, excepto: **XML**.

## 1.4 Servidores Web

Un Servidor Web es un programa que emplea la arquitectura cliente/servidor y el HTTP para compartir datos, documentos archivos multimedia en formato web, éste despacha los archivos que forman páginas web a los usuarios que representan clientes y que remiten sus peticiones al servidor.

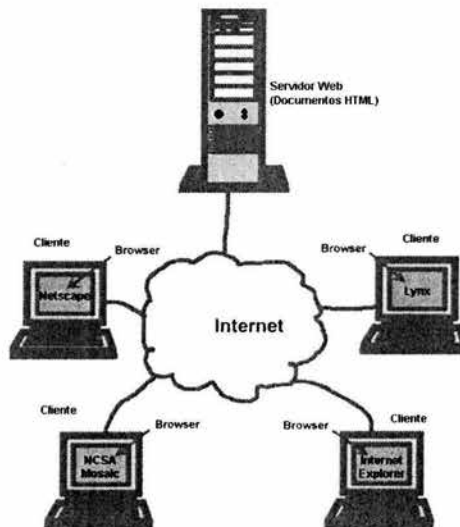
La Web se crea normalmente como un sistema abierto al que cualquiera puede contribuir y acceder desde cualquier punto de la red, la gran mayoría de los sitios web no requieren identificarse mediante un *login* o *password* como los sistemas tradicionales Cliente-Servidor.

Un servidor web envía información con la simple presión del ratón sobre de una liga u objeto dentro de una página web. La información enviada puede ser de cualquier tipo (datos, documentos, multimedia, etc.), gracias al uso de los estándares de Internet. La información enviada por un servidor web es de sólo lectura pues a diferencia de los sistemas normales Cliente/Servidor, el usuario no puede hacer cambios en el documento original. Debido a que un servidor web es de fácil acceso, hace posible publicar información de cualquier tipo de forma instantánea en cualquier lugar, esto es a través de su almacenamiento en el servidor.

Podemos ver de lo anterior que un servidor web puede ser un software, que sólo toma un archivo especificado con el comando GET, y lo envía al cliente.

La mayoría de servidores tienen algún nivel de seguridad en sus tareas. Por ejemplo, los servidores más avanzados añaden seguridad para permitir una conexión cifrada entre el servidor y el navegador para que la información de suma importancia, como números de tarjetas de crédito, pueda ser enviada por Internet, sin ser alterada.

Actualmente los dos servidores web más usados son Apache e IIS de Microsoft.





## **1.5 Sistemas administradores de bases de datos relacionales (DBMS)**

RDBMS significa Sistema de Administración de Bases de Datos Relacionales (*Relational Data Base Management System*), consiste en un conjunto de programas, procedimientos y lenguajes que proporcionan las herramientas necesarias para administrar una o más bases de datos, así como para proporcionar las interfaces necesarias al usuario o a la aplicación. En la actualidad existen diversos sistemas manejadores de bases de datos cuyas características varían dependiendo de los productos, las versiones y por supuesto las licencias. Algunos de éstos son de carácter libre como **Postgres**, y otros requieren de la compra del producto o licenciamiento como es el caso de **Oracle**.

Es importante conocer la composición del RDBMS para obtener el mejor desempeño y administración de los recursos así como de la información almacenada.

### **Administración**

Los datos pueden ser administrados de forma tal que los podemos colocar en la base de datos, y a éstos los podemos actualizar, añadir, borrar y consultar en una base de datos. Otras implicaciones de la "administración" son que, a medida que el volumen de datos y de usuarios crece, la herramienta debe ser capaz de acoplarse a este crecimiento automáticamente.

### **Base de Datos**

Es un conjunto exhaustivo, no redundante de datos estructurados, organizados independientemente de su utilización.

### Relacional

Dicho término implica que los datos contenidos en el sistema son capaces de ser relacionados con otras porciones de datos dentro del mismo sistema. Con todo esto un usuario puede ser capaz de realizar un análisis que examine las relaciones entre diferentes elementos de los datos.

Después de haber definido cada una de las partes de lo que es una RDBMS podemos definirlo como un conjunto de programas que se encargan de manejar la creación de objetos y controlar el acceso a los recursos y está compuesto por:

- ❖ **DDL** (Lenguaje de Definición de datos).
- ❖ **DML** (Lenguaje de Manipulación de datos).
- ❖ **DCL** (Lenguaje de Control de datos).
- ❖ **SQL** (Lenguaje estructurado de consultas).

## 1.6 Las Bibliotecas Digitales

Al introducir las tecnologías de información en las bibliotecas han surgido diferentes terminologías que comúnmente son utilizadas de manera indistinta y confusa. Para los fines de este trabajo es imprescindible aclarar las diferencias entre los términos: biblioteca electrónica, biblioteca virtual y biblioteca digital.

**Biblioteca electrónica:** Es aquella que cuenta con sistemas de automatización que le permiten una ágil y correcta administración de los materiales que resguarda, principalmente en papel. Así mismo, cuenta con sistemas de telecomunicaciones que le permitirán acceder a su información, en formato electrónico, de manera remota o local. Proporciona principalmente catálogos y listas de las colecciones que se encuentran físicamente dentro de un edificio.

**Biblioteca virtual:** Es aquella que hace uso de la realidad virtual para mostrar una interfaz y emular un ambiente que sitúe al usuario dentro de una biblioteca tradicional. Hace uso de la más alta tecnología multimedia y puede guiar al usuario a través de diferentes sistemas para encontrar colecciones en diferentes sitios, conectados a través de sistemas de cómputo y telecomunicaciones.

**Biblioteca digital:** Es un repositorio de acervos y contenidos digitalizados, almacenados en diferentes formatos electrónicos por lo que el original en papel, en caso de existir, pierde supremacía. Generalmente, son bibliotecas pequeñas y especializadas, con colecciones limitadas a sólo algunos temas. En este tipo de biblioteca junto con la creación de libros digitales, es en lo que centraremos nuestra aplicación para mostrar el manejo de datos a través de la WWW.

Las Bibliotecas digitales son organizaciones que proveen los recursos, incluyendo personal especializado, para seleccionar, estructurar, distribuir, controlar el acceso, conservar la integridad y asegurar la persistencia a través del tiempo de colecciones de trabajos digitales que estén fácil y económicamente disponibles para usarse por una comunidad definida o para un conjunto de comunidades.

Los documentos que integran el acervo de una biblioteca digital son textos, imágenes, video, sonido o combinaciones de éstos. Lo ideal para una Biblioteca Digital es que la recuperación de documentos sea en forma completa, realizando las búsquedas sobre el texto completo de los mismos y con posibilidad de recuperación inmediata.

Una Biblioteca Digital no es un ente aislado ya que requiere de tecnologías específicas para compartir y enlazar todos sus recursos, los enlaces entre sus recursos deben de ser transparentes para el usuario; y todo esto con un fin específico permitir el acceso universal a diversos acervos y servicios de

información que poco a poco se está convirtiendo en una meta obligada para la humanidad.

### 1.7 El documento electrónico

Un documento electrónico es un medio que permite plasmar información de valor con atributos y características específicas que le otorgan ese carácter único y oportuno.

Podemos agregar que un documento es cualquier expresión en lenguaje natural o convencional, una expresión gráfica, sonora o de imagen; en pocas palabras un documento es un mensaje (texto o diseño gráfico) sobre un determinado soporte, que está destinado a durar un periodo de tiempo.

Gracias al uso cada vez mas difundido de las nuevas tecnologías en materia de transmisión de datos de toda índole, el concepto de documento ha cambiado, ahora es cada vez mas frecuente escuchar que alguien hace referencia a un documento electrónico, los cuales emplean soportes magnéticos, como disquetes, cintas, en algunos casos CD-ROM, tarjetas inteligentes, mensajes digitalizados u otras cosas que sirven para la creación de éstos.

Los documentos electrónicos poseen los mismos elementos que un documento convencional con soporte en papel; constan de un soporte material (cintas, disquetes, circuitos, chips de memoria, redes o archivos de una computadora) que contienen un mensaje que está escrito en un lenguaje convencional de dígitos binarios o bits, que resultan ser entidades magnéticas y que los sentidos humanos no pueden interpretar directamente ya que requieren de algún medio para su interpretación; todos creados en un código o lenguaje determinado y que de igual manera que un documento convencional pueden ser atribuidos a una persona determinada en calidad de autor mediante una firma digital, clave o llave electrónica.

**Principales características de los documentos electrónicos:**

- a) Reducir el costo de reproducción de una obra.
- b) Con la tecnología actual, los usuarios pueden ejercer un mando personal en la selección de la información que se requiere, permitiendo que se creen mejores obras, con una mayor presentación.

De esto podemos ver que los documentos electrónicos brindan mejores formas de acceder, guardar, manipular, reproducir y transmitir obras completas, lo que representa una interesante propuesta para el fomento y preservación de la cultura y los conocimientos que genera la humanidad.

---

## Capítulo 2

### XML EN EL MUNDO

El lenguaje HTML, a pesar de su sencillez, es el más exitoso sistema de presentación de documentos, desde que apareció Internet, gracias a él se haya podido publicar y acceder a más información de la que jamás se ha podido imaginar.

El gran crecimiento de Internet, los intereses comerciales y la necesidad de poder realizar páginas Web vistosas, ha dado lugar a que en poco tiempo éste lenguaje haya evolucionado muy rápidamente pero por desgracia, no siempre por el camino más adecuado.

Estas razones han obligado a los miembros de W3C a, desarrollar un nuevo lenguaje (un metalenguaje) que han denominado XML, lenguaje extensible de marcas, (extensible porque no es un formato definido como HTML) que tiene las innegables ventajas del HTML pero que a su vez permite realizar muchas cosas más.

XML no es sólo un lenguaje de marcado: es un metalenguaje que permite definir un propio lenguaje de etiquetas para múltiples clases (estructuras lógicas) de documentos, esto es posible por que está escrito en SGML ISO 8879.

Los lenguajes de marcas no son equivalentes a los lenguajes de programación, aunque se definan también como lenguajes. Son sistemas complejos de descripción de datos, normalmente documentos que se ajustan a SGML, se pueden controlar desde cualquier editor ASCII. Las marcas más utilizadas suelen describirse por textos descriptivos encerrados entre signos de "<" y ">", los más común es que exista una marca de principio y otra al final, y así por

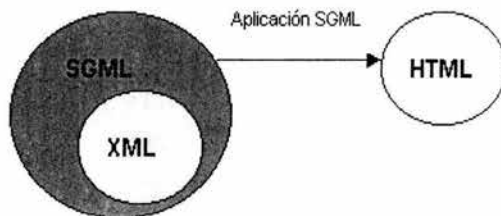
ejemplo, la siguiente marcación: **<NOMBRE> Carlos </NOMBRE>**, deja claro que "Carlos" corresponde a un nombre.

HTML es práctico, pero no indica lo que está representando, se preocupa principalmente de que algo tiene que ir en color azul, o con un tipo de letra determinada, pero no dice que lo que está representando es el título de un libro o el precio de un artículo; XML hace precisamente esto: define el contenido de la etiqueta. La potencia de esta forma de trabajar radica en que estamos etiquetando e identificando el contenido, olvidándonos en un principio por la forma de presentarlo.

SGML es la raíz de todo, es el estándar internacional para la definición de la estructura y el contenido de diferentes tipos de documentos electrónicos en múltiples campos de la actividad humana desde los historiales médicos de pacientes hasta notaciones musicales.

XML no es un tipo de documento SGML, es una versión abreviada de éste optimizada para su utilización en Internet. Esto significa que podemos definir nuestros propios tipos de documentos, nuestras propias etiquetas y ya no se depende de un único e inflexible tipo de documento como con HTML.

La diferencia entre SGML, XML y HTML se puede ver en el siguiente gráfico:



Sin embargo se pueden mencionar ventajas de usar XML en lugar de HTML, tales como:

- ❖ Autores y proveedores pueden diseñar sus propios tipos de documentos usando XML, en vez de limitarse a HTML. Los documentos pueden ser explícitamente “hechos a la medida”, autores y diseñadores son libres de inventar sus propias etiquetas.
- ❖ La información contenida es fácil de usar, porque las características de hipertexto de XML son mayores que las de HTML.
- ❖ Elimina muchas de las complejidades de SGML.
- ❖ La información es más accesible y reutilizable, por la flexibilidad de las etiquetas XML, ya pueden utilizarse sin tener que amoldarse a reglas específicas.
- ❖ Los documentos XML válidos también lo son en SGML, y pueden utilizarse fuera de la Web.

## **2.1 El nuevo estándar de Internet**

Si bien HTML, se puede considerar una aplicación de SGML, se ha convertido en el lenguaje estándar de Internet y pese a sus limitaciones que se han querido subsanar con la incrustación de scripts, JavaScripts, Active X, HTML dinámico, hojas de estilo en cascada (CSS), todo esto ha resultado insuficiente para crear una arquitectura abierta de tipo cliente/servidor.

Los creadores de XML veían en Internet un almacén de datos sin estructura que pedía a gritos ser organizado, así surgió la idea de añadir contexto y dotar de más estructura a la información de Internet.

En los años 60, se perseguía la idea de estructurar los documentos en forma organizada con el fin de facilitar el intercambio y la manipulación de los datos. IBM creó GML para las necesidades de sus sistemas internos de edición y lo utilizó



para producir libros, informes, y otros documentos a partir de un solo conjunto de archivos fuente.

La primera tecnología de información estandarizada y estructurada fue SGML; que también procedía de IBM se creó para dar formato y mantener documentos legales de la compañía. Luego comenzó a ser utilizado por un amplio número de sectores como estándar de información de propósito general. En 1986 SGML surge como estándar ISO, aunque es extremadamente potente, es muy complejo y es de precio elevado, por lo que no fue una alternativa para el hipertexto en los primeros tiempos de Internet.

En 1989, Tim Berners-Lee y Anders Berglund, dos investigadores del CERN, crearon un lenguaje basado en etiquetas para marcar documentos técnicos con el fin de compartirlos en Internet. Así surgió una aplicación simplificada del SGML llamado HTML, que era el primer formato de información estándar en la Web.

El W3C (World Wide Web Consortium), organismo que vela por el desarrollo de la World Wide Web, se replanteó crear un nuevo estándar aprovechando que SGML ofrecía las ventajas de poder y flexibilidad que le faltaban a HTML.

Así en 1996, el W3C presentó un equipo de expertos en SGML quienes tenían como objetivo crear una nueva tecnología de marcado con las ventajas de SGML y la relativa simplicidad de HTML.

De este modo surge la especificación XML 1.0 que fue ratificada por la W3C el **10 de febrero de 1998**, e interpretado por Don Connolly como "un sistema para definir, validar y compartir formatos de documentos en Internet". Se trata de un metalenguaje que ordena, estructura y describe los documentos de las páginas para Internet, permitiendo una descripción más minuciosa.

Algunos objetivos planteados por el Grupo de Trabajo XML y el W3C fueron:

- ❖ XML debía ser directamente utilizable sobre Internet.
- ❖ Debía soportar una amplia variedad de aplicaciones.
- ❖ Debía ser compatible con SGML.
- ❖ Debía ser fácil la escritura de programas que procesen documentos XML.
- ❖ El número de características opcionales en XML debían ser absolutamente mínimo, idealmente cero.
- ❖ Los documentos XML debían ser legibles por los usuarios de este lenguaje y razonablemente claros.
- ❖ El diseño de XML debía ser formal, conciso y preparado rápidamente.
- ❖ Los documentos XML debían ser fáciles de crear.
- ❖ La brevedad en las marcas XML es de mínima importancia.

A estos fines se le une unos estándares como el UNICODE e ISO/IEC 10646 para caracteres, el Internet RCF 1766 para las marcas de identificación de lenguajes, ISO 639 para los códigos de nombres de lenguajes y también el ISO 3166 para códigos de nombres de países, provee toda la información necesaria para entender XML versión 1.0 y construir programas de computadora que lo procesen.

### **2.2.1 Características Generales**

XML permite estructurar la información en un árbol. Por ejemplo hay que imaginar una receta como un componente, que a su vez está formado de componentes, y así sucesivamente. Cada componente podría tener texto y/o más componentes. Una estructura posible para este ejemplo sería pensar que la receta tiene un componente llamado "*necesitamos*". No todo el texto estaría dentro de "*necesitamos*", solamente aquellas cosas se necesitan en la receta para llevarla a cabo. Dentro podríamos tener uno o más componentes llamados *ingredientes*. Esto se ve de la siguiente manera utilizando la sintaxis de XML.

Ejemplo:

**<ficha>**

**<referencia>**

**<nota>El autor citado trabajó en el Banco Nacional</nota>**

**<nota>El gobierno ha solicitado la omisión de esa información</nota>**

**</referencia>**

**</ficha>**

Aquí podemos ver como simplemente con encerrar el texto que pertenece a un componente entre **<etiqueta>** y **</etiqueta>**. En realidad a los componentes se les suele llamar *tag*.

La presentación se realiza mediante una hoja de estilo, que es una descripción de la manera en que debe verse una información en determinado medio.

A un mismo documento XML se le pueden aplicar distintas hojas de estilo según convenga. Por ejemplo usando una hoja de estilo por cada medio en la que se debe presentar la información. Una para imprimir un libro, otra para una página de Internet y otra para un programa que lee la información.

Las primeras dos dirán algo así:

- ❖ Mostrar la etiqueta **<ficha></ficha>** como un recuadro blanco con borde rojo. Centrado.
- ❖ Por cada componente dentro de la etiqueta **<ficha>** mostrar el contenido en azul con una tipografía negrita.

En cambio, una hoja de estilo destinada a formatear la información para su lectura tendrá algo así como:

- ❖ Leer el contenido de **<ficha>** con una suave música de fondo.
- ❖ Recitar cada *nota* lentamente.

Las hojas de estilo también pueden controlar la conversión de esta "información pura" en otros formatos como RTF (de Word), HTML, etc. De esta manera, cuando ya tenga las 750 recetas capturadas y puestas en Internet mediante una hoja de estilo adecuada, se podrá publicar un libro con tal formato.

XML provee una forma de definir un vocabulario y las interfaces entre dos o mas partes que vayan evolucionando con el tiempo. Por ejemplo un sitio de compras por Internet dialogando con una empresa de tarjetas de crédito.

Una clase de objetos de datos llamados documentos XML describe parcialmente el comportamiento de programas que pueden procesarlos.

Los documentos de este tipo están hechos de unidades de almacenamiento llamadas entidades, las cuales contienen datos que pueden ser o no procesadas. Los datos procesados están hechos de caracteres, algunos de los cuales forman datos de texto y otros marcas. Las marcas codifican la descripción del esquema de almacenamiento y estructura lógica del documento. Además provee un mecanismo para imponer restricciones al esquema de almacenamiento y estructura lógica.

Un módulo de software llamado *Procesador de XML* es usado para leer documentos XML y proveer acceso a su contenido y estructura.

Se presupone que el procesador está haciendo su trabajo en beneficio de otro módulo, llamado la **aplicación**. Esta especificación describe el comportamiento requerido por el procesador en términos de cómo debe leer datos y la información que debe proveer a la aplicación.

## 2.2 Documentos XML

Las herramientas de creación XML se utilizan para crear y editar documentos XML, la más simple de todas es el editor de textos aunque es recomendable utilizar un editor específico. Dentro de estos, existen dos tipos principales:

- ❖ Los que representan los archivos en forma de árbol y nos permiten construir nuestro documento trabajando sobre este árbol y formularios adicionales.

Ejemplos: SoftQuadXMetal , XML Notepad de Microsoft , EditML, Visual XML, XMLwriter

- ❖ Los que representan el documento XML en su formato original y que normalmente son editores de archivos de texto con facilidades para XML.

Ejemplos: XED , PSGML de Emacs

Entre ambos tipos hay que diferenciar los que trabajan con una DTD y por lo tanto validan el contenido de lo que escribimos y los que simplemente nos aseguran que el documento XML esté bien formado, es decir, sintácticamente correcto respecto de las especificaciones de éste.

Los 4 requisitos para que un documento sea considerado válido son:

- ❖ El documento debe estar bien construido.
- ❖ El elemento root (raíz) debe coincidir con el nombre que proporcione la declaración de tipo de documento (DTD).
- ❖ El documento debe tener una DTD que declara todos los elementos, atributos y entidades que se utilicen en el documento.
- ❖ El documento debe adherirse a la gramática que establezca la DTD.

Los requisitos principales para que un documento se considere bien construido son:

- ❖ Debe tener un elemento de documento en el que se encuentren todos los demás elementos, o sea un único elemento raíz: Ejemplo:

```
<ObrasDigitales>
  <LibroDigital> ... </LibroDigital>
  <LibroDigital> ... </LibroDigital>
  <LibroDigital> ... </LibroDigital>
</ObrasDigitales>
```

- ❖ Todo elemento deberá tener una etiqueta de inicio y su correspondiente etiqueta de cierre, o una sola etiqueta vacía.

Ejemplo:

```
<CIUDAD>Atenas</CIUDAD>
```

En caso de trabajar con etiquetas "vacías", hay que incluir la "barra" (/) antes del signo "mayor" (>)

```
<ALUMNO NUMERO="1"/>,
```

```
<BR/>
```

```
<BR></BR>
```

- ❖ Todos los elementos deberán estar correctamente anidados.

Ejemplo:

```
<ROJO> Color rojo <CLARO> claro </CLARO> normal </ROJO>
```

Todos los valores de atributos deberán ir entre comillas.

Ejemplo:

```
<LIBRO ISBN="84-415-0845-3" />
```

- ❖ Todas las entidades que se utilicen deberán declararse en la DTD.
- ❖ XML distingue entre mayúsculas y minúsculas. Puede utilizar los dos tipos de letra, pero no las considera iguales.

Ejemplo:

```
<AMIGOS> <amigo> Juan Manuel </amigo> </AMIGOS>
```

### 2.2.1 Estructura de los Documentos XML

En un documento XML se distinguen 2 estructuras: **física y lógica**.

La estructura física está dada por una o más unidades de almacenamiento virtual, llamadas entidades. Todas estas unidades tienen contenido y todas ellas (excepto la entidad documento y el subconjunto externo del DTD) están identificadas por un nombre. Cada documento XML contiene una entidad, llamada raíz del documento, que sirve como punto de partida para el procesador XML y puede contener el documento completo.

Como se mencionó anteriormente, las entidades pueden ser analizadas o no (también llamadas procesadas o sin procesar). El contenido analizado se conoce como texto de reemplazo, y es parte integrante del documento. Las entidades que no son analizadas forman parte de los recursos (tales como enlaces) cuyo contenido puede o no ser texto. Cada entidad no asociada tiene una notación asociada, identificada por un nombre. La forma de invocar ambos tipos es a través de su nombre, en el caso de las analizadas a través de su referencia y en el de las no analizadas a través de sus atributos.

Las que son generales, son analizadas y se usan en el interior del documento. Las parametrizadas son las que se usan en el ámbito del DTD. Estos dos tipos usan distintos tipos de referencias y se reconocen en contextos distintos.

La estructura lógica esta determinada por los elementos que hay en ese documento junto con las relaciones que hay entre tales elementos, los cuales están delimitados por etiquetas de comienzo y de final o, en el caso de elementos vacíos, por una etiqueta de elemento vacío.

Cada elemento tiene un tipo identificado por un nombre, denominado identificador genérico que puede tener un conjunto de especificaciones de atributos.

Cada especificación de atributo tiene un nombre y un valor. Estas especificaciones no restringen la semántica, el uso o los nombres de los tipos de los elementos y atributos. Una analogía de la estructura física-lógica es una base de datos relacional, que emplea tablas y campos para la estructura lógica y en general archivos binarios para la estructura física.

### **Componentes de Mercado XML**

- ❖ Etiquetas de elementos.
- ❖ Instrucciones de procesamiento.
- ❖ Declaraciones de tipos de documentos.
- ❖ Referencias de entidades.
- ❖ Comentarios.
- ❖ Secciones marcadas.

Las etiquetas, referencias de entidad y declaraciones se denominan "marcas". Éstas son las partes del documento que un procesador XML debe entender. El resto del documento que se encuentra entre las marcas, son los datos que resultan entendibles por las personas.



Es sencillo reconocer las marcas en un documento XML. Son aquellas partes que empiezan con "<" y acaban con ">", o bien, en el caso de las referencias de entidad, empiezan por "&" y acaban con ";".

### Instrucciones de procesamiento

Son instrucciones especiales que son usadas por la aplicación que está procesando un documento XML. Siempre comienza con los signos <? y terminan con los signos ?>.

La primera instrucción, o "declaración XML", define la versión de XML usada. Además, en la "declaración XML" especificamos la codificación del documento, que puede ser:

US-ASCII (7 bits) o UTF-8 (código Unicode del que el ASCII es un subconjunto), ISO-10646-UCS-2 (Unicode), ISO-10646-UCS-4 (Unicode), de ISO-8859-1 hasta ISO-859-9, ISO-2022-JP (JIS X-0208-1997), EUC-JP (JIS X-0208-1997), Shift\_JIS (JIS-X-0208-1997). En general, y para uso con lenguajes europeos (incluyendo el juego de caracteres especiales del castellano, usamos UTF-7 o ISO-8859-1)

Se puede describir el estado autónomo de un documento que determina si se apoya o no en una fuente de información externa, como una DTD externa. Los valores posibles de standalone son "yes" y "no". Yes indica que el documento es autónomo y el No indica que el documento depende de una DTD, por eso es el valor predeterminado de standalone, lo que implica que se puede simplificar la declaración de documento autónomo si se piensa usar una DTD externa.

Ejemplo:

```
<?xml version="1.0"? encoding="UTF-7"standalone="no">
```

La segunda instrucción, o "declaración de tipo de documento", define el tipo de documento que estamos creando para ser procesado correctamente.

Ejemplo:

```
<!DOCTYPE librodecocina SYSTEM "librodecocina.dtd">
```

### **Etiquetas de elementos**

Son los componentes que definen la descripción de los elementos.

Un nombre empieza con una letra o signos de puntuación, y continúa con letras, dígitos, guiones, rayas, dos puntos o puntos, denominados de forma global como caracteres de nombre. Los nombres que empiezan con la cadena "xml", se reservan para la estandarización de esta o de futuras versiones del lenguaje.

No se pueden crear nombres que empiecen con la cadena "xml", "xMI", "XML" o cualquier otra combinación. Las letras y rayas se pueden usar en cualquier parte del nombre. También se pueden incluir dígitos, guiones y caracteres de puntuación, pero no se puede empezar por ninguno de ellos. El resto de caracteres, como algunos caracteres especiales y espacios en blanco, no se pueden usar.

Los elementos XML pueden tener contenido más elementos, caracteres, o ambos a la vez, o bien ser elementos vacíos.

Ejemplo:<titulo> **Manejo de datos a través de WWW** </titulo>

Siempre empieza con una <etiqueta> que puede contener atributos o no, y termina con una </etiqueta> que debe tener el mismo nombre. En XML siempre se debe "cerrar" un elemento.

Un elemento vacío, es aquel que carece de contenido. Al no tener una etiqueta de "cierre" que delimite un contenido, se utiliza la forma <etiqueta/>, que puede contener atributos o no.

Ejemplo:

```
<alumno numero="2"/>
```

### Referencias de entidades

Como las entidades son los bloques de construcción de los documentos XML, suelen estar formadas por otras entidades a través de referencias de ésta, que se utilizan para asignar alias a piezas de datos, sirven como nombre único para una pieza de datos XML.

Las referencias a entidades están compuestas por un "&" y ";", éstas se declaran automáticamente, sin que el procesador XML lo interprete como marcado:

&amp;	corresponde con ampersand (&)
&apos;	corresponde con el apóstrofe (')
&quot;	corresponde con las comillas ("")
&lt;	corresponde con menor que (<)
&gt;	corresponde con mayor que (>)

De esta manera se puede utilizar el carácter "<" evitando que se interprete como el comienzo de una etiqueta XML

### **Comentarios**

Los comentarios se ocupan para presentar información que técnicamente no forma parte del contenido de ese documento. Describen datos para provecho del usuario. Los analizadores u aplicaciones XML suelen ignorar los comentarios.

Se pueden utilizar en cualquier parte del documento en la que aparezcan datos de caracteres analizados sintácticamente.

Ejemplo:

```
<!--Este es un comentario en XML -->
```

### **Definición del Tipo de Documento**

Se emplea para especificar información acerca de un documento. Es importante cuando se establece si un documento es válido o si sólo está bien construido.

Las tres tareas que lleva a cabo una declaración tipo de documento son:

- ❖ Especificar el elemento raíz del documento
- ❖ Definir elementos, atributos y entidades específicas del documento (DTD internas)
- ❖ Identificar la DTD externa en el documento

La mayoría de las DTD se diseñan utilizando el enfoque de arriba-abajo (top-down), empezando por el elemento raíz (root). Normalmente se debe incluir una declaración XML al principio de este tipo de documentos, seguida de una declaración de tipo de documento. Es necesario hacer referencia a la DTD externa, que es el URI de un archivo que la contiene, en este caso `librodecocina.dtd`.

Ejemplo:

```
<?xml version="1.0"?>
```

```
<!DOCTYPE librodecocina SYSTEM "BibliotecaDigital.dtd">
```

El elemento raíz del documento es el elemento BibliotecaDigital. La DTD externa del documento es BibliotecaDigital.dtd .

### Secciones marcadas

Las secciones de datos de caracteres no analizados sintácticamente o secciones CDATA se emplean para bloquear texto que tiene que ser separados por un analizador XML.

### 2.2.2 Los esquemas y el documento de tipo de dato (DTD)

Un documento XML es un medio estructurado para almacenar información, se deberá establecer con exactitud a qué estructura debe adherirse ésta. Esto se logra a través de un esquema.

Un esquema describe el modelo de datos de un documento XML. Se pueden definir dos tipos de esquemas: las DTD y los esquemas de XML.

Se puede evaluar la importancia de cada solución disponible a la hora de decidir cual utilizar y porque. A continuación se listan las principales diferencias que hay entre las DTD y los esquemas XML:

- ❖ Las DTD están basadas en una sintaxis especializada, mientras que los esquemas están basados en XML por ello las primeras se pueden analizar sintácticamente y manipularse como cualquier otro documento XML.
- ❖ Las DTD son compactas, mientras que los esquemas no lo son.

- ❖ Existen mas herramientas disponibles para procesar documentos XML con DTD que para procesar con los esquemas XML.
- ❖ Los esquemas XML soportan una serie de tipos de datos: entero, flotante, booleano, fecha, etc. mientras que las DTD tratan todos los datos como cadenas o cadenas enumeradas.
- ❖ Los esquemas XML presentan un modelo de datos abierto, lo que permite ampliar los vocabularios y establecer relaciones de herencia entre los elementos sin invalidar a los documentos. Las DTD emplean un modelo de datos cerrado que no permite prácticamente su ampliación.
- ❖ Los esquemas XML soportan la integración de los espacios de nombres. Las DTD sólo permiten una asociación entre un documento y su DTD a través de la declaración de tipos de documento.
- ❖ Los esquemas XML soportan los grupos de atributos, las DTD soportan una forma primitiva de agrupación.
- ❖ Se puede pensar que los esquemas XML representan con mucha probabilidad el futuro del modelado de datos XML, las DTD siguen siendo vigentes ya que han existido desde hace mucho tiempo y son comúnmente aceptadas. Otro punto es que las DTD describen muy bien la estructura de los documentos XML. Lo importante es evaluar las necesidades y las herramientas que se van a utilizar antes de decidir entre las DTD y los esquemas XML.

Los esquemas sirven para permitir la validación de máquina de una estructura de documento. Se utilizan para modelar clases de datos, lo que también se conoce con el nombre de vocabularios XML. Un esquema describe un vocabulario XML específico definiendo claramente las relaciones entre los elementos del mismo. Los esquemas lo hacen de dos formas:

- ❖ Establecen el modelo de contenido de los documentos, lo que define el orden y anidación de los elementos.
- ❖ Establecen los tipos de datos del documento (no soportadas en las DTD).

### Composición del documento de tipo de dato DTD

Las DTD tienen su origen en SGML y sirven como mecanismo estándar de esquema para validar documentos SGML, como XML es un subconjunto de SGML, tiene sentido utilizar la misma solución de esquema.

En las DTD se pueden incluir los siguientes tipos de declaraciones de marcado:

- ❖ Tipos de elementos.
- ❖ Atributos que se pueden asignar a cada elemento.
- ❖ Entidades que se permiten en el documento.
- ❖ Notaciones que se permiten para usarlas con las entidades externas.

Se puede describir el tipo de dato de un documento utilizando DTD internas y externas, o una combinación de ambas.

La primera debe usarse como en la declaración de tipo de documento, mientras que si se hace referencia a la DTD externa, la sintaxis sería la siguiente:

```
<!DOCTYPE RootElemen SYSTEM ExternalDTDRef [internalDTDDecl]>
```

#### Ejemplo:

```
<?xml version="1.0"?>
```

```
<!DOCTYPE librodecocina SYSTEM "librodecocina.dtd">
```

```
<BibliotecaDigital>
```

```
<!--El contenido del documento va aquí-->
```

```
</ BibliotecaDigital >
```

**Ejemplo de una DTD interna:**

```
<?xml version="1.0"?>

<!DOCTYPE BibliotecaDigital[

    <!--el contenido del esquema va aquí-->

< BibliotecaDigital >

    <!--el contenido del documento va aquí-->

</ BibliotecaDigital >
```

**Componentes principales**

Son los elementos y los atributos, los primeros establecen la estructura lógica del documento como unidad lógica de información y los segundos son características de esa información.

Las declaraciones de tipo elemento deben empezar con "<!ELEMENT" seguidas por el identificador genérico de éste.

XML soporta 4 tipos de elementos:

<p><b>Vacío</b></p>	<p><b>El elemento no tiene contenido pero puede contener atributos:</b></p> <p>Declaración: &lt;!ELEMENT ElementName EMPTY&gt;</p> <p>Ejemplo: &lt;!ELEMENT imagen EMPTY&gt;</p> <p>&lt;imagen scr="clio.gif"&gt;&lt;/imagen&gt; ó</p> <p>&lt;imagen scr="clio.gif"/&gt;</p>
<p><b>Sólo elementos</b></p>	<p><b>Contiene elementos secundarios:</b></p> <p>Declaración: &lt;!ELEMENT ElementName ContentModel&gt;</p>



	<p>Ejemplo: &lt;!ELEMENT resumen(intro(educación,experiencia+)+,hobbie?,referencia*)&gt;</p> <p>El elemento resume contiene elementos secundarios</p> <p>El elemento intro contiene a educación y experiencia que deben aparecer al menos una vez (+ fuera del paréntesis), además educación una vez con experiencia más de una vez(+ dentro del paréntesis) El elemento hobbie puede aparecer exactamente una o ninguna vez</p> <p>El elemento referencia puede aparecer varias veces</p>
<b>Mixto</b>	<p><b>Contiene datos de caracteres y elementos secundarios:</b></p> <p>Declaración: &lt;!ELEMENT ElementName (#PCDATA)&gt;</p> <p>&lt;!ELEMENT ElementName (#PCDATA ElementList)*&gt;</p> <p>Ejemplo: &lt;!ELEMENT banda (#PCDATA)</p> <p>&lt;banda&gt;Los Platers&lt;/banda&gt;</p> <p>&lt;!ELEMENT memorando(#PCDATA  titulo  sujeto)*&gt;</p> <p>contiene datos de caracteres y la lista de opciones contiene elementos secundarios y se parece a un modelo de contenido de sólo elementos</p>
<b>Cualquier</b>	<p>Prácticamente no tiene estructura, puede contener datos de caracteres, tipos de elementos declarados, o una mezcla de ambos.</p> <p>Declaración: &lt;!ELEMENT ElementName ANY&gt;</p>

Los atributos se declaran por medio de una lista de atributos:

**<!ATTLIST ElementName AttrName AttrType Default>**

**Default:** tiene 4 valores predeterminados

- ❖ **#REQUIRED:** atributo obligatorio.
- ❖ **#IMPLIED:** atributo opcional.
- ❖ **#FIXED:** atributo con un valor fijo.
- ❖ **default:** valor predeterminado del atributo.

XML soporta 10 tipos de atributos:

CDATA	Datos de caracteres no analizados sintácticamente, una cadena de caracteres sencilla
Enumerados	Una serie de valores de cadena
NOTATION	Una notación declarada en alguna parte de la DTD
ENTITY	Una entidad binaria externa
ENTITIES	Múltiples entidades binarias externas separadas por espacios en blanco
ID	Un identificador único
IDREF	Una referencia a un ID declarado en alguna otra parte de la DTD
IDREFS	Múltiples referencias a los ID declarados en alguna otra parte del documento
NMTOKEN	Un nombre compuesto por caracteres con símbolo XML (letras, números, puntos, guiones, dos puntos y subraya)
NMTOKENS	Múltiples nombres compuestos por caracteres con símbolo XML

**Los tipos de atributos se clasifican en 3 grupos:**

De cadena: son los que más se utilizan y se representan por el tipo CDATA. Ejemplos de cómo declarar un atributo CDATA que debe ser definido en el elemento jugador, el atributo equipo puede ser obligatorio u opcional según sea el caso:

```
<!ATTLIST jugador equipo CDATA #REQUIRED>
```

```
<!ATTLIST jugador equipo CDATA #IMPLIED>
```

Enumerados: se componen de atributos cuyos valores están limitados a una lista de cadenas de texto predefinidas. Ejemplo:

```
<!ATTLIST jugador posición(central | defensor | delantero )"central">
```

Con símbolo: son atributos que son procesados como símbolos por un analizador XML los cuales son: ENTITY, ENTITIES, ID, IDREF, IDREFS, NMTOKEN, NMTOKENS.

Ejemplo: <!ATTLIST logo imagen ENTITY #IMPLIED>

El tipo ENTITY hace referencia a una imagen como entidad binaria, asocia una imagen con un tipo de elemento.

### Composición del Esquema

Es una aplicación XML, lo que significa que se implementa como vocabulario, define qué elementos puede contener un documento de este tipo, cómo están organizados, qué atributos y de qué tipo pueden ser sus elementos. Crear esquemas es similar a crear cualquier otro documento de estilo, con un vocabulario especializado.

En realidad es sólo una implementación parcial de la nota del W3C sobre como deben funcionar los esquemas de documentos. Por el momento Internet Explorer 5.0 y sus versiones posteriores son las que soportan esquemas. Un esquema descrito con "XML Schema" es similar a una DTD, no obstante en esta última, el contenido de los elementos se limita a cadenas y a otros tipos de datos primitivos.

**Los esquemas declaran elementos que se usan para describir a los documentos:**

ELEMENTO	DESCRIPCIÓN
Schema	Sirve como elemento raíz de los documentos XML Schema
Datatype	Describe los tipos de datos en los elementos y los atributos
ElementType	Describe el tipo de elemento
element	Puede consistir en abrir y cerrar un tag o etiqueta y siempre empieza con "<" y termina con ">". Por ejemplo : <element>Text</element>.
group	Organiza los elementos en grupos, con fines organizativos
AttributeType	Describe el tipo de atributo
attribute	Identifica un atributo que puede producirse en un tipo de elemento
description	Proporciona documentación para un elemento o un atributo

Los elementos incluyen atributos:

ELEMENTO	DESCRIPCIÓN
Schema	<p>name: nombre del esquema</p> <p>xmlns: espacio de nombres del esquema</p>
datatype	<p>dt:type: especifica el tipo de elemento o atributo (entity, entities, enumeration, boolean, char, date, etc.)</p>
ElementType	<p>name: el nombre del elemento</p> <p>model: si el modelo de contenido está abierto o cerrado</p> <p>content: el tipo de contenido que hay en el elemento (empty, textOnly, eltOnly, mixed)</p> <p>order: el orden de los elementos secundarios y grupos que hay en el elemento (one, seq, many).</p> <p>dt:type: el tipo de elemento</p>
element	<p>type: el tipo de elemento</p> <p>minOccurs: el mínimo número de veces que el elemento debe producirse</p> <p>maxOccurs: el máximo número de veces que el elemento debe producirse</p>
group	<p>order: el orden de los elementos secundarios que hay en el grupo (one, seq, many)</p> <p>minOccurs: el mínimo número de veces que el elemento debe producirse</p> <p>maxOccurs: el máximo número de veces que el elemento debe producirse</p>
AttributeType	<p>name: el nombre del tipo atributo</p> <p>dt:type: el tipo de datos del atributo</p> <p>dt:values: la lista de valores posibles de un atributo enumerado, sólo se aplica cuando dt:type está establecido a enumeration</p> <p>default: el valor predeterminado de un atributo</p> <p>required: indicador que muestra si debe proporcionarse el atributo en el elemento</p>
attribute	<p>type: el tipo de atributo</p>

ELEMENTO	DESCRIPCIÓN
	default: el valor predeterminado del atributo
	required: indicador que muestra que debe darse el atributo en el elemento
description	Es un elemento de sólo texto diseñado con fines de documentación

El esquema actúa como contenedor del resto del contenido de los esquemas. El atributo *xmlns* es considerado una instrucción de procesamiento ya que establece el espacio de nombres del esquema.

Ejemplo:      <Schema name =“libroschema”  
  
                  xmlns=“urn:schemas-microsoft-com:xml-data”  
  
                  xmlns:dt=“urn:schema-microsoft-com:datatypes”>  
  
                  <!--el contenido del esquema va aquí -- >  
  
                  </Schema

### Hojas de Estilo

#### ¿Qué es CSS (Cascade Style Sheet)?

CSS es un lenguaje sencillo para la aplicación de estilos a un elemento XML. Podemos tener un documento XML y diferentes posibles formatos dependiendo del uso que se le vaya a dar al mismo.

Las hojas de estilo en cascada, fueron introducidas en 1996 como estándar para añadir información de estilo a los documentos HTML. Pero este uso estaba restringido sólo a las etiquetas propias de HTML. Con XML estas reglas de estilo pueden aplicarse a todos los elementos.

CSS se expresa mediante reglas en un archivo de texto plano. Cada regla contiene el nombre del elemento al que se aplica y el estilo definido.

### CSS para XML

Para poder visualizar un documento XML es necesario especificar qué formato se debe utilizar. Para este propósito se utiliza la instrucción `xml-stylesheet`

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet href="miArchivo.css" type="text/css"?>
```

De este modo el archivo de estilos se incluye en el documento XML. La instrucción posee dos atributos.

**href:** indica la dirección absoluta o relativa donde se encuentra el archivo de estilos. En este caso el valor de href podría haber sido:

```
http://teseo.dcaa.unam.mx:8080/LibroDigital/miArchivo.css
```

En el ejemplo anterior el archivo se denomina `miArchivo.css` y debe encontrarse en el mismo directorio en el que estamos grabando.

**type:** Representa el tipo MIME que se va a utilizar que será `text/css` para un archivo CSS.

### Unidades CSS

La lista consta de un conjunto de pares de elementos separados por el carácter ";" formados por el nombre de una propiedad y un valor y separados por el carácter ":".

Sintaxis: `elector {propiedad:valor; propiedad:valor; ... propiedad:valor}`

#### **Selectores**

Una regla de estilo consta de dos partes: El selector y la cadena de estilos. El selector más utilizado es simplemente el nombre de la etiqueta a la que se refiere la regla. La etiqueta es la definida en el archivo XML.

```
<?xml version="1.0"?>
<?xml-stylesheet href="LibroDigital.css" type="text/css"?>
<!DOCTYPE LibroDigital SYSTEM "LibroDigital.dtd">
<LibroDigital>
<obra>
```

```
<titulo>Historia de México</titulo>
<capitulo>
<cap_titulo>México prehispánico</cap_titulo>
</capitulo>
<capitulo>
<cap_titulo>La Nueva España</cap_titulo>
</capitulo>
</obra>
```

### **XSL**

Las hojas de estilos XSL se usan para transformar el contenido de los documentos y darle formato para mostrarlo en navegadores web. Aunque la especificación completa de XSL sigue estando "en construcción", el componente XSL Transformation (XSLT) está alcanzando el estatus de recomendación del W3C.

#### **XSL e Internet Explorer**

Antes de empezar a construir hojas de estilos XSL, es importante saber exactamente qué navegadores las soportan, y a qué nivel. En el momento de realizar esta tesis, Internet Explorer Ver 5.0 y 6.0.2600 es el único navegador comercial que soporta XSL. Este soporte consta de una implementación parcial del componente XSLT de la tecnología XSL, que se usa para transformar documentos de un vocabulario XML a otro. Merece la pena mencionar que Microsoft no sigue exactamente el último borrador de trabajo de XSLT en la forma establecida por el W3C. De hecho, existen ciertas extensiones originales Microsoft de XSLT que sólo la soporta Internet Explorer 5.0

#### **Dentro de las hojas de estilos XSL**

Aunque las hojas de estilos XSL son muy potentes y pueden llegar a ser algo complicadas al aplicarse a documentos XML complejos, su estructura básica es muy simple. Una hoja de estilos XSL consta de una o más plantillas que describen los patrones, que se usan para cotejar el contenido XML en relación a los estilos.

Las dos construcciones fundamentales de una hoja de estilos XSL son: Plantillas y Patrones.

### **Plantillas**

Una plantilla es una estructura XSL que describe la salida a generar en base a ciertos criterios de coincidencia. La idea consiste en definir una estructura de transformación que se aplique a una cierta parte de un documento XML.

Las plantillas se definen en las hojas de estilos XSL utilizando el elemento **xsl:template**, que es, principalmente, un elemento contenedor de los patrones y datos de transformación. El elemento **xsl:template** utiliza un atributo opcional llamado **match** para hacer coincidir los patrones en un documento XML. El atributo **match** especifica una porción del árbol XML de un documento. La coincidencia más amplia posible de un documento consiste en establecer el atributo **match** a **"/**", lo que indica que la raíz del árbol tiene que coincidir. El resultado es que en la plantilla selecciona la totalidad del árbol:

```
<xsl:template match="/" />
</xsl:template>
```

El atributo **match** de una plantilla es algo así como una consulta de un lenguaje de base de datos, los datos del documento XML que coinciden con el patrón del atributo **match** se pasan a la plantilla para su procesamiento por ejemplo, lo siguiente sólo coincide con los elementos llamados **actor**:

```
<xsl:template match="actor">
  <xsl:value-of/>
</xsl:template>
```

En cierto modo, el mecanismo de cotejo de XSL se parece a los selectores de las CSS, pero el cotejo de patrones XSL va mucho más allá. De hecho, se trata en realidad de un lenguaje de consulta muy rico para XML. En el ejemplo anterior, es posible que se haya fijado en el elemento **xsl:value-of**. Este elemento se utiliza para insertar el contenido de un elemento o de un atributo para su salida.



En el ejemplo, el contenido del elemento **actor** se ofrece como parte de la transformación del documento. Esto tendrá mucho más sentido un poco más adelante en este capítulo, cuando veamos el elemento `xsl:value-of` utilizado en el contexto de una hoja de estilos completa.

### Patrones

Los patrones se usan en las plantillas XSL para llevar a cabo cotejos y, a fin de cuentas, son los responsables de determinar qué porciones de un documento XML se pasan a través de una determinada plantilla para su transformación. Un patrón describe una rama de un árbol XML, que a su vez consiste en una serie de nodos jerárquicos. La sintaxis que utilizan los patrones XSL es parecida en parte a la que se usa para especificar las rutas de una unidad de disco. Por ejemplo, el patrón `películas/película/título` selecciona los elementos `título` que aparecen bajo un elemento `películas`, que aparece bajo un elemento `películas`.

Para seleccionar un árbol de documento completo, se usa el patrón raíz, que consta de una sola barra (`/`). El patrón raíz se presupone que está en otros patrones si no se sustituye. Por ejemplo, se presupone que `películas/película/título` empieza en la raíz del documento, lo que significa que se hace referencia a `películas` para referirse a la raíz. Los patrones se utilizan en XSL para describir porciones de un árbol de documento con el fin de aplicar plantillas. Aparte del simple uso de los patrones que hemos visto en esta sección, los patrones XSL también pueden llevar a cabo tareas de cotejo de patrones más complejas. De hecho, los patrones forman un lenguaje de mini-consulta que se puede usar para proporcionar control a las porciones de un documento XML que se seleccionan para su transformación en plantillas.

## **2.3 Negocios en la WEB**

Con las posibilidades que ofrece XML de manejar e intercambiar información personalizada, Internet irá evolucionando hacia una red de negocios, en la cual el comercio electrónico superará en importancia a los sitios de ocio y entretenimiento que predominan hoy en día.

Con XML ha nacido una nueva generación de software de servidor web que permite al desarrollador añadir un nuevo servidor de nivel intermedio, y así poder extraer de una sola vez de la base de datos remota, la información que se ajusta al perfil del cliente y plasmando ésta en documentos XML, finalizando después de esto la conexión a servidores remotos.

Esta información enviada al nivel intermedio es homogénea, personalizada y precisa, conteniendo los datos abstractos reales y no la apariencia de la pantalla, minimizando el tráfico web, proporcionando un mayor grado de satisfacción e interacción del usuario.

### **2.3.1 Intercambio Electrónico de Datos (EDI)**

Sirve para las transacciones entre compañías y proveedores, antes era sólo un proyecto debido a las dificultades y costos que se debían implementar, pero XML ha sido una solución global al problema EDI, es capaz de aportar el formato de mensajes estructurado y flexible que combinados con Internet y sus servicios de conectividad de bases de datos, pasó EDI de ser una serie práctica de interconexiones únicas a ser un proveedor Web (fabricante genérico de inteligencia para hacer negocios en Internet)

### **2.3.2 Comercio Electrónico**

El cliente introduce determinados atributos sobre el artículo deseado y se iniciará automáticamente la búsqueda a través de miles de productos de numerosos comerciantes, apareciendo los resultados en una sola tabla, en

formato XML que permite la entrega de datos estructurados y en documentos uniformes para todos los sitios, sea cual fuere el aspecto original de los datos.

### **2.3.3 Traducción**

Uno de los mayores desafíos hoy día es simplificar el proceso de traducción, controlando manualmente al contenido (lo nuevo, reutilizado, repasado, traducido, revisado y aprobado). Combinando XML y los contenidos, las compañías logran automatizar muchas funciones para simplificar ciertos procesos, reducir los plazos de lanzamiento, evitar en lo posible repetir tareas ya realizadas y controlar los costos.

### **2.3.4 Ayuda de Escritorio (soporte técnico)**

A medida que va aumentando la complejidad de los productos, la ayuda de escritorio va adquiriendo un papel cada vez mayor por cumplir con las expectativas del consumidor.

Los ingenieros de soporte crean un documento de solución XML personal con información relativa a una llamada de consulta al sitio Web de la compañía etc., mediante los datos del cliente e información contenida en el almacén.

La nueva solución es corregida en el almacén y se generan metadatos (datos referente a datos) automatizados procedentes del sistema, respondiendo en medida exacta de sus necesidades al usuario que efectúa la consulta.

## **2.4 Perspectivas para XML**

Debido a los beneficios y por el propio impulso que genera el Internet, las diferentes organizaciones de estandarización comercial se han puesto de acuerdo en la adopción del XML como la nueva base de comunicación para el desarrollo de transacciones en el mundo de los negocios. Ciertamente si se está trabajando en el desarrollo de una plataforma de colaboración de B2B, se sugiere que se incluya en la estrategia la adopción del uso del XML, sin duda todos los esfuerzos se concentrarán en esta herramienta.

El lenguaje XML para páginas Web ofrece múltiples aplicaciones, y es aun algo nuevo que pocos conocen y saben utilizarlo, pero es accesible ya que se puede usar como extensión HTML, es cuestión de profundizar poco a poco en éste para ir conociéndolo para posteriormente obtener un perfil más profesional, accesible y eficaz a Internet.

### **2.4.1 Servicios Web en el futuro**

El Web esta pasando de una gran librería estática con el HTML, al intercambio dinámico de cualquier tipo de información, que pueda ser indexada, ordenada, clasificada y personalizada de acuerdo a las necesidades. Objetivos que hace posible XML.

El trabajo sobre la plataforma de servicios Web continuará su desarrollo y pronto aparecerán mejoras en tres áreas fundamentales. En primer lugar, se añadirán servicios de más alto nivel. Todo el mundo está de acuerdo en que debe existir un modo estándar de hacer más seguros los servicios Web, enviar mensajes, garantizar una entrega fiable de mensajes, definir semántica transaccional, etc. Estas características de propósito general se expanden más allá de los dominios del problema y no hay ninguna razón por la que cada desarrollador de servicios Web deba implementarlas individualmente. Microsoft, IBM y otros están realizando mucho trabajo en estas áreas.

La iniciativa Global XML GXA define un conjunto de especificaciones sobre cómo implementar estos servicios de infraestructura en términos de mensajes SOAP (por ejemplo, de un modo neutral respecto del protocolo de transporte).

En segundo lugar, seguirán estandarizándose especificaciones. El ciclo de vida de las especificaciones de servicios Web típicamente progresa desde una propuesta hasta un estándar de facto y desde éste hasta un estándar real. Con SOAP 1.2 y WSDL 1.2 las peculiaridades finales están siendo elaboradas de las especificaciones SOAP y WSDL y algunas de las especificaciones de servicios de mayor nivel, como WS-Security, ya están en el periodo “de facto” en el proceso de estandarización. Las empresas siguen proponiendo nuevas especificaciones como añadidos a la plataforma de servicios web y la industria en su conjunto necesitará acordar cuáles adoptar. Esas especificaciones necesitarán a continuación ser estandarizadas.

En tercer lugar, las herramientas y marcos de trabajo seguirán mejorando. Además de servicios de más alto nivel como la seguridad y los objetos adjuntos, se añadirá soporte para protocolos de transporte alternativos como SMTP o TCP. De modo más importante, los modelos de programación migrarán desde los servicios de tipo RPC hacia servicios centrados en documentos, para promover un acoplamiento débil. Todos estos cambios ocurrirán en paralelo, mientras los desarrolladores continúan desarrollando e implantando sistemas basados en servicios web.

---

---

# Capítulo 3

## JAVA

Como cualquier lenguaje humano, Java proporciona una forma de expresar conceptos. Si tiene éxito, la expresión media será significativamente más sencilla y más flexible que las alternativas, en medida que los problemas crecen en tamaño y complejidad.

Al igual que los demás lenguajes de programación que le precedieron, Java es una mezcla de los mejores elementos de su rica herencia combinados con los conceptos innovadores que requiere su entorno único. Aunque el lenguaje Java se ha convertido en algo inseparable del entorno de Internet, es importante recordar que éste es ante todo un lenguaje de programación.

No es posible asimilarlo como una simple colección de características debido a que algunas de éstas no tienen sentido aisladas o separadas. Por tanto el enfoque con que merece ser enfrentado es utilizando la suma de elementos sólo si se piensa en diseño, y no en simple codificación.

### 3.1 La creación de Java

Para comprender éste lenguaje, es necesario comprender las razones que condujeron a su creación, las fuerzas que lo modelaron, y el legado que hereda.

Java fue concebido por James Gosling, Patrick Naughton, Chris Warth, Ed Frank y Mike Sheridan en Sun Microsystems, Inc. en 1991. Se tardó dieciocho meses en desarrollar su primera versión. El lenguaje se llamó en un principio "Oak" pero después fue renombrado como "Java" en 1995. Entre la implementación inicial de Oak, en otoño de 1992, y el anuncio público en la primavera de 1995, muchas personas contribuyeron al diseño y desarrollo del

lenguaje. La contribución especial de Bill Joy, Honathan Payne, Frank Yellin y Tim Lindholm permitió alcanzar la madurez del prototipo original.

Algo sorprendente es que el impulso inicial para el lenguaje no fue Internet, sino la necesidad de un lenguaje de programación que fuera independiente de la plataforma, y que pudiera ser utilizado para crear software para diversos dispositivos electrónicos, como hornos de microondas, lavadoras y mandos a distancia.

Existen muchos tipos de CPU (*Unidad Central de Procesamiento*) que se utilizan como controladores. El inconveniente de C y C++ es que están diseñados para ser compilados por un elemento específico, aunque casi cualquier CPU puede compilar un programa escrito en C++, es necesario tener el compilador determinado, el problema es que resulta caro crear un compilador además de que requiere un tiempo determinado.

Por este motivo era necesario encontrar una solución más sencilla y eficiente en lo que se refiere al costo. Gosling y otros, en un intento de hallar la solución perfecta, comenzaron a trabajar en el desarrollo de un lenguaje de programación portable, que fuera independiente de la plataforma y que se pudiera ejecutar en distintos tipos de CPU bajo diferentes entornos, este esfuerzo fue el que condujo finalmente a la creación de Java.

Mientras se trabajaban distintos aspectos del lenguaje, surgió un segundo factor, y definitivamente el más importante que tendría un papel crucial en el futuro de Java, este factor fue: la *WWW*, si la Web no se hubiese desarrollado al mismo tiempo en que Java estaba siendo implementado, éste podría haber sido simplemente un lenguaje de utilidad para la programación de dispositivos electrónicos de consumo.

Sin embargo, con la aparición y desarrollo de la *WWW*, se ha visto impulsado al primer lugar en el diseño de los lenguajes de programación, porque la Web también demandaba programas portables, es decir independientes de la plataforma.

Aunque la búsqueda de programas eficientes, portables es tan antigua como la propia disciplina de la programación, ha ocupado un lugar secundario en el desarrollo de los lenguajes, debido a problemas cuya solución era más urgente.

Por otro lado, la mayoría de los sistemas operativos de las computadoras en el mundo se dividen en tres grandes grupos: Microsoft, Linux y UNÍX. Por ello muchos de los programadores se han mantenido en sus fronteras sin la necesidad de un código portable.

En 1993, para el equipo que estaba diseñando el lenguaje Java resultó obvio que el problema de la portabilidad cuando se creaba código para los controladores, era exactamente el mismo para crear código para Internet. En cierto sentido el mismo problema que Java intentaba resolver a pequeña escala se extrapolaba también a Internet en gran escala, esto hizo que cambiara su orientación pasando de ser aplicado a los dispositivos electrónicos de consumo a la programación en Internet. Por tanto, aunque la motivación inicial fue la de proporcionar un lenguaje de programación multiplataforma, ha sido Internet quien finalmente ha conducido al éxito de Java a gran escala.

Este lenguaje se basa en C y C++, sus diseñadores utilizaron la sintaxis de C y las características orientadas a objetos de C++ para conseguir que su nuevo lenguaje atrajese a muchos programadores experimentados por las semejanzas evidentes entre estos lenguajes, ya que comparte los atributos que lograron la aceptación de sus antecesores. Por tanto es un lenguaje coherente y consistente. Excepto por las restricciones que impone el entorno de Internet, además, permite al programador un control total, por lo tanto es un lenguaje para programadores profesionales.

## **3.2 Características de Java**

### **3.2.1 Sencillo**

Ofrece toda la funcionalidad de un lenguaje potente, pero sin las particularidades sofisticadas que provocan confusiones. Cabe mencionar que fue diseñado con la finalidad de que su aprendizaje y utilización resultaran naturales

---



para el programador profesional. Es fácil dominar el lenguaje, suponiendo que se tiene cierta experiencia programando. Si además se comprenden los conceptos de la programación orientada a objetos, aprenderlo es, incluso, sencillo. Y lo mejor de todo es que, si se tiene experiencia como programador de C / C++, cambiar a Java requiere un mínimo esfuerzo.

Al mismo tiempo muchos de los conceptos más complejos quedan descartados o se han implementado de una forma más clara, más accesible.

Más allá de sus semejanzas con sus antecesores, se tiene otra ventaja importante. Es fácil su aprendizaje y se ha pensado con el fin de que existan pocas formas de realizar una tarea determinada.

Java elimina muchas de las particularidades que adolecen otros lenguajes. Por ejemplo, mantiene reducidas las especificaciones del lenguaje y también añade otras propiedades muy útiles como el *"garbage collector"* que es un reciclador de memoria dinámica. No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un hilo de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria.

También elimina algunas complejidades que prevalecen en otros lenguajes como "C ó C++" entre las cuales destacan:

- ❖ Aritmética de apuntadores.
- ❖ No existen referencias como en C.
- ❖ Registros (struct).
- ❖ Definición de tipos (typedef).
- ❖ Macros (#define).
- ❖ Necesidad de liberar memoria.

El intérprete completo de Java sigue siendo muy pequeño, ya que ocupa aproximadamente 215 KB de RAM.

### 3.2.2 Orientado a Objetos

El lenguaje Java no fue diseñado para tener un código fuente compatible con cualquier otro lenguaje. Esto dio la libertad de partir de cero. Una consecuencia de esto fue una aproximación clara, pragmática y aprovechable a los objetos. Java ha tomado prestadas muchas ideas de entornos orientados a objetos de las últimas décadas, consiguiendo un equilibrio espectacular entre el modelo purista “**todo es un objeto**” y el modelo pragmático. El modelo de objetos es sencillo y de fácil ampliación, mientras que los tipos simples, como los enteros, se mantienen como “**no objetos**” de alto rendimiento.

Trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma orientado a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias, las cuales necesitan ser construidas y destruidas en espacios de memoria.

Incorpora la resolución dinámica de métodos. Esta característica deriva del lenguaje Objetivo C. En C++ se suele trabajar con bibliotecas dinámicas que obligan a recompilar la aplicación cuando se retocan las funciones que se encuentran en su interior. Este inconveniente es resuelto por mediante una interfaz específica llamada RTTI (*RunTime Type Identification*) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el tiempo de ejecución “*runtime*” que permite a los programadores hacer llamadas a este tipo de clase y enlazarla dinámicamente con el resultado de la búsqueda.

### 3.2.3 Distribuido

Se ha construido con extensas capacidades de interconexión TCP/IP. Existen bibliotecas que permiten interactuar con protocolos como http y ftp. Esto permite a los programadores disponer de información a través de la red con la misma facilidad como si se estuvieran leyendo y escribiendo sobre archivos de datos. La

versión original de Java (Oak) incluía facilidades que permitían a objetos situados en computadoras diferentes ejecutar procedimientos de forma remota. Java ha retomado estas interfaces en un paquete denominado RMI (*Remote Method Invocation*) Invocación a métodos remotos. Esta característica aporta un nuevo nivel de abstracción en la programación cliente / servidor.

Java en sí no es distribuido, pero proporciona las herramientas para que los programas sí puedan ser distribuidos es decir, que trabajen en máquinas y plataformas diferentes.

### 3.2.4 Robusto

El entorno multiplataforma de la Web es muy exigente con un programa, ya que éste se debe ejecutar de forma fiable en una gran variedad de sistemas. Por tal motivo, la capacidad para crear programas robustos tuvo una alta prioridad en el diseño de Java. Para ganar confiabilidad, se restringe al programador en unas pocas áreas clave, y así se consigue encontrar rápidamente los errores en el desarrollo del programa. Al mismo tiempo, libera al programador de preocuparse por las causas más comunes de error. Debido a que es un lenguaje estrictamente tipificado, realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Una característica clave es que se conoce el comportamiento del programa de una manera predecible en diversas condiciones.

Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Además, maneja la memoria para eliminar la necesidad de hacerlo por parte del programador. El manejo de la memoria es completamente automática ya que dispone de un sistema que se encarga de los objetos que ya no se utilizan conocida como (*garbage collector*).

Implementa arreglos, en lugar de listas enlazadas de apuntadores, con comprobación de límites, para evitar la posibilidad de sobre escribir o corromper memoria resultado de los apuntadores cuando señalan a áreas de memoria

equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java.

Para asegurar el funcionamiento de la aplicación, se realiza una verificación de los “*byte-codes*”, que son el resultado de la compilación de un programa Java. Los “*byte-codes*” es un código de máquina virtual interpretado por el intérprete. No es el código máquina directamente entendible por el hardware, pero ya ha pasado todas las fases del compilador: análisis de instrucciones, orden de operadores, etc., y ya tiene generada la pila de ejecución de órdenes.

En los entornos tradicionales, las excepciones surgen en situaciones especiales que se presentan tales como una división entre cero. Estas situaciones son capturadas y deben ser administradas mediante construcciones torpes y difíciles de leer. En esta área, se proporciona una variada y extensa administración de excepciones orientada a objetos. En un programa de Java correctamente escrito, todos los errores de ejecución pueden y deben ser administrados por el programa.

### **Java proporciona:**

- Comprobación de apuntadores.
- Comprobación de límites de arreglos.
- Manejo de Excepciones.
- Verificación de “*byte-codes*”.

### **3.2.5 Arquitectura neutral**

Una cuestión importante para los diseñadores de Java era la vigencia y portabilidad del código. Uno de los principales problemas a los que se enfrentan los programadores es que no tienen garantía de que el programa que escriben hoy podrá ejecutarse mañana.

Las actualizaciones de los sistemas operativos, los procesadores, y los cambios en los recursos básicos del sistema en su conjunto, pueden hacer que un programa funcione mal. Los diseñadores de Java tomaron decisiones difíciles en el lenguaje y en el interprete Java (JVM), en un intento por mejorar esta situación. Su meta se basó en la lógica **“Escríbese una vez ejecútese donde sea, en cualquier momento”**. Ese objetivo se consiguió en gran medida.

Para hacer de las aplicaciones escritas en Java, mecanismos que conformen una parte integral en un sistema independiente, se compila su código en un archivo objeto de formato que no depende de la arquitectura de la máquina en que será ejecutada. Cualquier máquina que tenga el sistema **“run-time”** puede ejecutar ese código objeto, sin importar la máquina en que ha sido generado. Actualmente existen sistemas **“run-time”** para Solaris, SunOs, Windows, Linux, Irix, Aix, Mac, sistemas operativos para dispositivos móviles y probablemente haya grupos de desarrollo trabajando en la portabilidad a otras plataformas.

El código fuente se compila a un código de bytes de alto nivel independiente de la máquina. Este código está diseñado para ejecutarse por un sistema que es dependiente de la máquina.

Podemos apreciar que lo que depende verdaderamente del sistema es la denominada Máquina Virtual de Java, así como sus bibliotecas básicas las cuales permiten tener acceso directamente al hardware de la máquina.

### 3.2.6 Seguridad

Como se sabe, cada vez que se transfiere un programa ejecutable a una máquina, se corre el riesgo de que el sistema sea infectado por algún virus.

No ha pasado mucho tiempo en que la mayoría de los usuarios evitaban descargar programas ejecutables, y cuando lo hacían, éstos eran examinados minuciosamente para detectar posibles virus.

Hoy sabemos que no solo los virus son un riesgo y un problema a enfrentar, también existen algunos programas escritos con el fin de corromper la seguridad

de los sistemas. Estos programas recompilan información privada como pueden ser números de tarjetas de crédito, cuentas bancarias, claves de acceso, etc. realizando una búsqueda en el sistema de archivos de la computadora local.

El lenguaje Java fue escrito pensando en ser un lenguaje lo más seguro posible. Así, la manera de hacer una instancia implícita como lo hacen los compiladores de C y C++ y el uso de apuntadores son eliminados con el objeto de prevenir accesos ilegales a la memoria.

Incluso cuando se utiliza Java para crear un navegador, se combinan características propias del lenguaje con protecciones de sentido común aplicadas al propio navegador.

En el lenguaje C, por ejemplo, se tienen algunas vulnerabilidades de seguridad importantes, como son los errores de alineación. Los programadores de C utilizan apuntadores en conjunto con operaciones aritméticas. Esto le permite al programador que un apuntador haga referencia a un lugar conocido de la memoria y pueda realizar operaciones para referirse a otro lugar de la memoria. Si otros programadores conocen nuestras estructuras de datos es posible que puedan extraer información confidencial de nuestro sistema.

Otro tipo de ataque, es el Caballo de Troya. Se presenta un programa como una utilidad, resultando tener una funcionalidad destructiva.

Por ejemplo, en UNIX se visualiza el contenido de un directorio con el comando "ls". Si un programador deja un comando destructivo bajo esta referencia, se puede correr el riesgo de ejecutar código malicioso, aunque el comando tenga la funcionalidad para lo cuál fue desarrollado.

El código en Java pasa por una serie de pruebas antes de ejecutarse en una máquina. Dicho código se pasa a través de un verificador de "byte-codes" que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal, que viola derechos de acceso sobre objetos o que intenta cambiar el tipo o clase de un objeto.

Cuando se utiliza un navegador compatible con Java, es posible descargar pequeñas aplicaciones conocidas como “*applets*” de forma segura sin temor a los virus o intentos de acceso maliciosos. Se consigue esta protección limitando el programa al entorno de ejecución no permitiéndole el acceso a otras partes del sistema. Incluso muchos consideran que esta capacidad, la de transferir “*applets*” con la seguridad de que no se producirá ningún daño y de que no se violará la seguridad, es la característica más importante de Java.

Si los “*byte-codes*” pasan la verificación sin generar ningún mensaje de error, entonces se sabe que:

- ❖ El código no produce errores de desbordamiento de memoria (*overflow*).
- ❖ El tipo de los parámetros de todos los códigos de operación son conocidos y correctos.
- ❖ No ha ocurrido ninguna conversión ilegal de datos, tal como convertir enteros en apuntadores.
- ❖ Se asegura que el acceso a objetos sea válido mediante especificadores que pueden ser de tipo público, privado o protegido, y son éstos quienes determinan que parte del código se encuentra disponible para el programador y que parte no lo es. Estos niveles de control de acceso se encuentran desde el nivel máximo de disponibilidad cuando son públicos por ejemplo, hasta el mínimo cuando son privados.
- ❖ No hay ningún intento de violar las reglas de acceso y seguridad establecidas.

El Cargador de Clases también ayuda a mantener la seguridad, separando el espacio de nombres del sistema de archivos local, al de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior.

Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen. Cuando una clase del espacio de nombres privado accede a otra clase, primero se busca en las clases predefinidas (del sistema local) y luego en el espacio de nombres de la clase que hace la referencia. Esto imposibilita que una clase suplante a una predefinida.

En resumen, las aplicaciones de Java resultan extremadamente seguras, ya que no acceden a zonas delicadas de memoria o de sistema, por tanto, evitan la interacción con virus, debido a que no posee una semántica específica para modificar la pila de programa, la memoria libre o utilizar objetos y métodos de un programa sin los privilegios del kernel del sistema operativo. Además, para evitar modificaciones por parte de aplicaciones maliciosas, implementa un método muy seguro de autenticación por clave pública. El Cargador de Clases puede verificar una firma digital antes de realizar una instancia de un objeto. Por tanto, ningún objeto se crea y almacena en memoria, sin que se validen los privilegios de acceso. Es decir, la seguridad se integra en el momento de compilación, con el nivel de detalle y de privilegio que sea necesario.

Debido a la concepción del lenguaje en sí y que todos los elementos se mantienen dentro del estándar marcado por Sun, no hay peligro. Java imposibilita, también, abrir ningún archivo de la máquina local (siempre que se realizan operaciones con archivos, éstos trabajan sobre el disco duro de la máquina de donde partió el "*applet*"), no permite ejecutar ninguna aplicación nativa de una plataforma e impide que se utilicen otras computadoras como puente, es decir, nadie puede utilizar la máquina en que se desarrolla para hacer peticiones o realizar operaciones sobre otra. Además, los intérpretes que incorporan los navegadores de la Web son aún más restrictivos. Bajo estas condiciones (y dentro de la filosofía de que el único sistema seguro es aquel que está apagado, dentro de una cámara acorazada en un bunker y rodeado por soldados del ejército), se puede considerar que es un lenguaje seguro y que los applets están libres de virus.



Respecto a la seguridad del código fuente, JDK proporciona un desensamblador de *"byte-code"*, que permite que cualquier programa pueda ser convertido a código fuente, lo que para el programador significa una vulnerabilidad total a su código. Utilizando javap no se obtiene el código fuente original, pero sí desmonta el programa mostrando el algoritmo que se utiliza, que es lo realmente interesante. La protección de los programadores ante esto es usar llamadas a programas nativos, externos incluyendo C, C++, Perl, Python, de manera que no sea compilable todo el código; aunque así se pierda portabilidad. Esta es otra de las cuestiones que se tienen pendientes en Java.

### 3.2.7 Portabilidad

Hoy en día muchos tipos de sistemas operativos como equipos de computadoras están siendo utilizados por todo el mundo, éstos, generalmente están conectados a Internet, para que cualquier programa pueda transferirse dinámicamente a los distintos tipos de plataformas conectados a Internet, es necesario que exista un medio que permita generar código ejecutable y portable. El mismo mecanismo que sirve para garantizar la seguridad en Java, sirve también para crear la portabilidad. Por lo tanto, la solución que brinda a estos dos problemas es elegante y al mismo tiempo eficiente.

### El *bytecode* de Java

Tanto el problema de la seguridad como el de la portabilidad son resueltos en Java debido a que el código fuente es procesado por el compilador y el código resultante no es ejecutable, sino *"byte-code"*.

El *"byte-code"* son un conjunto de instrucciones altamente optimizadas que han sido diseñadas para ser ejecutadas por una máquina virtual que emula al intérprete Java.

Esto puede resultar en cierta medida sorprendente. Como se sabe, el lenguaje C++ es compilado para generar un código ejecutable. De hecho, la mayoría de los lenguajes modernos se han diseñado para ser compilados, no tanto interpretados. Sin embargo, el hecho de que un programa sea interpretado permite resolver los problemas más importantes asociados a la transferencia de programas en Internet.

Traducir un programa Java a *"byte-code"* hace que su ejecución resulte mucho más sencilla, y la razón es que para cada plataforma, sólo es necesario implementar el intérprete. Una vez que se dispone del programa de ejecución para un sistema determinado, cualquier programa puede ejecutarse en esa plataforma. Si Java fuera un lenguaje compilado, entonces tendrían que existir diferentes versiones del mismo programa para cada tipo de CPU conectada a Internet. Ya que esta solución no es factible, se puede decir que la interpretación del *"byte-code"* es la forma más sencilla de crear programas auténticamente portables.

El hecho de que el lenguaje sea compilado y al mismo tiempo interpretado, logra en gran medida hacerlo un lenguaje muy seguro. Ya que la ejecución de cada programa escrito en éste lenguaje está bajo el control del intérprete, se puede contener al programa e impedir que se generen efectos no deseados en el resto del sistema, ciertas restricciones que existen sirven para mejorar la seguridad.

Cuando un programa es interpretado generalmente su ejecución es substancialmente más lenta que si fuese compilado; sin embargo, esta diferencia no es muy grande. El uso del *"byte-code"* habilita al intérprete para que los programas se ejecuten mucho más rápido de lo esperado.

### 3.2.8 Interpretado

El intérprete Java que es un sistema *"run-time"* puede ejecutar directamente el código objeto. Enlazar un programa, normalmente, consume menos recursos que compilarlo, por lo que los desarrolladores pasarán más tiempo desarrollando y menos esperando por la máquina. No obstante, el compilador actual del JDK es bastante lento en comparación con otros lenguajes de programación, por el hecho de que debe ser interpretado y no ejecutado como sucede en cualquier otro lenguaje tradicional.

Se dice que el lenguaje es entre 10 a 30 veces más lento que C y en otras proporciones con otros lenguajes de programación. Entre ésta y aún mas aseveraciones, existe una reñida discusión entre que lenguaje de programación es más rápido y eficiente, generando una cantidad de información en pro y en contra de los distintos lenguajes contendientes con Java.

Para ser un lenguaje independiente del sistema operativo y del procesador que incorpore la máquina utilizada, es tanto interpretado como compilado, esto no es ningún contrasentido ya que el código fuente escrito con cualquier editor se compila generando *"byte-code"*. Este código intermedio es de muy bajo nivel, pero sin alcanzar las instrucciones máquina propias de cada plataforma y no tiene nada que ver con el *"p-code"* de Visual Basic. El *"byte-code"* corresponde al 80% de las instrucciones de la aplicación. Ese mismo código es el que se puede ejecutar sobre cualquier plataforma. Para ello hace falta el *"run-time"*, que sí es completamente dependiente de la máquina y del sistema operativo, que interpreta dinámicamente el código y añade el 20% de instrucciones que faltaban para su ejecución. Con este sistema es fácil crear aplicaciones multiplataforma, pero para ejecutarlas es necesario que exista el correspondiente al sistema operativo utilizado.

### **3.2.9 Multithilo**

Fue diseñado para satisfacer los requerimientos del mundo real, de crear programas en red interactivos. Para ello, proporciona la programación multithilo que permite la escritura de programas que hagan varias cosas simultáneamente. El intérprete dispone de una solución elegante y sofisticada para la sincronización de múltiples procesos que permiten construir fácilmente sistemas interactivos. El método multithilo, de utilización sencilla, permite ocuparse únicamente del comportamiento específico del programa, en lugar de pensar en el sistema multitarea.

Los “*hilos*” a veces llamados, procesos ligeros, son básicamente pequeñas aplicaciones o piezas independientes que en su conjunto realizan una tarea en específico, al estar contruidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++.

El beneficio de ser multithilo ó multivanado, consiste en tener un mejor rendimiento que es interactivo y por ende, un mejor desempeño en tiempo real. Aunque este comportamiento está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, Linux, etc).

Cualquiera que haya utilizado la tecnología de navegación concurrente, sabe lo frustrante que puede ser esperar por una gran imagen que se está trayendo. En Java, las imágenes se pueden ir trayendo en un “*hilo*” independiente, permitiendo que el usuario pueda acceder a la información en la página sin tener que esperar por el navegador.

### **3.2.10 Dinámico**

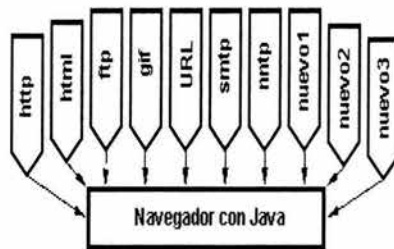
Los programas Java se transportan con cierta cantidad de información que es utilizada para verificar y resolver el acceso a objetos en el tiempo de ejecución. Esto permite enlazar el código dinámicamente de una forma segura y viable. Esta característica especial contribuye de manera importante a la robustez del entorno

de los “*applets*”, en el que pequeños fragmentos de “*byte-code*” pueden ser actualizados dinámicamente en un sistema que está ejecutándose.

Java se beneficia todo lo posible de la tecnología orientada a objetos no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las bibliotecas nuevas o actualizadas no paralizarán las aplicaciones actuales.



Monolito: cada pieza de código se compacta dentro del código del navegador



Sistema Federado: el navegador es un coordinador de piezas, y cada pieza es responsable de una función. Las piezas se pueden añadir dinámicamente a través de la red

También simplifica el uso de protocolos nuevos o actualizados. Si un sistema ejecuta una aplicación sobre la red y encuentra una pieza de la aplicación que no puede manejar, es capaz de hacer una llamada automáticamente a cualquiera de los elementos que el sistema necesita para funcionar.

Para evitar que los módulos, los objetos o nuevas clases se tengan que importarse continuamente de la red cada vez que se requieran, el lenguaje implementa las opciones de persistencia, para que no se eliminen cuando se limpie el caché de la máquina.

Con la implementación de Java se tienen las siguientes ventajas y beneficios:

- ❖ **Primero:** No es necesario volver a escribir el código si se quiere ejecutar el programa en otra máquina. Un solo código funciona para todos los navegadores compatibles con Java o donde se tenga una Máquina Virtual de Java (Mac, PC, Sun, etc).
- ❖ **Segundo:** Es un lenguaje de programación orientado a objetos, y tiene todos los beneficios que ofrece esta metodología de programación.
- ❖ **Tercero:** Un navegador compatible con Java deberá ejecutar cualquier programa hecho en ese mismo lenguaje, esto ahorra a los usuarios el tener que estar insertando aplicaciones llamadas "plug-ins" y demás programas que muchas veces quitan tiempo y espacio en el disco.
- ❖ **Cuarto:** Es un lenguaje y por lo tanto puede hacer todas las cosas que puede hacer un lenguaje de programación: Cálculos matemáticos, procesadores de palabras, bases de datos, aplicaciones gráficas, animaciones, sonido, hojas de cálculo, etc.
- ❖ **Quinto:** Si el enfoque son las páginas de Web, ya no se tiene que ser estáticas, se le pueden poner toda clase de elementos multimedia y permiten un alto nivel de interactividad.

**También existen algunas limitantes:**

- ❖ **Velocidad:** Los programas escritos en este lenguaje no son muy rápidos, supuestamente se está trabajando para mejorar esto. Como los programas son interpretados, nunca alcanzan la velocidad de un verdadero programa ejecutable.
- ❖ **Java es un lenguaje de programación:** Esta es otra gran limitante, por más que se diga que es un lenguaje orientado a objetos y que es fácil de

aprender, sigue siendo un lenguaje y por lo tanto aprenderlo requiere tiempo y esfuerzo. Especialmente para los no programadores.

- ❖ **Es nuevo:** En pocas palabras todavía no se conocen bien todas sus capacidades.

Pero en general Java posee muchas ventajas es posible desarrollar aplicaciones interesantes. Muchas de las implementaciones de Java actuales son sólo intérpretes.

Los “*byte-codes*” son interpretados por el sistema “*run-time*” de Java sobre la máquina del usuario. Aunque ya hay ciertos proveedores que ofrecen compiladores nativos JIT (*Just-In-Time*). Si la Máquina Virtual Java dispone de un compilador instalado, las secciones (clases) del “*byte-code*” de la aplicación se compilarán hacia la arquitectura nativa de la máquina del usuario.

### 3.3 Herramientas de Desarrollo

Es necesario realizar muchas mejoras antes de que Java pueda considerarse una plataforma importante de desarrollo. Muchos lenguajes de programación empiezan como lenguajes basados en texto. Se escriben los programas de manera sencilla y éstos se ejecutan de la misma manera, las interfaces y las salidas también son de tipo texto.

Desde este punto de vista, con frecuencia se agregan interfaces gráficas para usuario GUI (*Graphic User Interface*), las cuales permiten una interacción natural con los programas y los vuelven más amigables y accesibles para el usuario.

Dichas interfaces de programación basadas en gráficos se conocen como IDE (*Integrated Development Enviroments*) que en gran medida facilitan la programación en cualquier lenguaje. Para comprender lo útiles que son, habría que dar un vistazo a los métodos actuales de Java.

Un programa se crea empleando un editor de texto. El código se ejecuta a través de un compilador de Java, y si existen errores, el compilador genera una lista de detallada de los mismos. Por lo que se tendrá que regresar al editor y corregirlos. Después de arreglar los errores, se vuelve a compilar el código. Si todo se corrigió correctamente, entonces esta listo para ejecutarse. En caso de que no haya ningún error, pero el programa no hace lo que se requiere, entonces habrá que volver a comenzar.

Como puede verse este proceso puede ser un tanto tedioso. Con un IDE, el proceso sería el siguiente:

Se crea el código mediante un editor y luego hace clic en un botón que al ser presionado compila el código.

El compilador resalta los posibles errores en el código y se pueden corregir desde la ventana de edición. Se puede compilar sin abandonar el editor. Cuando los errores estén corregidos, el IDE ejecuta el código de manera automática. Ahora se puede ver como se ejecuta el *applet* y corregir errores en la ventana de edición.

En la actualidad el desarrollo de Java se realiza por completo en la línea de comando. No existe un editor de Java especializado y el "*appletviewer*" no es un ambiente en tiempo de ejecución completo.

### **3.3.1 Integración con otras Aplicaciones**

El acceso a métodos nativos o compiladores cruzados puede estar disponible en un futuro próximo para ayudar a incrementar el diseño y el desempeño de las aplicaciones.

Incluso podría interconectar colecciones de aplicaciones para crear algunas más grandes y sofisticadas. Además, con la ayuda de algunas herramientas de creación de scripts, como "*JavaScript*" es posible sacar aún más provecho y de esta manera producir programas más grandes sin tener que escribir código Java.



### 3.4 Javascript

JavaScript es uno de los avances más prometedores en el horizonte de Java. Desde sus inicios en Netscape fue un lenguaje diseñado para brindar capacidades de escritura en el WWW.

Lo importante de Java Script, es la capacidad de manipular objetos de Web y de Java sin emplear un lenguaje de programación de alto nivel. JavaScript permite que los usuarios personalicen sus *applets* e incluso que puedan controlar el funcionamiento o ambiente en el que se ejecutan. También es una poderosa herramienta para escribir juntos los *scripts* de objetos Java en lugar de emplearlos en código.

En sus etapas actuales, JavaScript está limitado a la escritura de *scripts* para varias funciones de Web y de visualizador, hacer que los *applets* funcionen con Java Script involucraría por lo tanto un desarrollo de ambas tecnologías. Consecuentemente, será necesario contar con métodos aceptados para controlarlos y personalizarlos.

El lenguaje Java también deberá incorporar métodos para intercambiar control y comunicarse con los códigos de JavaScript. La promesa de flexibilidad, facilidad e implementación tendrá que esperar hasta Futuras versiones.

### 3.5 El Futuro de Java

A continuación se dará un vistazo a lo que el futuro le promete para este lenguaje de programación.

El desarrollo de aplicaciones web se ha convertido en una batalla entre las nuevas tecnologías que compiten duramente por hacerse un espacio en el mercado. El objetivo es conseguir un nuevo entorno distribuido y heterogéneo de

desarrollo, pero hasta que se consiga una solución que asuma el liderazgo real las incompatibilidades técnicas seguirán siendo el precio a pagar.

Java y XML son los dos grandes pilares sobre los que descansan las esperanzas multiplataforma de los desarrolladores de aplicaciones web. Obviamente, estos estándares no han eliminado el desarrollo propietario de API, formatos y tecnologías por parte de los fabricantes, pero gracias a su amplia aceptación, están haciendo posible que los desarrolladores de software construyan aplicaciones Web no demasiado rudimentarias capaces de funcionar en múltiples entornos de cliente y de ser portadas a varios servidores web, aunque en México la aceptación no ha sido total, Java ya es un gran protagonista.

A diferencia de otros lenguajes de programación que normalmente se van estableciendo a base de pequeñas mejoras, Java ha continuado evolucionando a un ritmo exponencial. Poco después de la versión 1.0, los diseñadores ya habían creado la versión 1.1. Las facilidades que añadía la versión 1.1 eran más significativas y substanciales de lo que cabría pensar teniendo en cuenta el número asignado a la versión. La siguiente versión fue Java 2, que marcó el comienzo de la era “moderna” del lenguaje que evolucionaba rápidamente. Java 2 añade nuevas facilidades, tales como los componentes Swing y las Colecciones (*Collections*), y mejoraba el intérprete así como varias herramientas de programación.

A raíz de su aparición, el lenguaje de programación Java sigue siendo uno de los grandes elementos catalizadores de la red. La retórica generada en torno a Sun habla de Java como el verdadero factor multiplataforma, pero no faltan las críticas que lo califican de demasiado lento, molesto e insuficientemente escalable como para satisfacer las necesidades de las grandes organizaciones. Como siempre, es difícil conseguir la unanimidad de los expertos y usuarios, máxime cuando tantos intereses industriales están de por medio.

Java, lejos de concentrarse en un desarrollo común, se ha fragmentado en campos cada vez más separados propiciados respectivamente por firmas como la propia Sun, Netscape, Microsoft y otros grandes compañías . Todos prometen

lealtad al objetivo de conseguir un único conjunto de estándares Java, pero no hay dos fabricantes de este tipo de software que ofrezcan el mismo conjunto de plantillas, Beans (interfaz de programación de Sun que permite construir aplicaciones o partes de programa reutilizables en forma de componentes), clases, métodos y otras características. Es decir, antes de que los desarrolladores de aplicaciones, si es que pueden, cumplan la premisa de “escribir una vez y ejecutar en cualquier sitio”, han de pasar por un tedioso proceso de diseño, análisis y planificación.

### **Java servirá para programar cualquier tipo de aplicación**

JavaOne promocionado por Sun, está hablando del futuro de la red, que según la mayoría de los desarrolladores estará muy ligado al lenguaje de programación Java.

La nueva versión de Java incluye interfaces que permiten programar cualquier aplicación, desde dispositivos móviles hasta aplicaciones robustas alojadas en servidores, incluye herramientas para desarrollo de entornos gráficos, seguridad y mayor soporte para otros idiomas. Además se prevén otras aplicaciones, como transacciones, sonido, tratamiento avanzado de imágenes, etc.

Los programadores podrán cambiar fácilmente el aspecto de sus programas, que podrán parecer de entorno Mac, de PC o cualquier otra combinación de ambos, que es como tener disponible una interfaz dinámica para cada usuario.

De lo anterior Java consigue así, establecer la piedra angular para las computadoras, y la única forma de practicar una computación segura. JavaSoft pretende pasar del lenguaje de programación, a la plataforma de desarrollo con la creación de JavaBeans, componentes que pueden interactuar entre sí, escritos completamente en Java.

Un ejemplo de esto fue un navegador sencillo, construido exclusivamente con JavaBeans, que sirvió como demostración de las posibilidades de esta herramienta. El programa se creó en muy poco tiempo y tenía un tamaño de sólo 300 KB.

### ¿Cuál será el futuro de la seguridad en Java?

Actualmente se trabaja en la API de Seguridad de Java, que tiene como meta el proporcionar servicios de validación y cifrado para las aplicaciones Java. Debido a la extrema dependencia de la seguridad a lo que se decida en este API, ha surgido una gran polémica en torno a qué medidas de seguridad adicionales deberán incorporarse.

Un rasgo típico de que ya gozan los applets, es la capacidad de ser firmadas por alguna autoridad de certificación. De esta manera, el usuario puede decidir si la aplicación recibida es de confianza o no. Las applets seguras tendrán el privilegio de acceder de forma controlada a los recursos, mientras que al resto se les negará el mismo.

---

## Capítulo 4

### EVALUACIÓN DE HERRAMIENTAS

Existe un impresionante número de herramientas para la construcción de aplicaciones Web, sin embargo, una revisión cuidadosa de sus características revela que la mayoría de las soluciones se concentra en la implementación prestando poca atención al proceso de diseño de una aplicación para Internet.

Los marcos de trabajo de aplicaciones web que nos interesan en ésta tesis son los utilizados para la construcción de aplicaciones que publican documentos digitales en Internet y que se caracterizan por mantener grandes volúmenes de datos, por lo que las necesidades de disponer de herramientas que cubran todo el ciclo de vida del desarrollo de un proyecto resulta determinante si queremos diseñar una solución capaz de adaptarse a los cambios.

Debido al desarrollo de la tecnología Java y a las aportaciones en cuanto a software libre de la *Apache Software Foundation* a través del proyecto Jakarta, se ha aumentado la posibilidad de contar con componentes desarrollados por terceros en forma de bibliotecas de etiquetas basadas en XML, por lo que se ha facilitado el uso de las aplicaciones en esta tecnología.

Siendo este último rubro el que satisface prácticamente en su totalidad los requerimientos de desarrollo para una aplicación de publicación en línea, debido a se facilita compartir información de manera segura y práctica, cumpliendo con las expectativas planteadas en el marco teórico.

## **4.1 La importancia de XML**

Hasta ahora la Web ha sido diseñada directamente gracias a la colaboración de muchos desarrolladores. En un futuro no muy lejano, será necesario utilizar herramientas que permitan construir sistemas flexibles capaces de ofrecer servicios inteligentes tales como: agentes de búsqueda y filtros de información.

Para ello se requiere de una mayor funcionalidad e interoperabilidad en comparación con los sistemas actuales, por tanto, se ha promovido un fuerte impulso mediante la estandarización con XML por parte de la W3C.

Es un hecho que todos los desarrolladores de Internet están adoptando XML para resolver sus problemas de interoperabilidad con el manejo de la información.

### **4.1.1 Alternativas de desarrollo para la aplicación**

La complejidad de las soluciones crea la necesidad de contar equipo de trabajo capacitado, así como disponer de herramientas que ofrezcan soluciones efectivas a lo largo del ciclo del desarrollo de las aplicaciones. Por tanto, es necesario tomar en cuenta el planteamiento de ingeniería de software en todas las fases del desarrollo, obviamente no deberá estar excluida la fase de mantenimiento.

El problema de ingeniería que se presenta, debe estar sustentado bajo una metodología de proceso, por lo que es preciso ocupar modelos y herramientas para su análisis, diseño, pruebas, gestión entre otras.

Aunque muchas de estas tecnologías utilizadas en arquitecturas de Internet pudieran ser consideradas innovadoras, pueden tener la restricción de ser un tanto desconocidas, ajenas o incluso que únicamente interpreten de manera distinta el proceso de interacción con Internet.

Pero es preciso mencionar que aquellas que revolucionan la manera de hacer negocios efectivos y seguros a través de Internet, son las que cuentan con mayor importancia por ahora. Bajo este ámbito, las tecnologías de vanguardia hoy por hoy

son XML y Java, es posible que sigan vigentes o por lo menos así se vislumbra para los próximos 10 años.

XML se ha convertido en el formato estándar para la representación de datos en Internet. Estos datos se pueden procesar e interpretar en cualquier plataforma, desde un dispositivo móvil a una unidad central.

Java es el lenguaje de programación más adecuado para manipular XML. La mayoría de los analizadores de sintaxis de XML están escritos en Java, y proporciona una colección extensa de API (Interfaz para programas de aplicación) de Java pensada específicamente para construir aplicaciones basadas en él. La tecnología JSP tiene acceso a todo esto puesto que puede utilizar todo el poder de la plataforma Java para acceder a objetos del lenguaje de programación para analizar y transformar este tipo de documentos.

En la siguiente tabla se presentan algunas de las herramientas que pueden ser utilizadas para el desarrollo de una aplicación que maneje y administre datos a través de Internet. A continuación se muestran las características y particularidades para cada una de ellas, como son la infraestructura de puesta en punto ejemplo el modelo cliente – servidor que ocupan, como se efectúa la ejecución de las aplicaciones, es decir si se compila o se interpreta el código, las ventajas y desventajas que existen, el costo en consumo de procesamiento por mencionar algunas de las mas notables.

ESTA TESIS NO SALE  
DE LA BIBLIOTECA

HERRAMIENTA	RESIDENCIA	EJECUCIÓN	VENTAJAS	DESVENTAJAS
HTML	Cliente	Interprete	Estándar	No hace acceso a bases de datos
Java Script	Cliente	Interprete	Crea páginas HTML dinámicas	Puede tener fallos
CGI	Servidor	Compilador o Interprete	Ejecución eficiente	Código no portable
Java	Cliente	compilador	Portable	Requiere compilarse
JSP	Servidor	interprete	Portable	Requieren servidor JSP
ASP	Servidor	interprete	Crea páginas interactivas	Requiere productos Microsoft
PHP	Servidor	interprete	Sencillo de programar	Requiere servidor PHP
Servlets	Servidor	compilador	Portable	Requiere compilarse
XML	Cliente	interprete	Portable	Define estructuras solamente
XHTML	Cliente	interprete	Genera páginas HTML sin errores	Requiere depurar páginas HTML

**Tabla: Herramientas para desarrollo de aplicaciones en Internet**



Las herramientas mostradas en la tabla de arriba se pueden combinar para obtener la funcionalidad necesaria para el desarrollo de aplicaciones, estas se presentan en la siguiente tabla, dependiendo de la plataforma que se desee utilizar:

HERRAMIENTA	CGI	JAVA	Servlets	JSP / ASP	PHP
HTML	SI	SI	SI	SI	SI
JavaScript	opcional	opcional	opcional	opcional	opcional
CGI	SI				
Java		SI	SI		
Servlets		SI	SI		
JSP / ASP				SI	
PHP					SI
XML	opcional	opcional	opcional	opcional	opcional

Tabla: Herramientas requeridas para desarrollar aplicaciones para Internet

#### 4.1.2 Evaluación de XML

Las empresas y organizaciones utilizan la Web cada día como un nicho de negocios de primera necesidad para administrar una variedad de tareas que van desde el intercambio de información, hasta el comercio electrónico.

Ya que la cantidad de datos en la Web es abrumadora y se incrementa día con día, resulta todavía más difícil organizarlos eficientemente.

Incluso las computadoras más poderosas con ancho de banda amplio no son capaces de administrar durante mucho tiempo la cantidad de datos que fluyen a través de la Web cuando se pone a prueba una exhaustiva demanda. Existe la necesidad inmediata de una herramienta que administre una variedad de tareas intensivas con datos.

XML es quizás la tecnología más potente presentada desde Java. Ésta es ofrecida como la solución para una multitud de problemas en campos tan diversos como el diseño de páginas Web, desarrollo de bases de datos, estandarización de publicaciones y especialización en productos bancarios. Se puede decir que lo que Unicode está haciendo para el soporte de lenguaje universal, lo está haciendo XML para el intercambio de datos.

Para los desarrolladores de hoy, es importante saber como evoluciona la tecnología de software, XML puede ser aprovechada al máximo en una miríada de aplicaciones.

XML es un formato extremadamente conveniente para acceder dinámicamente a datos desde la Web a través de bases de datos así como clientes, sobrepasa las tecnologías de software anteriores que incrustaban datos en las páginas Web, los datos son mejor aprovechados, cercanos a sus formatos nativos y como consecuencia permiten mejores desempeños para obtener respuestas inmediatas.

Al aplicar una simple hoja de estilo o una aplicación Java Script, es posible crear el comportamiento y la apariencia de los datos como se requiera, aunado a esto, XML permite enviar la información al usuario más rápidamente y con menos carga sobre el servidor de aplicaciones.

### 4.1.3 Ventajas de XML para servidores Web

XML ofrece muchas ventajas para el desarrollador que está sirviendo datos en la aplicación Web. Quizá lo más importante sea la velocidad, ya que hace una envoltura muy ligera para contener los datos de manera que se lean y se escriban más rápidamente.

Muchos portales están preparados para una crear ambientes de información mediante HTML desde una base de datos empleando XML. Algunos de estos sitios ocupan tecnología CGI o productos adicionales como aplicaciones servidor para conectar sus páginas Web a bases de datos subyacentes. Sin embargo, esto obliga al servidor a realizar mucho trabajo ya que éste tiene que leer todos los códigos

HTML que se requieran, también debe buscar las solicitudes en la base de datos incrustadas en los registros, ejecutar dichas solicitudes a la base de datos, después regresar para insertar los resultados en código HTML antes de enviarlo al cliente con la presentación final del documento. Más aun, si el cliente desea realizar cambios en la presentación por ejemplo, obtener los resultados ordenados por fechas en lugar de por precios, el proceso entero se debe repetir. Esta es una utilización ineficaz de los recursos del servidor. Servir los datos en XML permite aprovechar mejor los recursos del servidor ya que el servidor que utiliza código XML hace una administración más rápida y eficaz de la información. Es más personalizado, y puede se intercambiar datos con otras bases de datos de una manera mas transparente.

#### **4.1.4 Rendimiento máximo**

La generación e interacción de datos mediante documentos XML representa un mejor aprovechamiento de los recursos porque los documentos tienen la flexibilidad de actuar como lo hacen bases de datos: almacenar, buscar y ordenar datos. Al ocupar un navegador web para generar la plantilla HTML y procesar los datos, se libera la carga de trabajo en el servidor de forma que éste podrá aceptar más solicitudes o atender a más peticiones. Cuando se utiliza XML, sólo se envían los datos al navegador. El cliente reúne entonces los datos en una vista y éstos son presentados al usuario. La mayor parte del trabajo está trasladado al cliente el cual permite servir muchas más solicitudes en menos tiempo utilizando el mismo ancho de banda de la red. Y, lo que consigue el cliente está mejor construido en comparación a las rudimentarias páginas HTML.

Es posible utilizar Java o Java Script para ordenar o buscar en tiempo real una serie de resultados.

### 4.1.5 Personalizable

La manera más común para integrar XML es a través de formularios y hojas de estilo. Los navegadores muestran a los usuarios un formulario estándar HTML. El usuario llena y envía el formulario.

Se recibe la información y el servidor responde con una respuesta en XML que incluye los datos y los detalles sobre dónde se puede encontrar una hoja de estilo adecuada a dicha información. En este punto el trabajo del servidor está hecho, y está preparado para procesar la próxima solicitud desde cualquier otro cliente. El navegador entonces utiliza una hoja de estilo que reconoce cómo dar formato a los datos para presentarlos al usuario.

El servidor puede también enviar al navegador un programa JavaScript que indica al navegador cómo hacer los datos dinámicos, de manera que el usuario pueda ordenarlos o buscarlos en tiempo real. El suministro de una hoja de estilo mostrará al usuario una vista diferente de los datos. Y todo esto se puede hacer sin sobrecargar al servidor de nuevo con más solicitudes.

## 4.2 El nuevo estándar en el intercambio de datos

XML no es sólo para la Web. Tiene también otros usos. Es un formato ideal para compartir e intercambiar información con terceras personas, está puede estar en un RDBMS, pero quizás los clientes utilicen otro, esto no es un problema; ya que se pueden mover datos de un sistema a otro con "exportar" en XML, utilizando XSL para transformarlos en un formato estándar XML, y después importarlos en la nueva base o sistema. Y podemos decir que esta es la clave para la expansión del B2B (**Business to Business**).

## 4.3 ¿Por qué utilizar XML?

XML ofrece muchas ventajas para el desarrollo de aplicaciones que requieren enviar o recibir datos; Primero, XML es un estándar que puede ser implementado y utilizado para lograr cualquier propósito dentro del mundo de las aplicaciones. Con este estándar, no importa la aplicación que se este desarrollando, ya que gracias a él, una parte del trabajo ya está hecha y puede ser fácilmente incorporada en cualquier proyecto con un bajo costo y un mínimo esfuerzo.

XML es más rápido en comparación con otras soluciones patentadas. Los sitios Web que utilizan JSP, u otros productos de índole middleware<sup>1</sup> para realizar transacciones entre el usuario y la base de datos, necesitan que el servidor realice mucho trabajo de lectura de los documentos HTML, buscar las peticiones a la base de datos incrustadas en los archivos, ejecutar éstas y luego insertar los resultados en el HTML antes de enviarlos al cliente. Si el cliente quiere cambiar algo, no importa que sea pequeño, debe volver al servidor que repetirá todo el proceso.

Entonces XML trabaja mucho mejor porque todo lo que el servidor tiene que hacer es preprocesar la información representada como documentos XML y hacer un direccionamiento al cliente para enviárselos a éste como resultados de una simple solicitud en forma de HTML, es decir que todo el trabajo se realiza en el

---

<sup>1</sup> Ver Glosario de Términos página 183

servidor dando formato a los datos y llevando a cabo cualquier manipulación que se requiera.

En conclusión la finalidad de nuestra propuesta es la de poder interactuar con todo tipo de clientes dando un completo soporte gracias a XML, además de permitir la publicación de obras digitales empleado como base documentos XML en la Web.

### 4.4 ¿Porqué JAVA?

**¿Cuáles son las razones por las que hemos elegido Java para el desarrollo de la aplicación segura para Internet?**

La plataforma Java es fundamentalmente una nueva forma de desarrollar tecnología de sistemas, basada en el poder de las redes y en la idea de que el mismo software debería ejecutarse en diferentes tipos de computadoras, dispositivos personales y otros equipos.

Con Java, se puede utilizar la misma aplicación desde cualquier equipo, ya sea una PC, dispositivos o asistentes personales (PDA) <sup>2</sup>, computadoras Macintosh, una terminal o cualquier tipo de tecnología nueva como los video-teléfonos de Internet.

Acoplada con el poder de las redes, la plataforma Java está ayudando a los usuarios de computadoras a hacer cosas que anteriormente eran inimaginables. Por ejemplo, los usuarios pueden de forma segura acceder información personal y aplicaciones cuando están lejos de la oficina usando cualquier computadora que esté conectada a Internet, así que será capaz de acceder aplicaciones desde un teléfono móvil basado en la plataforma Java o incluso usar Tarjetas Inteligentes para obtener acceso desde un simple cajero automático.

---

<sup>2</sup> Ver glosario de términos página 183

### ¿Porqué tecnología Java?

Las redes requieren de software que sea portable, modular y seguro, es decir, todas las áreas donde este lenguaje en este momento, tiene ventajas de sobra porque fue diseñado para el uso en redes desde su concepción.

La tecnología Java elimina muchos de los problemas asociados con la instalación y la ejecución de las aplicaciones. Eso es porque generalmente el usuario no tiene que configurar, cargar, o instalar nada. Las actualizaciones son automáticas, haciendo la instalación y configuración obsoleta.

Es toda una nueva forma de pensar en el desarrollo de sistemas de computadoras. Además, desde el inicio, la plataforma Java fue diseñada para ejecutar programas de forma segura en las redes, lo cual significa que se integra de forma confiable con los sistemas existentes en la red.

---

# Capítulo 5

## ARQUITECTURA Y DISEÑO

### 5.1 Arquitecturas cliente-servidor

Internet es un sistema en el que participan dos partes: por un lado está la computadora desde la cual accedemos a la red, (*máquina o servidor local*), la cual trabaja con una aplicación conocida como: *programa cliente*, y en el otro extremo se encuentra la computadora que emite la información o servicios (*servidor remoto*), que emite y facilita la información que necesitamos gracias a un *programa servidor*.

El programa cliente administra la comunicación con el servidor y ofrece las herramientas necesarias para poder trabajar con él.

El programa servidor se encarga de transmitir la información en la forma más adecuada para el usuario o usuarios, ya que éste admite múltiples accesos simultáneos.

Los programas cliente-servidor pueden ser muy variados y funcionar sobre sistemas operativos diversos como: UNIX, Linux, Windows, OS/2, IRIX, AIX, por mencionar algunos, pero éstos programas deben utilizar el protocolo de comunicaciones TCP/IP.

El protocolo TCP controla la transmisión de archivos electrónicos; divide la información en porciones apropiadas numeradas para que puedan volver a unirse en su totalidad, o si hubiera algún error en la transmisión lo permitiría identificar al instante; y la labor de recomposición también la realiza el propio protocolo TCP.

Por su lado, el protocolo IP asigna a cada paquete de información electrónica que fluye por la red la dirección apropiada en Internet: dirección IP. Dirección que



identifica e individualiza a todos y cada una de las computadoras conectadas a Internet, que reciben el nombre de host o anfitrión. Esta dirección o número IP se compone por cuatro grupos sucesivos de cifras, con un valor comprendido entre 0 y 255, que están separados por puntos. Por ejemplo una dirección IP puede aparecer así: 188.132.56.23, aunque también existe una forma alternativa de tipo alfabética llamada DNS.

El modelo cliente-servidor es un conjunto de tecnologías que compiten por la creación de aplicaciones para este modelo, han reinventado la computación cliente-servidor, no sólo en cuanto a bases de datos, sino también con los objetos distribuidos y el web con herramientas como Java.

### **Servidores de Bases de Datos**

El modelo cliente-servidor es el modelo base utilizado en la mayoría de las redes de computadoras actualmente, el término *servidor* se aplica a cualquier programa que ofrece un servicio y que puede tener acceso a través de la red, aceptan peticiones recibidas a través de la red, realizan una determinada labor y regresa el resultado al que generó la petición.

También pueden realizar tareas simples o bastante complejas, si el principal objetivo de una máquina es el de soportar un programa de este tipo en particular, entonces el término "servidor" puede ser aplicado también a dicha máquina.

El servidor comienza su ejecución antes del inicio de cualquier interacción y continúa aceptando peticiones y enviando respuestas indefinidamente. Un cliente es cualquier programa que efectúa una petición y espera una respuesta, generalmente termina después de usar un servidor un número finito de veces.

En una arquitectura de este tipo centrada en base de datos, por lo general una aplicación cliente solicita datos y servicios relacionados con datos (como

clasificación y filtración) a un servidor de bases de datos, también llamado mecanismo de SQL, responde a las solicitudes del cliente y ofrece acceso protegido a datos compartidos. Con una sola instrucción SQL, una aplicación puede recuperar y modificar un conjunto de registros de la Bases de Datos del servidor.

**Características:**

- ❖ Administra el control y ejecución de comandos SQL.
- ❖ Proporciona los modelos lógico y físico de los datos.
- ❖ Permite que múltiples aplicaciones accedan a la misma bases de datos al mismo tiempo.
- ❖ Administra los aspectos de recuperación, concurrencia, seguridad y consistencia de una base de datos.

**Arquitectura de los Servidores:**

Existen tres tipos de arquitecturas empleadas para tratar bases de datos remotas y sus servidores:

□ ***Arquitectura de proceso por cliente:***

A cada cliente de la Base de Datos se le otorga su propio espacio de direccionamiento del proceso. Esto protege a los usuarios entre sí y al mismo tiempo, protege al administrador (de la Base de Datos), de los usuarios.

Ejemplos de servidores que implementan esta arquitectura: Sybase, DB2, Informix y Oracle 9i por mencionar algunos.

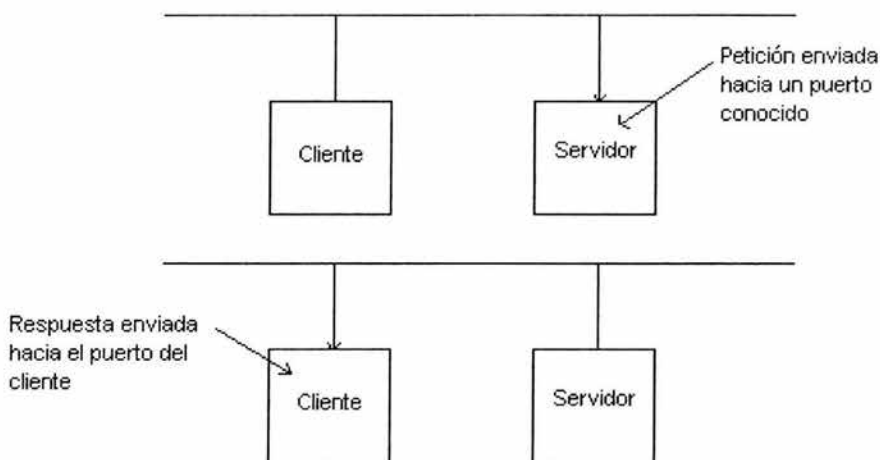
### □ *Arquitectura multihilos:*

Ejecutan todas las conexiones del usuario, las aplicaciones y la base de datos en el mismo espacio de direccionamiento.

### □ *Arquitecturas híbridas:*

Poseen tres componentes:

- 1) Un programa de red multihilos que participa en la conexión inicial asignando el cliente a un proceso planificador.
- 2) Una base compartida de planificadores que colocan los mensajes en una cola interna de solicitudes / respuestas, que después sacan las respuestas de ésta para enviarla al cliente.
- 3) Una base compartida de procesos del servidor que extraen el trabajo de la cola, lo ejecutan y regresan las respuestas a la misma cola.

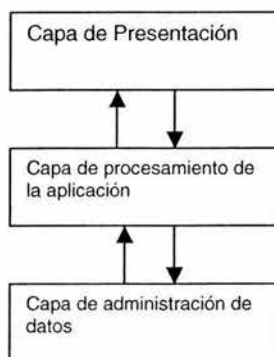


**Modelo Cliente/Servidor**

En una arquitectura cliente-servidor, una aplicación se modela como un conjunto de servicios proporcionados por los servidores y un conjunto de clientes que utilizan estos servicios. Los clientes deben saber qué servidores están disponibles, pero por lo general no conocen la existencia de otros clientes. Los clientes y servidores son procesos diferentes.

No existe una relación uno a uno entre los procesos y los procesadores del sistema. Éstas ejecutan los procesos cliente y servidor. En general, cuando se habla de clientes y servidores se refiere a aquellos procesos lógicos más que a las computadoras físicas sobre las cuales estos procesos se ejecutan.

El diseño de sistemas cliente-servidor debe reflejar la estructura lógica de la aplicación que se desarrolla. Una forma de observar una aplicación se ilustra en la siguiente figura, que muestra una aplicación estructurada en tres capas. La capa de presentación tiene que ver con la presentación de la información al usuario y con la interacción del usuario. La capa de procesamiento de la aplicación se refiere a la implantación de la lógica de la aplicación y la capa de administración de datos se refiere a todas las operaciones de la base de datos. En los sistemas centralizados no se requiere que estas capas estén claramente separadas. Sin embargo, cuando se diseña un sistema distribuido, debe hacerse una clara distinción entre ellas puesto que de esta forma es posible distribuir cada capa en una computadora diferente.



### Capas de la aplicación

A la arquitectura cliente-servidor más simple se le denomina arquitectura cliente-servidor de dos capas, en donde una aplicación se organiza como un servidor (o diversos servidores idénticos) y como un conjunto de clientes.

Como se ilustró en la figura anterior, las arquitecturas cliente-servidor de dos capas presentan dos formas:

**Modelo de cliente delgado.** En este modelo, todo el procesamiento de la aplicación y la administración de datos se realiza en el servidor. El cliente es simplemente responsable de ejecutar el software.

**Modelo de cliente grueso.** En este modelo, el servidor sólo es responsable de la administración de datos. El software en el cliente implementa la lógica de la aplicación y las interacciones con el usuario del sistema.

Una arquitectura de dos capas de cliente delgado es el enfoque más simple que se utiliza cuando los sistemas heredados que son centralizados evolucionan a una arquitectura cliente-servidor. La interfaz de usuario para estos sistemas se migra a las computadoras y la aplicación de la misma actúa como un servidor y maneja todo el procesamiento de la aplicación y la administración de datos. Un modelo de cliente delgado también puede ser implementado cuando los clientes son dispositivos de red sencillos en lugar de PC o estaciones de trabajo.

Los dispositivos de red ejecutan un navegador Web y la interfaz de usuario es implementada a través de ese sistema.

Una gran desventaja del modelo de cliente delgado es que coloca una gran carga de procesamiento tanto en el servidor como en la red. El servidor es responsable de todos los cálculos y esto implica la generación de bastante tráfico en la red entre el cliente y el servidor. Existe bastante poder de procesamiento disponible en las PC modernas y este no es del todo utilizado en el enfoque de cliente delgado.

En contraste, el modelo de cliente grueso utiliza este poder de procesamiento disponible y distribuye tanto procesamiento lógico de la aplicación como la presentación al cliente. Esencialmente, el servidor es un servidor de transacciones que administra todas las transacciones de la base de datos. Un ejemplo conocido de este tipo de arquitectura son los sistemas bancarios ATM o Modo de Transferencia Asíncrona<sup>3</sup>, donde el ATM es el cliente y el servidor es una “mainframe” que ejecuta la base de datos de la cuenta del cliente.

En esta red ATM, éstos no se conectan directamente a la base de datos del cliente sino al monitor de tele procesamiento (TP). Este monitor es un sistema “middleware” que organiza las comunicaciones con los clientes remotos y coloca en serie las transacciones de los clientes para ser procesadas por la base de datos. Utilizar transacciones en serie significa que el sistema puede recuperarse de fallas sin corromper los datos del sistema.

Aunque el modelo del cliente grueso distribuye el procesamiento de forma más efectiva que el cliente delgado, la administración del sistema es más compleja. La funcionalidad de la aplicación se distribuye en muchas computadoras diferentes. Cuando la aplicación del software se tiene que cambiar, esto implica la reinstalación en cada computadora cliente. Esto tiene un costo mayor si existen cientos de clientes en el sistema.

---

<sup>3</sup> Ver página 179

La llegada de Java y los applets permitieron el desarrollo de un modelo cliente-servidor que esta en algún lugar entre el modelo de cliente delgado y de cliente grueso. Una parte del software de procesamiento de la aplicación se descarga en el cliente como applets de Java, de esta manera se libera la carga al servidor. La interfaz de usuario se construye utilizando un navegador Web que puede ejecutar las distintas aplicaciones de Java. Sin embargo, los navegadores Web de diferentes fabricantes e incluso las versiones no siempre se despliegan en la misma forma.

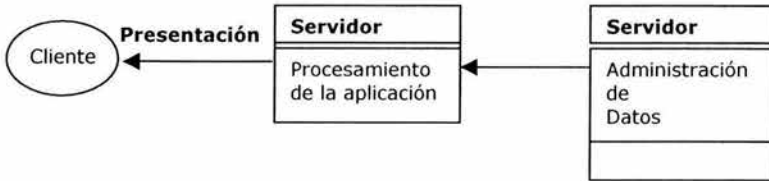
El problema fundamental con un enfoque cliente-servidor de dos capas es que las tres capas lógicas: presentación, procesamiento de la aplicación y administración de datos, deben asociarse a dos sistemas de cómputo. Pueden existir problemas ya sean de escalabilidad y desempeño si se elige el modelo de cliente delgado o problemas de administración del sistema si se utiliza el modelo de cliente grueso. Para evitar estos problemas, un enfoque alternativo es utilizar una arquitectura cliente-servidor de tres capas.

En esta arquitectura, la presentación, el procesamiento de la aplicación y la administración de datos son procesos separados lógicamente. Por otra parte, una arquitectura de software cliente-servidor de tres capas no implica necesariamente que existan tres sistemas de cómputo conectados a la red. Un solo servidor puede ejecutar tres sistemas de cómputo conectados a la red. Sin embargo, si el tiempo de respuesta es mínimo, es importante considerar en separar el procesamiento de la aplicación y la administración de datos para ejecutarlos en procesadores separados.

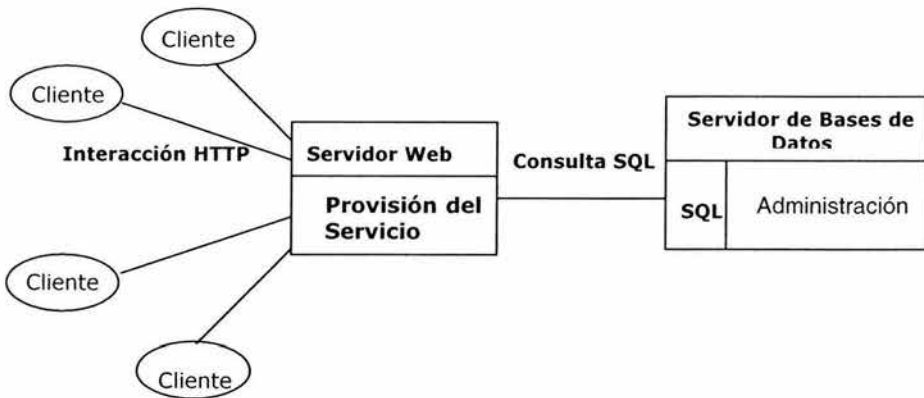
Un sistema bancario en Internet es un ejemplo de tipo de sistemas que se pueden implementar utilizando una arquitectura cliente-servidor de tres capas. La base de datos de clientes del banco suministra los servicios de administración de datos, un servidor Web provee los servicios de aplicación como los recursos para transferir efectivo, generar estados de cuenta, realizar pagos, etc, y el cliente es la computadora del usuario con un navegador Web.

En nuestro caso, el prototipo para un Sistema de Administración de Libros

digitales utiliza una arquitectura cliente-servidor de distribución de dos capas.



**Figura:** Arquitectura Cliente-servidor de tres capas



**Figura:** Arquitectura de distribución de un sistema bancario en Internet

La utilización de una arquitectura de tres capas en este caso permite transferir la información entre el servidor Web y el servidor de la base de datos a optimizarse. La comunicación entre estos sistemas no tiene por qué basarse en estándares de Internet pero puede utilizar protocolos de comunicación de bajo nivel muy rápidos. Para manejar la recuperación de la información de la base de datos se utiliza un “middleware” eficiente que permita consultas a la base de datos en SQL (Lenguaje de Consultas Estructurado).

En algunos casos, es apropiado ampliar el modelo cliente-servidor de tres capas a una variante de múltiples capas en el cual se agregan servidores al



sistema. Los sistemas de múltiples capas se utilizan cuando las aplicaciones necesitan acceder y utilizar datos de diferentes bases de datos. En este caso, un servidor de integración se ubica entre el servidor de aplicación y los servidores de bases de datos que se acceden. El servidor de integración recolecta los datos distribuidos y los presenta a la aplicación como si éstos estuvieran disponibles en una sola base de datos.

- Las arquitecturas cliente-servidor de tres capas y las variantes de múltiples capas distribuyen el procesamiento de la aplicación sobre varios servidores son inherentemente más escalables que las arquitecturas de dos niveles. Una gran ventaja de la utilización de estos modelos radica en que el tráfico de la red se reduce en contraste con las arquitecturas de dos capas de cliente delgado. El procesamiento en algunos casos, se distribuye entre la lógica de la aplicación y los servidores de administración de datos, permitiendo así una respuesta más rápida a las peticiones de los clientes.

Los diseñadores de arquitecturas cliente-servidor deben tomar en cuenta una serie de factores para poder definir la adecuada a sus requerimientos.

## 5.2 Diseño orientado a objetos

El diseño orientado a objetos es una estrategia de diseño en el cual los diseñadores del sistema piensan en términos de “cosas” en lugar de operaciones o funciones. El sistema se compone de objetos que interactúan entre ellos y que mantienen su propio estado local y suministran operaciones de esa información del estado. Ocultan información de la representación del estado y después limitan su acceso. Un proceso de diseño orientado a objetos comprende el diseño de clases de objetos y las relaciones entre estas clases. Cuando el diseño se implementa como un programa ejecutable, los objetos requeridos se crean de manera dinámica utilizando las definiciones de las clases.

El diseño orientado a objetos es parte del *desarrollo orientado a objetos* en el que se utiliza una estrategia definida orientada a objetos:

- **El análisis orientado a objetos** comprende el desarrollo de un modelo orientado a objetos del dominio de aplicación. Los objetos identificados reflejan las entidades y las operaciones que se asocian con el problema a resolver.
- **El diseño orientado a objetos** comprende el desarrollo de un sistema de software para implementar los requerimientos identificados.
- **La programación orientada a objetos** se refiere a llevar a cabo el diseño de software utilizando un lenguaje de programación de este tipo. Java es un lenguaje de este tipo que permite la implementación directa de los objetos y suministra recursos para definir las clases.

Los sistemas orientados a objetos deben ser permiten un fácil mantenimiento debido a que éstos son entidades independientes. Estos objetos se deben entender y modificar de manera independiente.

En un sistema orientado a objetos, la implementación de un objeto o la agregación de servicios no afecta a las demás entidades del sistema. Por otro lado, puesto que los objetos se encuentran asociados a cosas, existe una correspondencia clara entre las entidades del mundo real (como los componentes de hardware) y sus objetos de control en el sistema.

Los objetos son componentes potencialmente reutilizables debido a que son encapsulamientos independientes del estado y las operaciones. Los diseños se pueden desarrollar utilizando objetos creados en diseños anteriores. Esta ventaja en el diseño reduce los costos, programación y validación. También conduce a la utilización de objetos estándar (por lo que se mejora la comprensión del diseño) y reduce los riesgos implicados en el desarrollo de software. Sin embargo, algunas veces la reutilización se implementa aún con mayor efectividad si se utilizan colecciones de objetos (componentes o marcos de trabajo) en lugar de objetos individuales.

Se han propuesto diversos métodos de diseño orientado a objetos (Coad y Yourdon, 1990; Robinson, 1992; Jacobson et al., 1993; Booch, 1994; Graham, 1994). Se ha definido una unificación de las notaciones utilizadas en estos

métodos (UML, *Undefined Method Language*) junto con el proceso de diseño asociado (Rumbaugh et al., 1999). Por tanto se considera de gran trascendencia utilizar esta notación.

### 5.3 Objetos y clases de objetos

En la actualidad los términos “objeto” y “orientado a objetos” se utilizan ampliamente. Se aplican a diversos tipos de entidades, métodos de diseño, sistemas y lenguajes de programación. Por lo general se acepta que un objeto es un encapsulamiento de la información.

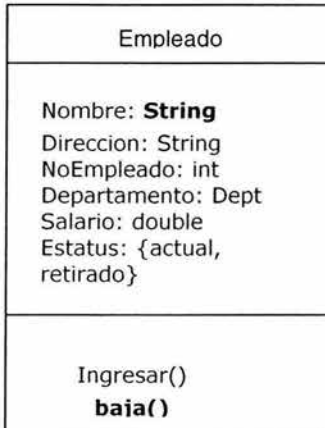
Un objeto es una entidad que tiene un estado y un conjunto de operaciones definidas que operan sobre ese estado. El estado se representa como un conjunto de atributos del mismo. Las operaciones que son asociadas con él, brindan servicios a otras entidades ó clientes que los solicitan cuando se requiere realizar alguna operación o función.

Los objetos se crean conforme a una definición de clases. Dicha definición sirve como plantilla para crear objetos. Ésta incluye las declaraciones de todos los atributos y operaciones asociados con un objeto de esa clase.

La notación que se utiliza para definir las clases es la de UML. Una clase de objetos se representa como un rectángulo al cual se le han asignado un nombre y dos secciones. Los atributos del objeto se listan en la sección superior. Las operaciones que se encuentran asociadas con los objetos se listan en la sección inferior. En la notación UML, el término “operación” es la especificación de alguna acción; el término “método” se utiliza para referirse a la implementación de la operación.

La clase Empleado define varios de los atributos que contienen información de los usuarios, incluyendo su nombre y dirección, teléfono, antigüedad, número de empleado, en fin, una serie de datos que definen a dicho empleado, así como las

operaciones que se lleguen a efectuar sobre él. Las operaciones asociadas con el objeto son ingresar() (que se llama cuando un usuario es agregado), baja() (que se llama cuando el usuario se da de baja del sistema), cambiar() (utilizado para hacer actualizaciones en los datos del usuario).



En algunos sistemas distribuidos, las comunicaciones de los objetos se implementan directamente como mensajes de texto que los objetos intercambian. El objeto receptor analiza el mensaje, identifica el servicio y los datos asociados, y lleva a cabo el servicio solicitado. Sin embargo, cuando los objetos coexisten en el mismo programa, las llamadas al método se implementan de la misma forma que una llamada a un procedimiento o función en un lenguaje como C.

Cuando las peticiones del servicio se implementan de esta forma, la comunicación entre los objetos es síncrona. Esto es, el objeto invocador espera a que la petición del servicio se complete. No obstante, si los objetos se implementan como procesos concurrentes o cadenas, la comunicación es asíncrona. El objeto invocador continúa en operación mientras que el servicio solicitado se ejecuta.

Las clases de objetos se pueden arreglar en una generalización o jerarquía de herencia que muestra las relaciones entre las clases de objetos generales y más específicas. La clase de objetos más específica es completamente consistente con la clase de objetos general, pero incluye información adicional. Esto se representa en UML mediante una flecha que apunta hacia la clase madre. La clase hija hereda los atributos y operaciones de la clase madre.

Actualmente se trabaja en el API de Seguridad de Java, que tiene como meta el proporcionar servicios de validación y cifrado para las aplicaciones Java. Debido a la extrema dependencia de la seguridad a lo que se decida en este API, ha surgido una gran polémica en torno a qué medidas de seguridad adicionales deberán incorporarse.

Un rasgo típico de que ya gozan los applets, es la capacidad de ser firmadas por alguna autoridad de certificación. De esta manera, al presentar estas aplicaciones, una firma digital, el usuario puede decidir si la ésta es de confianza o no. Las aplicaciones seguras tendrán el privilegio de acceder de forma controlada a los recursos, mientras que al resto de las demás se les negará dicho acceso.

---

## Capítulo 6

# ANÁLISIS Y DISEÑO DE UN SISTEMA DE PUBLICACIÓN EN LÍNEA

### 6.1 Manejo de datos aplicados a un modelo real

El manejo de datos a través del WWW cuenta con una amplia gama de aplicaciones que van desde un simple nicho de comercio electrónico, aplicaciones de transacciones empresariales robustas, hasta votaciones electrónicas con niveles sofisticados de seguridad y de confidencialidad.

El planteamiento analizado en esta ocasión, nació de la necesidad de elaborar un sistema de publicación de obras digitales que en términos prácticos conocemos como: Biblioteca Digital. Anteriormente ya se han mencionado algunos de los beneficios proporcionados por las bibliotecas digitales, pero debido a una reestructuración en los requerimientos del sistema, actualmente definimos nuestra aplicación como un portal robusto, con la capacidad de almacenamiento y búsqueda que ofrece una Biblioteca Digital, además de tener la posibilidad de actuar como una herramienta de construcción de documentos digitales al vuelo. Por tanto, al sistema analizado y desarrollado se le ha atribuido el nombre de **Libro digital** por sus versátiles características.

Nuestro interés y reto ha sido la definición de un sistema específico de información que permita la creación, edición y publicación de distintas obras digitales, apegados a algunos de los modelos y esquemas particulares para la publicación digital.

Los modelos ocupados fueron creados por la *Dublín Core*<sup>4</sup> y se tomaron en gran medida como referencias de cabecera para definir las entidades y sus relaciones, como requerimiento para el levantamiento de un modelo de bases de datos que representará,

---

<sup>4</sup> Mas detalles acerca de la **Dublín CORE** ver glosario de términos página: 181

en el ámbito contextual, las particularidades específicas de la ambiciosa aplicación. De igual medida, el uso de este tipo de modelos sirvió para el diseño e implementación de las etiquetas de marcado extendido definidas con el lenguaje XML para la creación de estructuras de meta datos, así como tipos de dato de documentos (DTD). Con éstos, se confeccionaron los elementos más importantes para la construcción de libros al vuelo y la presentación de la información de éstos en un formato estándar. Todos estos esfuerzos cuentan con la posibilidad de extender el sistema a posteriori pensando en aplicaciones futuras.

Es decir en este momento se modeló el sistema para la implementación de publicaciones en línea de libros digitales, pero a su vez en un futuro próximo podría ser aplicada la misma estructura al desarrollo de la implementación de una aplicación de publicaciones de revistas digitales por ejemplo. Esto gracias a su carácter modular y la programación orientada a objetos. Existe también la posibilidad de desarrollar e implementar módulos para la comunicación o descarga para dispositivos móviles como lo son las PDA (Asistentes Digitales Personales) todo esto gracias a las características homogéneas y portables que en el sistema se han implementado.

### **6.2 Marco de referencia tecnológica y derechos de autor**

La aparición de Internet y la ingeniería de software en la vida común del ser humano, han modificado sustancialmente el quehacer social en distintos ámbitos en tan solo un par de décadas. La explosión tecnológica en comunicaciones y sistemas han abarrotado el mercado de un sin número de aplicaciones y dispositivos que procuran simplificar la vida diaria.

No es de extrañar que tanto la educación como la difusión cultural se han beneficiado de las tecnologías de la información y de las comunicaciones. Cabe señalar que ha sido tanta la influencia que hoy en día, dichas tecnologías son decisivas para su desarrollo debido a que elevan la rentabilidad social y educativa.

Dado que la UNAM se encuentra inmersa en esta realidad, la sistematización de los procesos y el alcance de estos a través de Internet nos compromete a brindar, cada día más y, mayores beneficios a los usuarios finales. Sin embargo, algunas de las barreras que se pueden presentar si no se tiene el debido cuidado en el análisis, son los efectos que puedan aportar las aplicaciones mismas en el impacto social.

De tal suerte que si bien una aplicación puede simplificar muchos de los procedimientos de la vida cotidiana, también están sujetos a que estos procesos no afecten la autenticidad o el derecho legal de los productos finales o de los servicios ofrecidos.

En nuestro caso y partiendo del análisis para la edición de obras digitales, detectamos que se requiere un pleno conocimiento legal enfocado a los derechos de autor. Los detalles de ésta labor implicarían fácilmente la elaboración amplia de otro trabajo de tesis pero en nuestro contexto, y bajo una lógica definida de implementación, mencionaremos a modo de resumen los siguientes supuestos:

- ❖ Consideramos necesario el contemplar la creación de un comité o arbitro que se encargue de regular las actividades y los impactos de los derechos de autor para proteger la autenticidad y originalidad de los documentos que se publiquen.
- ❖ Establecer o delimitar el alcance de la aplicación de publicación, para mantener un marco interno que controle y regule los actos que los usuarios tienen derecho a ejercer.
- ❖ Apegarse a las normas nacionales e internacionales de digitalización de la información, así como a la propiedad de las obras digitales.



- ❖ Obtener los convenios necesarios con las distintas casas de publicación de obras, ó por otro lado, el permiso de los autores o titulares propietarios de la publicación.
- ❖ Definir los apartados necesarios para estandarizar formatos de digitalización y como descarga; así como también las definiciones para los derechos de autor de los libros digitales.

**Definición Legal del Derecho de Autor.** Es el reconocimiento que hace el Estado a favor de todo creador de "Obras Literarias y Artísticas" previstas en el artículo 13 de la **Ley Federal del Derecho de Autor (LFDA)**, en virtud del cual otorga su protección para que el autor goce de prerrogativas y privilegios exclusivos de carácter personal y patrimonial. Los primeros integran el llamado derecho moral y los segundos el patrimonial.

Dentro del derecho de autor se tienen las siguientes funciones:

- ❖ Protección a los autores.
- ❖ Protección a las obras literarias y artísticas.
- ❖ Protección a artistas, intérpretes, ejecutantes, editores, productores de fonogramas, videogramas y organismos de radiodifusión.
- ❖ Protección a las culturas populares.
- ❖ Protección a títulos de publicaciones y difusiones periódicas, personajes humanos de caracterización o ficticios o simbólicos; personas o grupos dedicados a actividades artísticas y promociones publicitarias.

### 6.3 Requerimientos del Sistema

Tomando como modelo la construcción de una biblioteca digital para la Dirección General de Servicios de Cómputo Académico (DGSCA); se realizó el levantamiento de información que nos fue proporcionada por algunos usuarios y funcionarios de la biblioteca de ésta misma dependencia. La toma de información se elaboró a partir de pláticas y cuestionarios con los usuarios y administradores de dicha biblioteca, que no hemos anexado por razones de confidencialidad, pero que sirvieron para dimensionar el planteamiento de la problemática, lo cual se traduce en la siguiente descripción general del sistema elaborado.

En el momento en que se hicieron los análisis de requerimientos para el prototipo, la DGSCA estaba trabajando en la implementación de una biblioteca digital, por tanto nuestro proyecto se define a nivel de prototipo de implementación en el que se plantean un modelado de datos orientado a componentes que interactúan con XML para la creación y consulta de obras digitales.

Los siguientes puntos definen las características y singularidades del sistema, haciendo la aclaración de que únicamente se plantean la definición, el análisis y diseño del prototipo, pero de antemano cabe señalar que éste capítulo no es exactamente un documento de requerimientos del Sistema, ya que éstos llevan una estructura específica en el momento de la liberación del software y este tipo de documentos son mucho más extensos y minuciosos.

Aún así, es necesario mencionar los elementos medulares que sirvieron durante el análisis y diseño del sistema por lo que consideramos que es de gran utilidad agregar las descripciones más detalladas que sirvieron como base para definir la especificación general de la aplicación en términos de ingeniería de software.

Únicamente se tomaron los puntos de interés para plasmarlos en este trabajo de tesis como base para la definición del prototipo Libro Digital.

## 6.4 Descripción General del Sistema

Se tuvo el interés por plantear un sistema que utilizara plantillas específicamente definidas para el ingreso de información con el objetivo de que ésta se fuera capturando de manera secuencial.

Estas plantillas no son más que un cúmulo de interfaces administradas por una aplicación Web que se encuentra montada en una plataforma de servidor Web. En este caso la aplicación es ejecutada por el servidor de aplicaciones Jakarta Tomcat <sup>5</sup>. Es importante decir que el servidor de aplicaciones se encuentra instalado bajo una plataforma Linux Red Hat 7.3. y cabe mencionar que la aplicación se encuentra desarrollada bajo el esquema **Ciente – Servidor**. Nuestro sistema se encuentra soportado por una base de datos que es la encargada de almacenar y administrar la información que se va a ingresar en él.

La base de datos se planteó bajo un modelo relacional (**RDBMS**) y por tanto se tiene a detalle el diseño de las entidades así como de sus relaciones. Siguiendo con los cánones del diseño de bases de datos, también se generó un documento que aporta cada detalle de los atributos o campos requeridos para el pleno control de datos que conocemos como diccionario de datos.

En base a un diseño orientado a objetos, se definieron clases de objetos que son las herramientas ocupadas para la aplicación de negocios y la generación de interfaces dinámicas, por tanto debemos mencionar que por un lado las clases de objetos definen a la capa de negocio mientras las páginas JSP, HTML junto con las hojas de estilo, definen a la capa de presentación de la aplicación. Finalmente, la capa de administración de datos se encuentra controlada por la misma base de datos a la que se robusteció con seguridad a nivel de esta misma.

---

<sup>5</sup> Definición del proyecto Jakarta Tomcat, ver página: 182, o visitar la página: <http://jakarta.apache.org/>

Dentro de la capa de negocio, se definieron implementaciones de validación a nivel de presentación ocupando objetos de Java así como de Java Script. Y a nivel base de datos ocupando procedimientos almacenados, por lo que podemos afirmar que resultó ser una aplicación quizá no infalible pero con un nivel de fiabilidad y certeza bastante amplio.

### **El sistema se compone por cinco módulos principales:**

- 1) Módulo de consulta y navegación.** Este módulo permite la elaboración de consultas al acervo almacenado y cuenta con detalles finos para la generación de búsquedas avanzadas dentro de los documentos. La navegación se realiza mediante consultas a la base de datos y con la ayuda de las hojas de estilo de XML y sus transformaciones es que se presenta el libro en una construcción al vuelo de las páginas consultadas.
- 2) Módulo de Edición y actualización de obras digitales.** Este módulo se encarga de la construcción y edición de la obra digital. Es quizá el módulo más importante y sofisticado ya que en éste se define la estructura completa de la obra así como la alimentación por partes de la misma.

El usuario puede construir su obra ocupando las interfaces secuenciales, o una vez que ha definido el índice que es la generadora de la estructura del libro, puede ocupar un proceso batch para subir su información al servidor. En el caso de ocupar el proceso batch, se requiere que todos los datos que suban se validen contra la estructura que él mismo definió para evitar inconsistencias o redundancia de información. La validación del documento se realiza en el lado del servidor mediante una aplicación de análisis de etiquetas de XML.

Para lograr tal efecto, cabe mencionar que en el servidor físico únicamente se escriben las imágenes requeridas en un directorio escrito al

momento de crear la obra, mientras todo el texto plano de la obra sin excepción, es almacenado en la base de datos. De esta suerte tenemos que el texto de la obra completa se almacena lógicamente, las imágenes en el servidor, y las referencias de estas imágenes también son guardadas en la base. La funcionalidad de XML es ocupada como herramienta para la definición de datos y construcción de estructuras definidas por documentos de tipo de dato DTD. Por tanto no importa como se guarde la información, la recuperación de ésta es de manera inmediata ya que se hacen consultas y de esta forma se recupera la información para la construcción al vuelo y la presentación de ésta mediante aplicaciones de hojas de estilo que transforman el XML bruto a páginas HTML.

- 3) **Módulo de alta de usuarios de consulta así como de edición.** Este módulo se encarga de dar de alta a las personas interesadas en consultar o editar sus obras digitales. Cabe mencionar que será cuestión de la definición de una reglas de negocio posterior si esta aplicación es utilizada en un plano público o institucional.
- 4) **Módulo de descargas de libros.** Los libros pueden ser descargados de acuerdo a los convenios que se tengan por derechos de autor y uso de las obras digitales de manera total o parcial. También en base a estos convenios, se ha pensado en la habilitación o bloqueo de teclas y de eventos para evitar o permitir el copiado de la información de las páginas HTML.
- 5) **Módulo de administración.** Este módulo no esta completamente desarrollado en la aplicación pero sí contemplado en el diseño. En este módulo se cuenta con un usuario que será capaz de administrar todo el sitio, asignar y revocar cuentas de usuario, eliminación, edición o alta de nuevas obras digitales, así como monitoreo y control de bitácoras de acceso para libros y usuarios.

## 6.5 Análisis del proceso mediante casos de uso

En las siguientes páginas se muestra el análisis del sistema definido por casos de uso que fueron implementados ocupando un modelo orientado a objetos. A continuación se muestran los casos implementados:

Caso de uso	Autenticación				
	Id. del CU	1	Prioridad	Alta / Media / Baja	Estado
<b>Actores participantes</b>	<b>Usuario de Internet</b> – Cualquier usuario del sitio: usuario de administración, consulta o de edición.				
<b>Breve descripción</b>	Permite el acceso de los usuarios al sistema mediante la verificación de la información de login y contraseña.				
<b>Pre-condiciones</b>	<p><i>Del proceso</i></p> <ul style="list-style-type: none"> <li>El usuario debe contar con una cuenta de administrador, consulta o edición y conocer sus políticas de uso.</li> </ul> <p><i>De sistema</i></p> <ul style="list-style-type: none"> <li>El tiempo de expiración por inactividad será de 30 minutos.</li> </ul>				
<b>Flujo principal</b>	<p>Este caso de uso inicia cuando el usuario ingresa:</p> <ul style="list-style-type: none"> <li><b>Login.</b> Clave o nombre con el que el sitio identifica al usuario.</li> <li><b>Contraseña.</b> Clave password del usuario. Se permiten caracteres alfabéticos, numéricos y sin importar su orden.</li> </ul> <p>Una vez ingresados estos datos, el sistema confronta la información proporcionada (login y contraseña) contra la información de los usuarios autorizados en el sistema (E-1). Verifica que dicho usuario no tenga una sesión abierta y le permite ingresar al sistema (E-2). EL sistema determina automáticamente el perfil de usuario (administración, consulta o edición).</p> <p>Se presentan las opciones correspondientes al usuario.</p>				

<b>Caso de uso</b>	<b>Autenticación</b>
<b>Flujos de excepción</b>	<p><b>E-1 Usuario y/o contraseña no válidos</b></p> <p>El sistema notifica: "Usuario y/o contraseña no válidos. Ingrese nuevamente".Permite volver a ingresar la información.</p> <p><b>E-2 El usuario tiene una sesión abierta</b></p> <p>El sistema notifica que "El usuario tiene una sesión abierta. Debe terminarla para ingresar nuevamente".</p>

Caso de uso	Presentación de obras en línea				
Id. del CU	2	Prioridad	Alta / Media / Baja	Estado	En elaboración/propuesto/ <u>validado</u>
Actores participantes	<b>Usuario de Internet</b> – Como usuario de consulta.				
Breve descripción	Permite la consulta del acervo contenido en el sitio a los usuarios al sistema de consulta mediante la verificación de la información de su login y contraseña.				
Pre-condiciones	<b>Del proceso</b> <ul style="list-style-type: none"> <li>▪ El usuario debe contar con una cuenta de consulta.</li> </ul>				
Flujo principal	<p>Este caso de uso inicia cuando el usuario ingresa al sitio como un usuario de consulta :</p> <p>Una vez ingresado al sitio, se ingresa a la sección de consultas en Presentación de Obras en Línea.</p> <p>El sistema presenta los diferentes criterios de búsqueda:</p> <ul style="list-style-type: none"> <li>⇨ Título</li> <li>⇨ Autor</li> <li>⇨ Año</li> <li>⇨ Apellido</li> <li>⇨ Editorial</li> <li>⇨ Tema</li> <li>⇨ ISBN</li> </ul> <p>Se ingresan los datos de al menos un criterio de búsqueda, permitiendo realizar combinaciones entre los diferentes criterios. Se realiza la búsqueda y se presentan los libros encontrados (E-1).</p> <p>El libro que sea del interés del usuario puede examinarse su contenido.</p> <p>El caso de uso Presentación de Obras en Línea termina.</p>				



<b>Caso de uso</b>	<b>Presentación de obras en línea</b>
<b>Flujos de excepción</b>	<b>E-1 Título no encontrado</b> El sistema notifica: "Título no encontrado". Permite volver a ingresar la información.
<b>Post-condiciones</b>	Ninguna

Caso de uso	Libro en edición y editar				
Id. del CU	3	Prioridad	Alta / Media / Baja	Estado	En elaboración /propuesto/ <u>validado</u>
<b>Actores participantes</b>	<b>Usuario</b> – Usuario del sitio como de edición.				
<b>Breve descripción</b>	Permite la creación de libros digitales, mediante la captura de la información que los conformara.				
<b>Pre-condiciones</b>	<i>Del proceso</i> <ul style="list-style-type: none"> <li>▪ El usuario debe contar con una cuenta de edición.</li> </ul>				
<b>Flujo principal</b>	<p>Este caso de uso inicia cuando el usuario ingresa al sitio como un usuario de edición :</p> <p>Una vez ingresado al sitio, se ingresa a la sección “<i>Libros en edición y editar</i>” en el menú de edición de obras digitales, El sistema presentará los libros en edición para este usuario y la opción de comenzar a editar un nuevo libro al presionar el botón “<i>Siguiente</i>”.</p> <p>Presentándose así sucesivamente las siguientes pantallas de captura para la recopilación de la información.</p> <ul style="list-style-type: none"> <li>⇒ <b>Sección de edición A-1</b> – Permite la captura de los datos básicos del libro digital.</li> <li>⇒ <b>Notas del autor A-2</b> – Permite la captura de información relacionada con el autor; así como también de a quien va dirigida la obra y los derechos que esta tendrá durante su estancia en el sitio.</li> <li>⇒ <b>Abstract A-3</b> – Permite la captura del resumen del libro digital.</li> <li>⇒ <b>Prefacio y agradecimientos A-4-</b> Permite la captura del prefacio y agradecimientos del libro digital.</li> </ul>				

Caso de uso	Libro en edición y editar
	<p>⇒ <b>Edición de la portada A-5-</b> Permite la captura de la portada del libro.</p> <p>⇒ <b>Temario A-6-</b> Permite la captura del temario que aparecerá en el libro digital.</p> <p>⇒ <b>Generación del índice A-7-</b> Permite la captura de los temas y subtemas que componen el temario</p> <p>⇒ <b>Editor de páginas A-8-</b> Permite la captura de las páginas que componen el libro digital.</p> <p><b>Salir del editor de páginas</b> – Termina el caso de uso.</p>
<p><b>Flujos alternos</b></p>	<p><b>A-1 Sección de edición</b></p> <p>El usuario introduce todos los datos básicos para dar de alta el nuevo libro digital:</p> <ul style="list-style-type: none"> <li>▪ <b>Nombre del Autor.</b> Nombre(s) del autor.</li> <li>▪ <b>Primer Apellido.</b> Apellido paterno del autor.</li> <li>▪ <b>Segundo Apellido.</b> Apellido materno del autor.</li> <li>▪ <b>Título de la obra.</b> Título del libro digital que comenzará a ser editado.</li> <li>▪ <b>Colección.</b> Título de la colección a la que pertenece el libro digital a que se desea editar.</li> <li>▪ <b>Tema.</b> Área específica del conocimiento a la que pertenece el libro digital.</li> <li>▪ <b>ISBN.</b> International Standard Book Number.</li> <li>▪ <b>País.</b> País en el cual se esta realizando la edición del libro digital.</li> <li>▪ <b>Año de Publicación.</b> Año en el cual esta siendo editado el libro digital (año en curso).</li> </ul>

Caso de uso	Libro en edición y editar
	<ul style="list-style-type: none"><li>▪ <b>Editorial.</b> Es la casa editorial responsable del libro digital.</li><li>▪ <b>Organización.</b> Organización o institución a la que pertenece la obra digital.</li><li>▪ <b>ISBN.</b> Número del libro bajo el estándar internacional.</li><li>▪ <b>Idioma.</b> Idioma en el cual esta siendo editado el libro digital.</li></ul>
	<p><b>A-2 Notas del Autor</b></p> <p>El usuario introduce todos los datos de a quien va dirigida la obra, los derechos de esta; así como una reseña de si mismo:</p> <ul style="list-style-type: none"><li>▪ <b>Notas acerca del autor.</b> Reseña sobre la trayectoria del autor.</li><li>▪ <b>A quien va dirigida la obra.</b> Información relacionada sobre el tipo de lectores a los que esta enfocada la obra.</li></ul> <p><b>Derechos.</b> Derechos bajo los cuales la obra será expuesta en el sitio.</p>
	<p><b>A-3 Abstract</b></p> <p>El usuario introduce un resumen de la obra:</p> <ul style="list-style-type: none"><li>▪ <b>Abstract.</b> Se introduce una reseña sobre la obra.</li></ul>
	<p><b>A-4 Prefacio y agradecimientos</b></p> <p>El usuario introduce los datos del prefacio y agradecimientos de la obra:</p>

Caso de uso	Libro en edición y editar
	<ul style="list-style-type: none"><li>▪ <b>Agradecimientos.</b> Se introducen los agradecimientos que aparecerán en el libro digital.</li><li>▪ <b>Prefacio.</b> Se introduce el texto que aparecerá como prefacio.</li></ul>
	<p><b>A-5 Edición de la portada</b></p> <p>El usuario introduce la ruta de la imagen que servirá como portada de la obra:</p> <ul style="list-style-type: none"><li>▪ <b>Subir imagen.</b> Se introducen los agradecimientos que aparecerán. en el libro digital.</li></ul> <p><b>Prefacio.</b> Se introduce el texto que aparecerá como prefacio.</p>
	<p><b>A-6 Temario</b></p> <p>El usuario introduce el temario que conformará al libro digital:</p> <ul style="list-style-type: none"><li>▪ <b>Agregar capítulo.</b> Se introducen los capítulos conformarán el temario del libro digital.</li></ul>
	<p><b>A-7 Generación del índice</b></p> <p>El usuario introduce los temas y subtemas que conformarán el temario del libro digital:</p> <ul style="list-style-type: none"><li>▪ <b>Tema.</b> Se introducen el título de un tema referenciado a un punto del temario previamente definido.</li><li>▪ <b>Subtema.</b> Se introducen el título de un subtema referenciado a un punto de un tema y del temario previamente definido.</li></ul>

Caso de uso	Libro en edición y editar
	<p data-bbox="305 248 567 275"><b>A-7 Editor de páginas</b></p> <p data-bbox="305 336 1101 407">El usuario introduce los datos necesarios para poder construir una página del libro digital:</p> <ul data-bbox="342 469 1101 1033" style="list-style-type: none"> <li data-bbox="342 469 1101 495">▪ <b>Capítulo.</b> Se introduce el capítulo que pertenece a la página.</li> <li data-bbox="342 513 1101 583">▪ <b>Título.</b> Se introduce el título que aparecerá en la página del libro digital.</li> <li data-bbox="342 601 1101 672">▪ <b>Subtítulo.</b> Se introduce el Subtítulo que aparecerá en la página del libro digital.</li> <li data-bbox="342 689 1101 760">▪ <b>Texto.</b> Se introduce el texto que se presentará en la página del libro digital.</li> <li data-bbox="342 777 1101 848">▪ <b>Imagen.</b> Se introduce la ruta de una imagen que aparecerá en la página del libro digital.</li> <li data-bbox="342 866 1101 936">▪ <b>Título de la imagen.</b> Se introduce el título de la imagen que aparecerá en la página del libro digital.</li> <li data-bbox="342 954 1101 1024">▪ <b>No. de página.</b> Se introduce el número de página que aparecerá en la página del libro digital.</li> </ul>
<p data-bbox="161 1068 290 1121"><b>Flujos de excepción</b></p>	<p data-bbox="305 1086 389 1113">Ninguno</p>

Caso de uso	Registro de usuarios				
	Id. del CU	4	Prioridad	Alta / Media / Baja	Estado
<b>Actores participantes</b>	<b>Usuario de Internet</b> – Cualquier persona que navegue en Internet y que desee registrarse en el sitio				
<b>Breve Descripción</b>	Permite registrarse al sitio como usuarios de consulta o de edición.				
<b>Pre-condiciones</b>					
<b>Flujo principal</b>	<p>El caso de uso inicia cuando el usuario realiza alguna de las siguientes acciones:</p> <p>⇒ <b>Para editar A-1</b> – Permite crear una cuenta para poder editar un libro.</p> <p>⇒ <b>Para usuario de consulta A-2</b> – Permite crear una cuenta para poder realiza búsquedas avanzadas en el sitio.</p> <p><b>Salir de registro de usuarios</b> – Termina el caso de uso.</p>				
<b>Flujos alternos</b>	<p><b>A-1 Para editar</b></p> <p>El usuario introduce todos sus datos personales, para lo cual debe ingresar la siguiente información:</p> <ul style="list-style-type: none"> <li>▪ <b>Login.</b> Se introduce el nombre con el que el usuario va ser reconocido en el sitio.</li> <li>▪ <b>Password.</b> Se introduce la clave personal del usuario para que pueda ingresar al sistema.</li> <li>▪ <b>Nombre(s).</b> Nombre(s) del Usuario.</li> </ul>				

Caso de uso	Registro de usuarios
	<ul style="list-style-type: none"> <li>▪ <b>Apellido paterno.</b> Apellido paterno del usuario.</li> <li>▪ <b>Apellido materno.</b> Apellido Materno del usuario.</li> <li>▪ <b>País.</b> País de origen del usuario.</li> <li>▪ <b>Dirección.</b> Calle y número del usuario.</li> <li>▪ <b>Colonia.</b> Colonia del domicilio.</li> <li>▪ <b>Delegación.</b> Delegación del domicilio del usuario.</li> <li>▪ <b>C.P.</b> Código postal del domicilio del usuario</li> <li>▪ <b>Nivel de estudios.</b> Escolaridad del usuario.</li> <li>▪ <b>Ocupación.</b> Ocupación del usuario.</li> <li>▪ <b>Organización.</b> Entidad (educativa, empresarial) a la que pertenece el usuario.</li> <li>▪ <b>Teléfono.</b> Teléfono donde se pueda localizar al usuario.</li> <li>▪ <b>E-Mail.</b> Cuenta de correo electrónico del usuario.</li> <li>▪ <b>Solicitud.</b> Motivos por los cuales el usuario solicita una cuenta de edición.</li> </ul>
	<p><b>A-2 Para usuario de consulta</b></p> <p>El usuario introduce todos sus datos personales, para lo cual debe ingresar la siguiente información:</p> <ul style="list-style-type: none"> <li>▪ <b>Login.</b> Se introduce el nombre con el que el usuario va ser reconocido en el sitio.</li> <li>▪ <b>Password.</b> Se introduce la clave personal del usuario para que pueda ingresar al sistema.</li> <li>▪ <b>Nombre(s).</b> Nombre(s) del Usuario.</li> <li>▪ <b>Apellido Paterno.</b> Apellido paterno del usuario.</li> <li>▪ <b>Apellido Materno.</b> Apellido Materno del usuario.</li> <li>▪ <b>País.</b> País de origen del usuario.</li> </ul>



<b>Caso de uso</b>	<b>Registro de usuarios</b>
	<ul style="list-style-type: none"><li>▪ <b>Dirección.</b> Calle y número del usuario.</li><li>▪ <b>Colonia.</b> Colonia del domicilio.</li><li>▪ <b>Delegación.</b> Delegación del domicilio del usuario.</li><li>▪ <b>C.P.</b> Código postal del domicilio del usuario</li><li>▪ <b>Nivel de estudios.</b> Escolaridad del usuario.</li><li>▪ <b>Ocupación.</b> Ocupación del usuario.</li><li>▪ <b>Organización.</b> Entidad (educativa, empresarial) a la que pertenece el usuario.</li><li>▪ <b>Teléfono.</b> Teléfono donde se pueda localizar al usuario.</li><li>▪ <b>E-Mail.</b> Cuenta de correo electrónico del usuario.</li></ul>
<b>Flujos de excepción</b>	Ninguno
<b>Post-condiciones</b>	Ninguna

Caso de uso	Evaluar libro				
	Id. del CU	5	Prioridad	Alta / Media / Baja	Estado
<b>Actores participantes</b>	<p><b>Usuario de edición</b> – Usuario de edición que desea editar un libro.</p> <p><b>Comité del Sitio</b> - Conjunto de personas encargadas de aprobar las solicitudes de los usuarios para editar un libro.</p>				
<b>Breve descripción</b>	Permite que un usuario de consulta realice una petición al Comité del Sitio para poder crear o editar su libro digital.				
<b>Pre-condiciones</b>	El usuario debe de ser de edición.				
<b>Flujo principal</b>	<p>El caso de uso inicia cuando el usuario solicita una evaluación de un libro. Debe de llenar una solicitud que contiene la siguiente información:</p> <ul style="list-style-type: none"> <li>▪ <b>Nombre(s).</b> Nombre(s) del Usuario.</li> <li>▪ <b>Apellido paterno.</b> Apellido paterno del usuario.</li> <li>▪ <b>Apellido materno.</b> Apellido Materno del usuario.</li> <li>▪ <b>Título de la obra.</b> Título del libro con que pretende el usuario dar de alta la obra digital.</li> <li>▪ <b>Tema.</b> Tema del libro.</li> <li>▪ <b>Organización.</b> Organización a la que pertenece a el usuario.</li> <li>▪ <b>Descripción de motivos de la obra digital.</b> Descripción detallada del contenido del libro para que el comité del sitio pueda evaluar si es conveniente permitir la edición del libro en el sitio.</li> <li>▪ <b>Correo electrónico.</b> Cuenta de correo electrónico del usuario.</li> </ul> <p><b>Salir de registro de evaluar libro</b> – Termina el caso de uso.</p>				

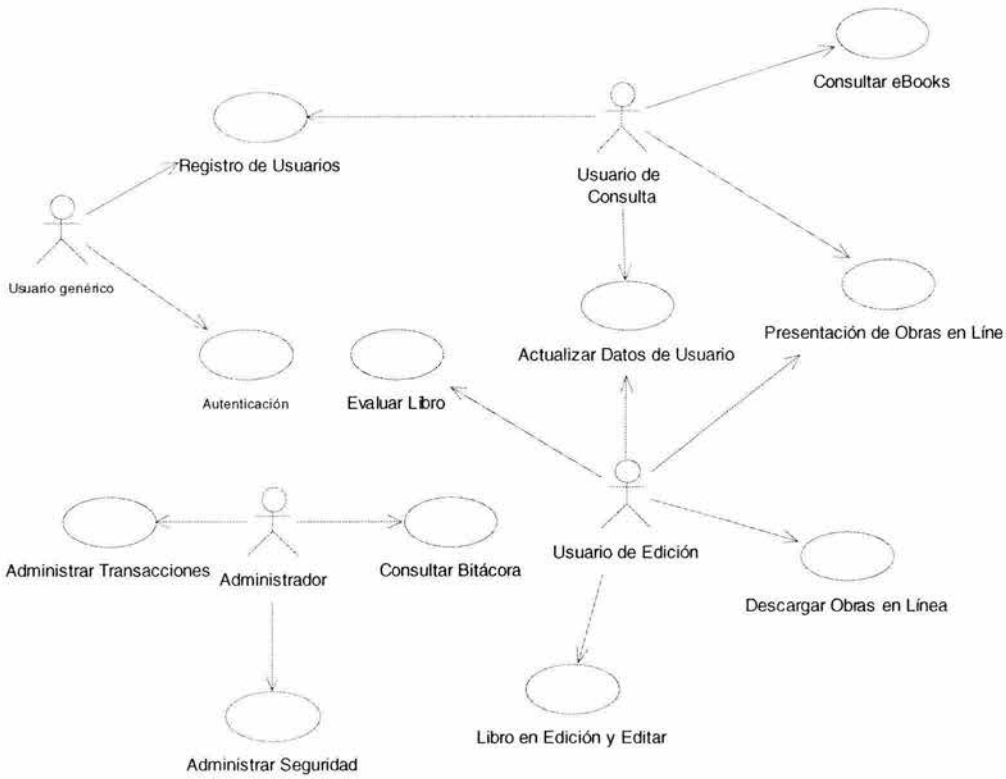
<b>Caso de uso</b>	<b>Evaluar libro</b>
<b>Flujos de excepción</b>	Ninguno
<b>Post-condiciones</b>	Ninguna

Caso de uso	Consulta de libros digitales				
Id. del CU	6	Prioridad	Alta / Media / Baja	Estado	En elaboración/propuesto/validado
<b>Actores participantes</b>	<b>Usuario de Internet</b> – Usuario que tiene acceso a la página principal del sitio y realiza una consulta al acervo del mismo.				
<b>Breve descripción</b>	Permite la consulta del acervo contenido en el sitio.				
<b>Pre-condiciones</b>	Ninguna				
<b>Flujo principal</b>	<p>Este caso de uso inicia cuando el usuario ingresa al sitio:</p> <p>Una vez ingresado al sitio, se ingresa a la sección de Consultas de Ebooks.</p> <p>El sistema presenta los diferentes criterios de búsqueda:</p> <ul style="list-style-type: none"> <li>☞ Título</li> <li>☞ Autor</li> <li>☞ Tema</li> </ul> <p>Se ingresan los datos de al menos un criterio de búsqueda. Se realiza la búsqueda y se presentan los libros encontrados, así como un resumen del número de libros encontrados.</p> <p>El caso de consulta de libros digitales termina.</p>				
<b>Flujos de excepción</b>	Ninguno				

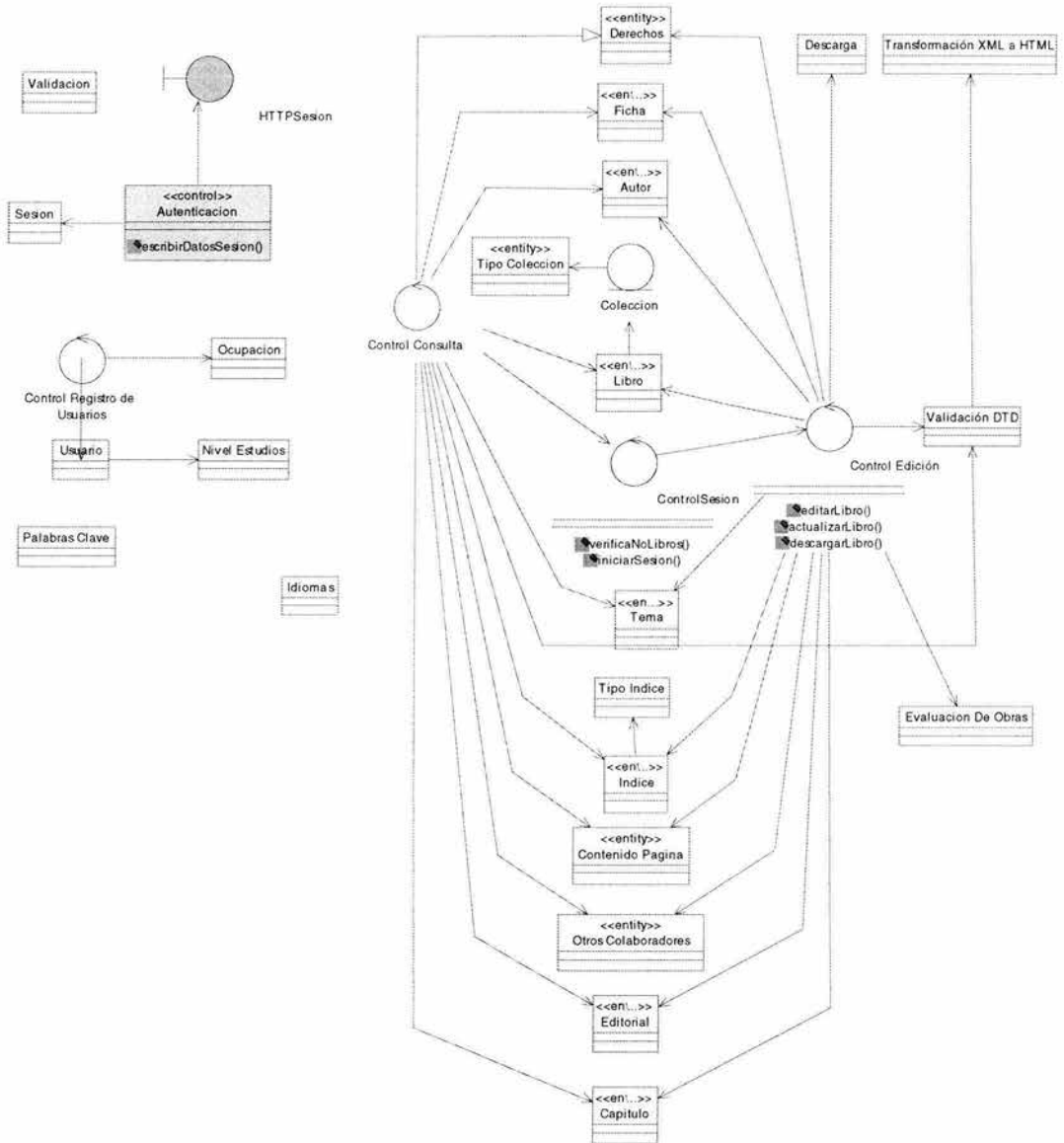
Caso de uso		Actualizar datos de usuario			
Id. del CU	7	Prioridad	Alta / Media / Baja	Estado	En elaboración/propuesto/validado
<b>Actores participantes</b>	<p><b>Usuario de consulta</b> – Usuario que accede a la página principal del sitio y realiza una consulta del acervo.</p> <p><b>Usuario de edición</b> – Usuario de edición que desea editar un libro.</p>				
<b>Breve descripción</b>	Permite la modificación de datos personales de los usuarios de edición y consulta.				
<b>Pre-condiciones</b>	El usuario debe de ser de edición o consulta.				
<b>Flujo Principal</b>	<p>Este caso de uso inicia cuando el usuario ingresa al sitio:</p> <p>Una vez ingresado al sitio, se ingresa a la sección de actualización de datos del usuario.</p> <p>El sistema presenta los siguientes rubros para la modificación de datos.</p> <ul style="list-style-type: none"> <li>▪ <b>Nombre(s).</b> Nombre(s) del Usuario.</li> <li>▪ <b>Apellido paterno.</b> Apellido paterno del usuario.</li> <li>▪ <b>Apellido materno.</b> Apellido Materno del usuario.</li> <li>▪ <b>País.</b> País de origen del usuario.</li> <li>▪ <b>Dirección.</b> Calle y número del usuario.</li> <li>▪ <b>Colonia.</b> Colonia del domicilio.</li> <li>▪ <b>Delegación.</b> Delegación del domicilio del usuario.</li> <li>▪ <b>C.P.</b> Código postal del domicilio del usuario</li> <li>▪ <b>Nivel de estudios.</b> Escolaridad del usuario.</li> <li>▪ <b>Ocupación.</b> Ocupación del usuario.</li> <li>▪ <b>Organización.</b> Entidad (educativa. empresarial) a la que pertenece el usuario.</li> <li>▪ <b>Teléfono.</b> Teléfono donde se pueda localizar al usuario.</li> </ul>				

<b>Caso de uso</b>	<b>Actualizar datos de usuario</b>
	<ul style="list-style-type: none"><li>▪ <b>E-Mail.</b> Cuenta de correo electrónico del usuario.</li></ul> <p>Se ingresan los datos que se desean modificar y se realiza el cambio.</p> <p>El caso de consulta de actualización de datos de usuario termina.</p>
<b>Flujos de excepción</b>	Ninguno
<b>Post-condiciones</b>	Ninguna

### 6.6 Diagrama de casos de uso



### 6.7 Diagrama de Clases





Detalle de las Clases

Usuario
setUsr_Nombre()
getUsr_Nombre()
setUsr_ApPaterno()
getUsr_ApPaterno()
setUsr_ApMaterno()
getUsr_ApMaterno()
setUsr_Login()
getUsr_Login()
setUsr_Password()
getUsr_Password()
setUsr_Direccion()
getUsr_Direccion()
setUsr_Telefono()
getUsr_Telefono()
setUsr_Mail()
getUsr_Mail()
setUsr_Pais()
getUsr_Pais()
setUsr_Ciudad()
getUsr_Ciudad()
setUsr_Organizacion()
getUsr_Organizacion()
setUsr_Motivos()
getUsr_Motivos()
setUsr_Status()
getUsr_Status()

Sesion
sesClave : Integer
sesUsrLogin : String
sesIP : String
sesInicio : Timestamp
sesTiempo : Integer
sesInactividad : Integer
sesUsuPerf
sesDepClave
sesUniClave
sesNombre
sesApelPat
sesApelMat
setClaveSesion()
getClaveSesion()
setLogin()
getLogin()
setIP()
getIP()
setIniSesion()
getIniSesion()
setTiempoSesion()
getTiempoSesion()
setInactividad()
getInactividad()
obtenerDatosSesion()
crearSesion()
terminarSesion()
validarSesion()

<<entity>> Ficha
setFicha_Id()
getFicha_Id()
setFicha_Autor()
Ficha()
getFicha_Autor()
setFicha_Titulo()
getFicha_Titulo()
setFicha_Titulo()
getFicha_Titulo()
setFicha_Anio()
getFicha_Anio()
setFicha_Mes()
getFicha_Mes()
setFicha_Dia()
getFicha_Dia()
setFicha_Contenido()
getFicha_Contenido()
setFicha_TipoContenido()
getFicha_TipoContenido()

<<entity>> Libro
setLib_ISBN()
getLib_ISBN()
setLib_Anio()
getLib_Anio()
setLib_SitioPubli()
getLib_SitioPubli()
setLib_OrgResp()
getLib_OrgResp()
setLib_PortmagNombre()
getLib_PortmagNombre()
Libro()

<<entity>> Autor
setAutor_Id()
getAutor_Id()
setAutor_Nombre()
getAutor_Nombre()
setAutor_ApellPat()
getAutor_ApellPat()
setAutor_ApellMat()
getAutor_ApellMat()
Autor()

Validacion
validarFecha()
validarAlfaNum()
validarLong()
validarExt()
validarNum()
validarAlf()
validarCorreoElectr()
validarNombre()
validarApel()

<<entity>> Indice
setInd_Tema()
getInd_Tema()
setInd_SubTema()
getInd_SubTema()
Indice()

<<entity>> Derechos
setDer_Id()
getDer_Id()
setDer_Desc()
getDer_Desc()
Derechos()

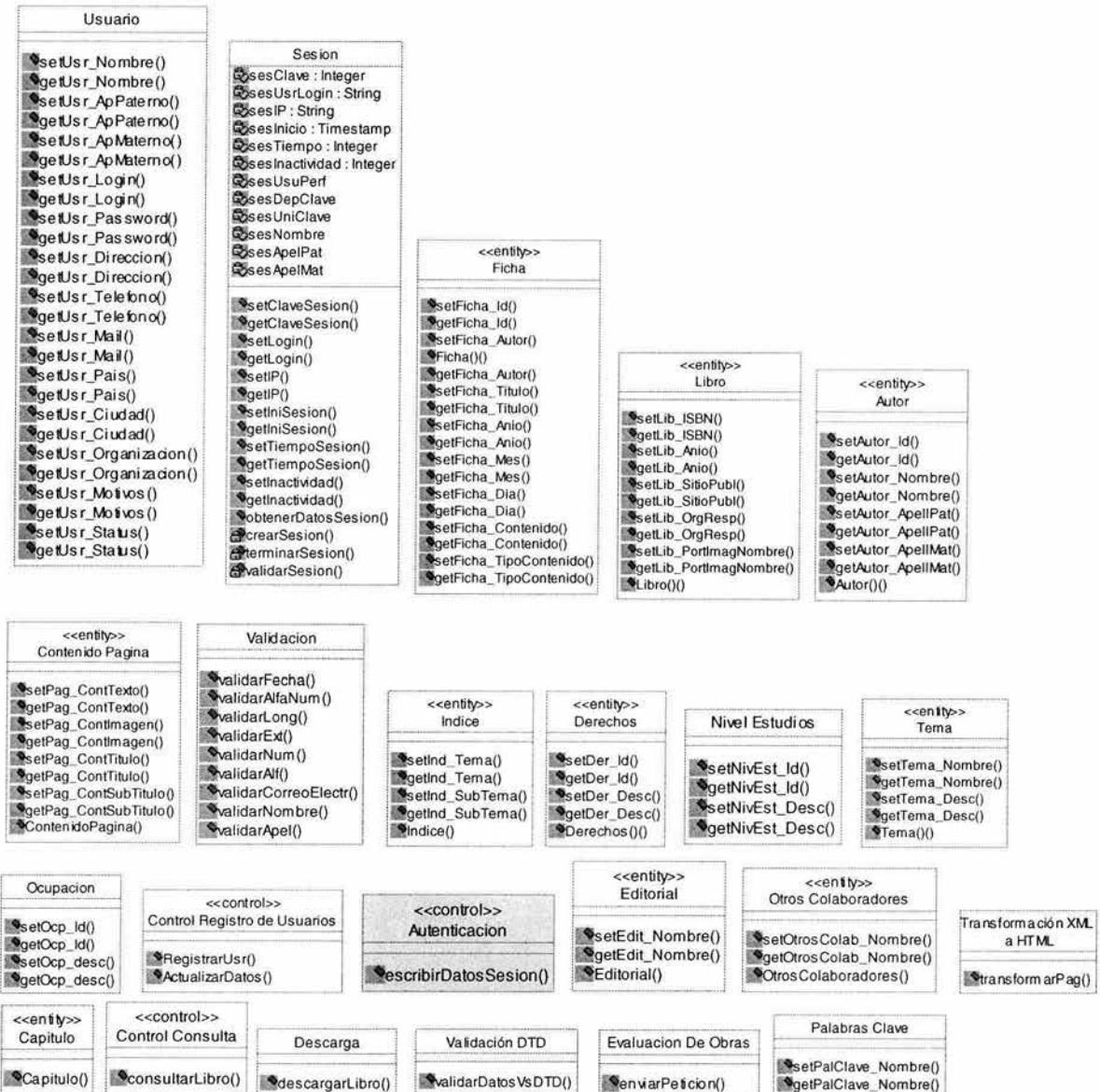
Nivel Estudios
setNivEst_Id()
getNivEst_Id()
setNivEst_Desc()
getNivEst_Desc()

<<entity>> Tema
setTema_Nombre()
getTema_Nombre()
setTema_Desc()
getTema_Desc()
Tema()

Ocupacion
setOcp_Id()
getOcp_Id()
setOcp_desc()
getOcp_desc()

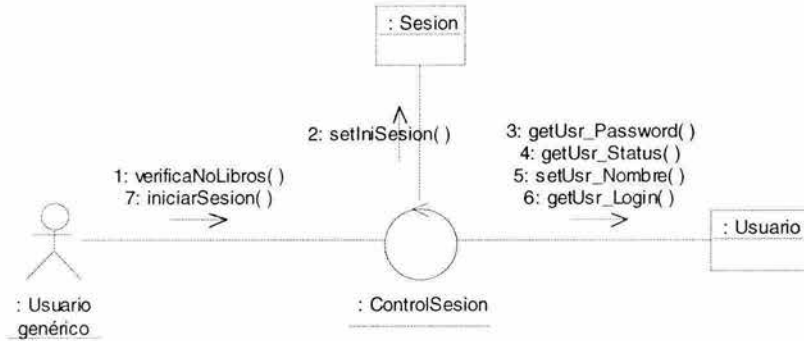
<<control>> Control Registro de Usuarios
RegistrarUsr()
ActualizarDatos()

<<control>> Autenticacion
escribirDatosSesion()

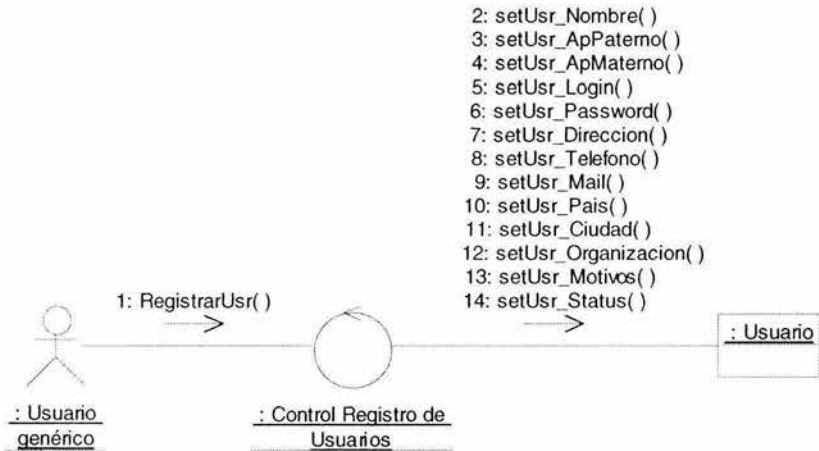


## 6.8 Diagramas de colaboración

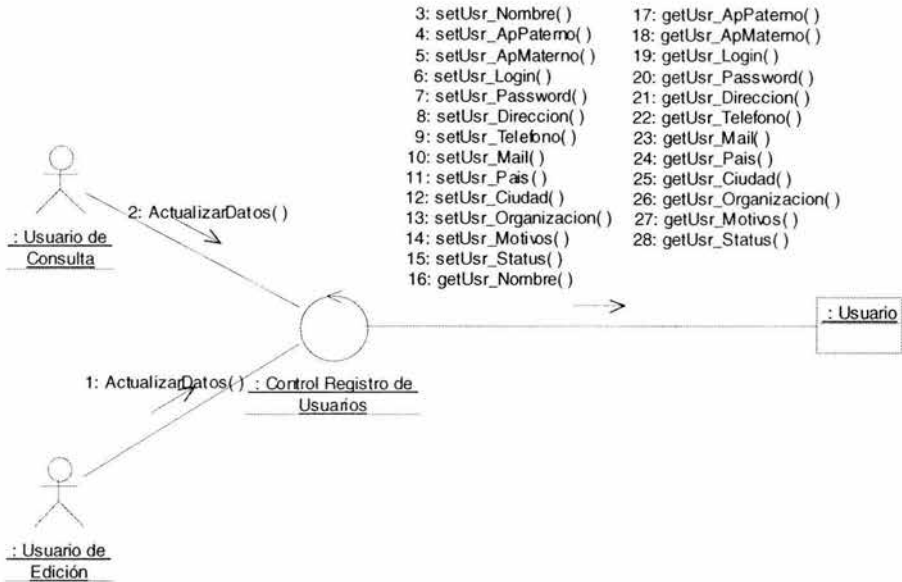
### a) Autenticación:



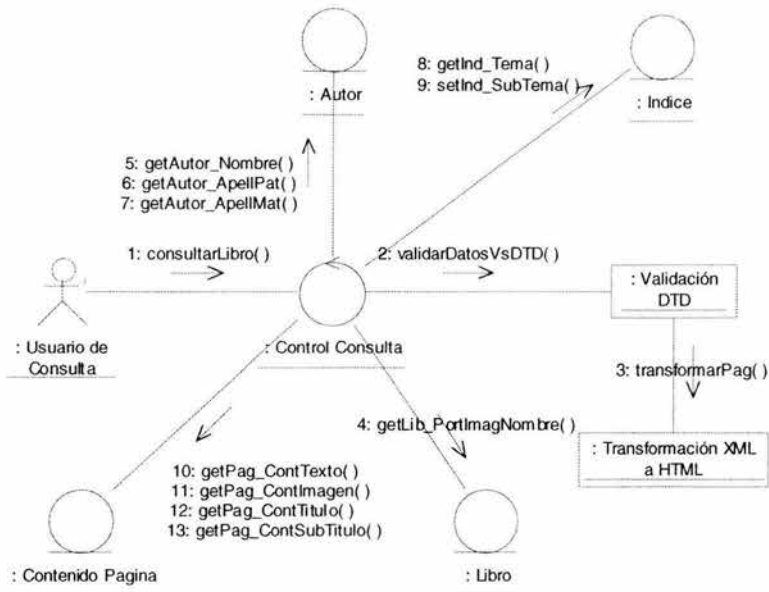
### b) Registro de usuarios:



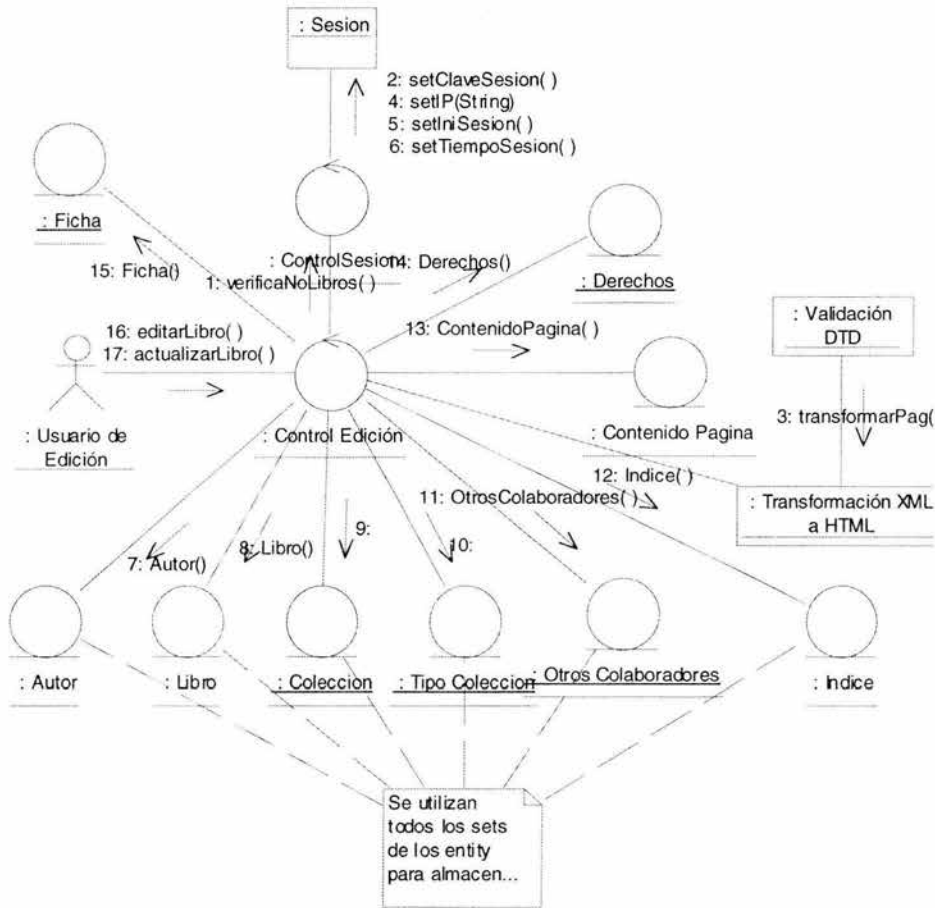
c) Actualizar datos de usuario:



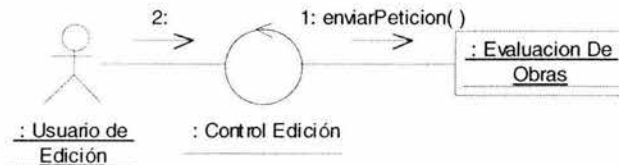
d) Presentación de obras en línea:



e) Libro en edición y editar:

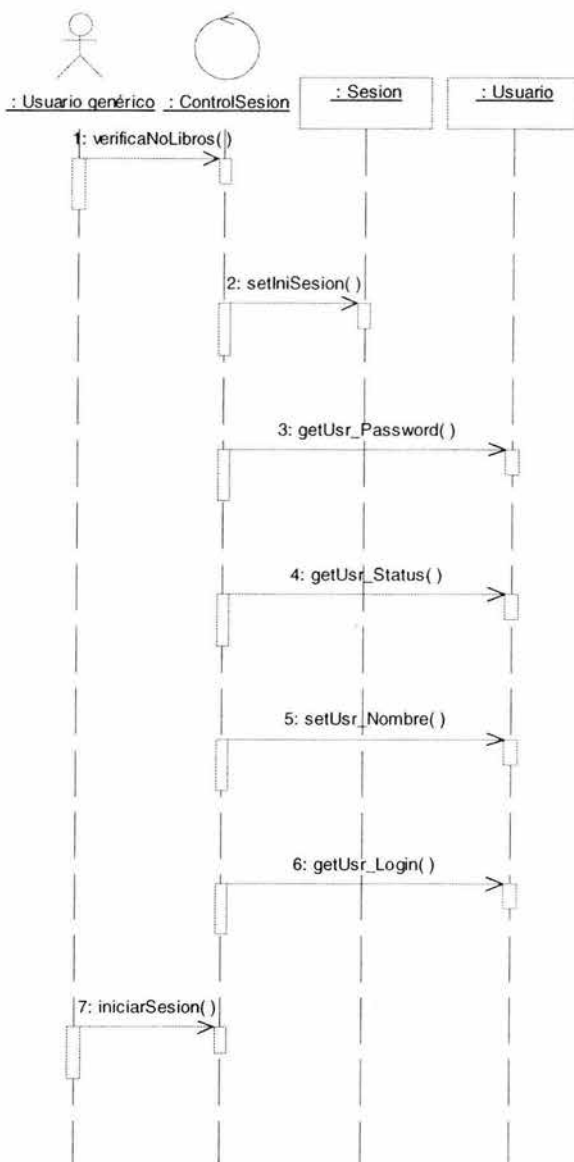


f) Evaluar libro:

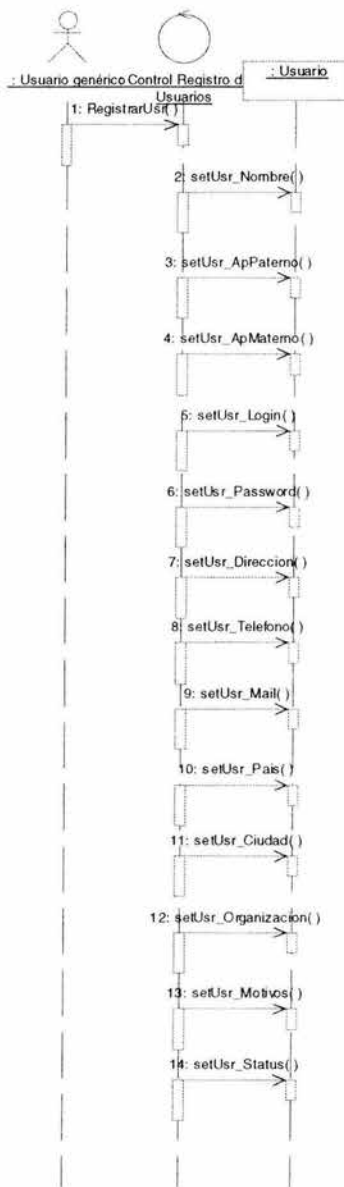


## 6.9 Diagramas de secuencia

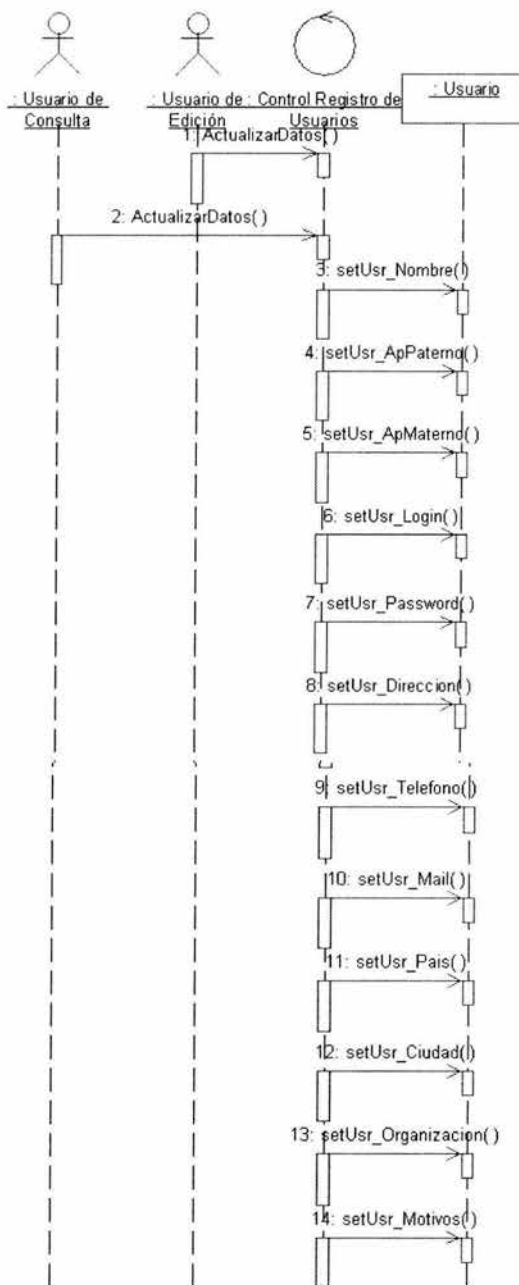
### a) Autenticación:



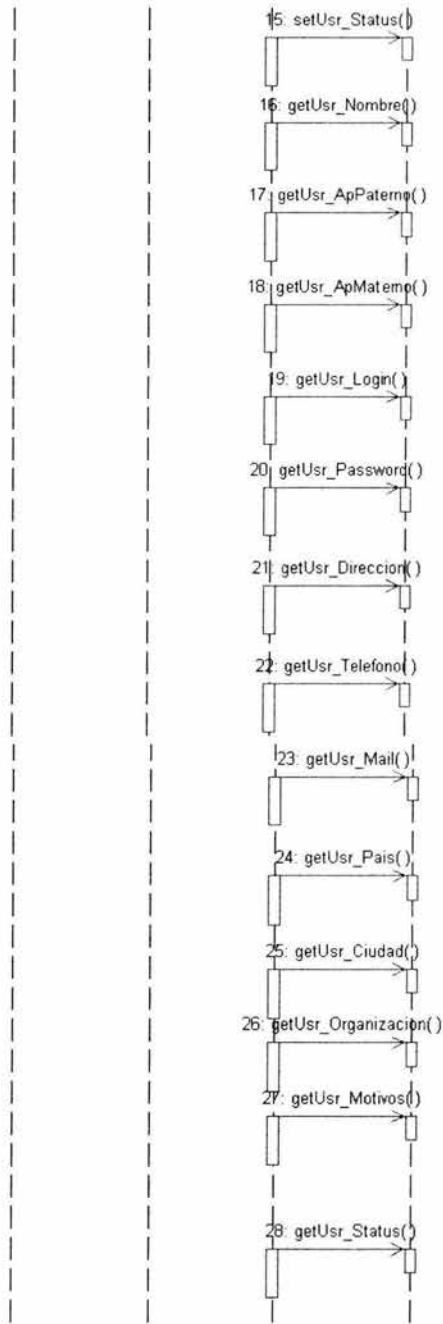
b) Registro de usuario:



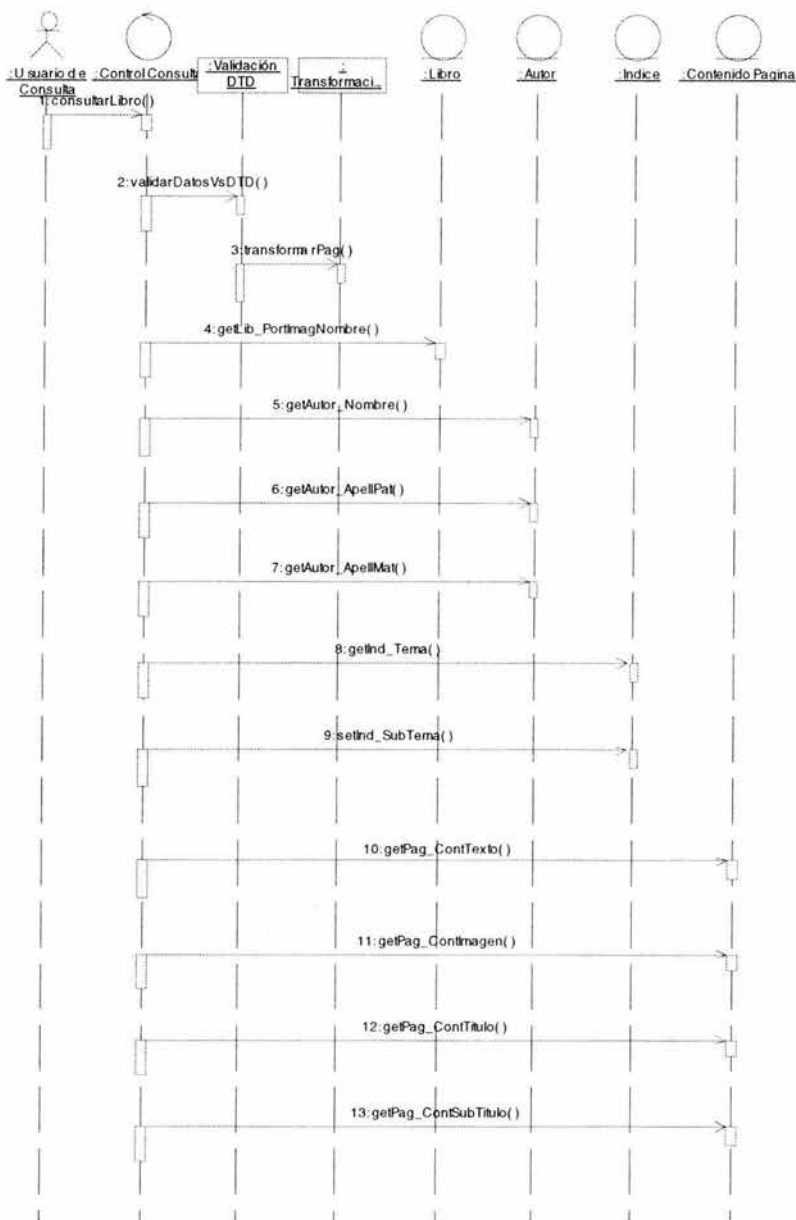
c) Actualizar datos de usuario:



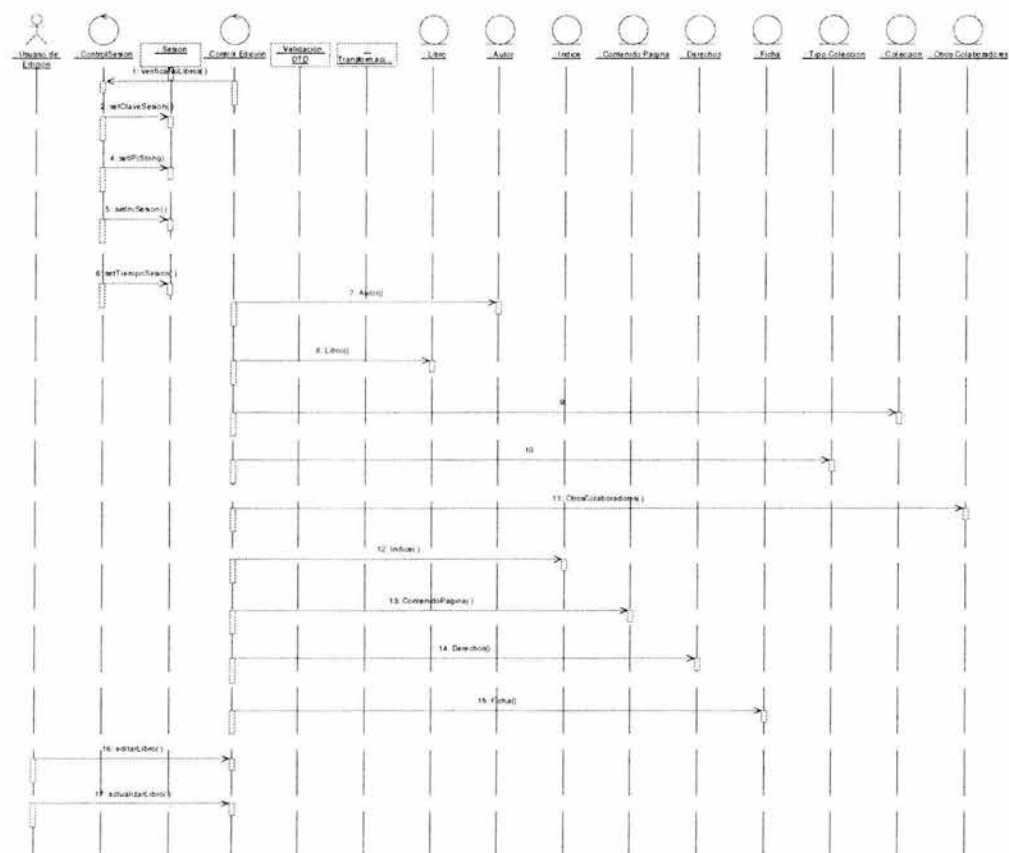




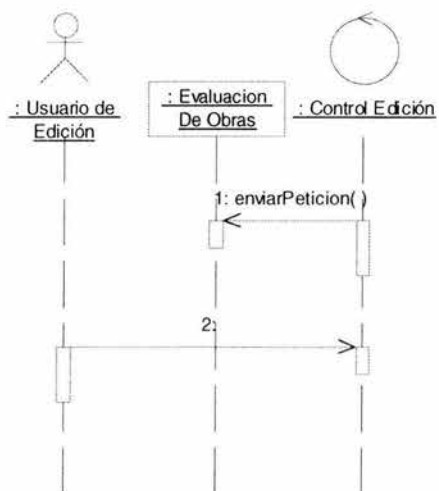
d) Presentación de obras en línea:



d) Libro en edición y editar:



e) Evaluar libro:



## 6.10 Requerimientos de hardware y software

### **Servidor:**

#### **Hardware.-**

- ❖ Procesador Pentium III o posterior.
- ❖ Disco duro de 80 GB. Esta característica depende de las necesidades que requiera la aplicación para su implementación.
- ❖ Memoria RAM mínima de 512 MB.
- ❖ Tarjeta de red con una transferencia de datos mínima de 10/100 Mbps.

#### **Software.-**

- ❖ Sistemas Operativos Linux (Red Hat 7.3 o posterior), Solaris 8.0 o posterior, Windows, NT, 2000, XP.

**Nota:** Para los sistemas Windows se tienen las siguientes alternativas:

1.- Distribuir la aplicación en dos máquinas; servidor web y servidor de bases de datos.

2.- Crear la base de datos en MS-SQL Server y adecuar la conexión al respectivo RDBMS.

- ❖ Base de Datos Sybase versión 11.9.2.
- ❖ Servidor web: Jakarta-Tomcat-4.1.12 o posterior.
- ❖ Java versión: jdk1.4.1\_01 o posterior.
- ❖ XML versión 1.0.

*Cliente:*

Hardware.-

- ❖ PC con procesador mínimo de Pentium I.
- ❖ Memoria RAM mínima de 64 MB.
- ❖ Modem con una velocidad de transferencia de 56 Kbps.

Software.-

- ❖ Navegador Explorer Versión. 6.0 en adelante, Mozilla ó Netscape versión 7.0.

### **6.11 Alcances**

- ❖ Consultar obras digitales del acervo en línea.
- ❖ Editar ya sea en línea o mediante un proceso batch, obras digitales bien estructuradas.
- ❖ Descargar capítulos o la obra digital de acuerdo a los pertinentes permisos por parte del autor.
- ❖ Navegar y presentar las obras digitales bajo estructuras de documentos bien construidos.

### **6.12 Exclusiones**

- ❖ No se ha considerado interacción con ningún otro sistema de Libros Digitales.
- ❖ Por el momento la parte de descargas de obras digitales no está incluida.
- ❖ La presentación en cuanto al texto es uniforme para todas las obras digitales con esto se quiere decir que se va a tener un mismo tipo de letra, tamaño, estilo, conformación de la página y tipo de imágenes.

El sistema excluye cualquier funcionalidad que no haya sido incluida expresamente dentro de los alcances del mismo.

### **6.13 Posibilidades de expansión**

El sistema tiene todas las posibilidades de expansión, ya que debido a la plataforma en la que fue desarrollado permite crecer libremente, realizando mejoras de casi cualquier tipo.

El usuario podrá proponer sus propias etiquetas enriqueciendo la DTD para un manejo mas personalizado de la información de los temas, y serán evaluadas por el comité editorial, dichas sugerencias serán mediante correo electrónico.

Se podrán recibir diferentes formatos de texto (archivos con extensión **txt**, **doc**, etc.) para poder publicar obras previamente terminadas por el autor y subirlas al sitio realizando una validación contra las DTD correspondientes. Esto se realizará por capítulos mediante un proceso en batch realizado por el sistema y la rapidez dependerá de las limitantes físicas de la PC del usuario.

Se podrá definir la alineación del párrafo e imágenes (Izquierda, Centrado, Derecha), así como el tipo de letra

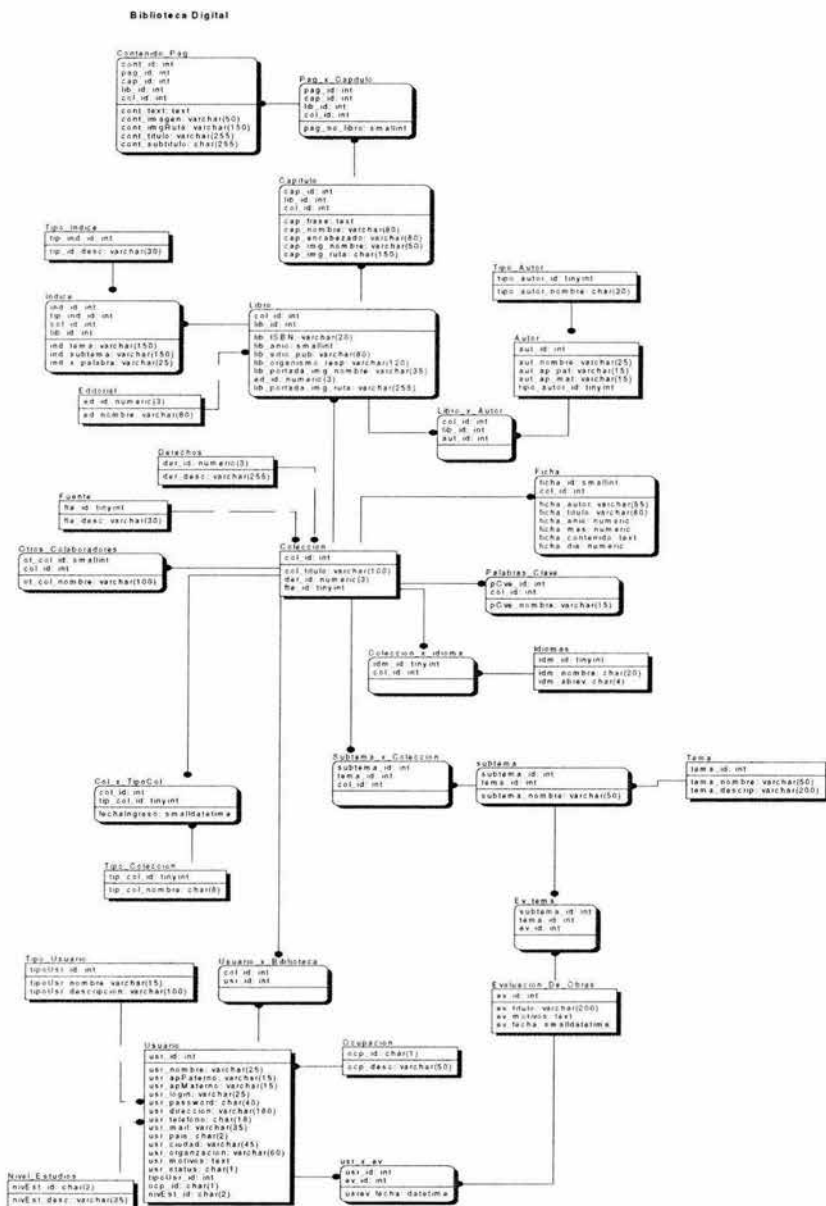
### **6.14 ¿Quién puede utilizar el sistema?**

Personas interesadas en realizar consultas bibliográficas en el acervo, o aquellas que se encuentren interesadas en publicar sus propias obras digitales en un contexto cultural - educativo.

### **6.15 Base de datos**

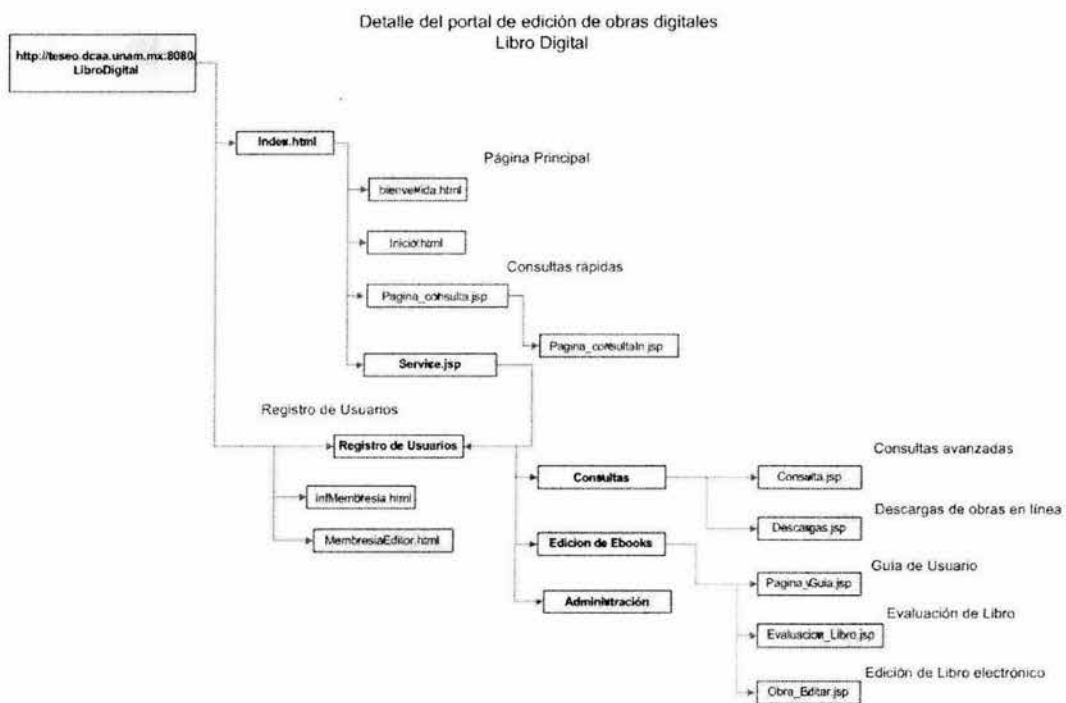
La base de datos utilizada es **Sybase versión 11.9.2** para Linux Red Hat. Sybase es una de las mayores compañías independientes de software, y su manejador de base de datos es libre para el Sistema Operativo Red Hat con fines educativos.

## Diagrama Entidad - Relación





## 6.16 Diagrama del sitio



---

---

## CONCLUSIONES

Las características más importantes de la información, se basan en que ésta sea: confiable, clara, puntual, oportuna, eficaz, veraz y por supuesto, no redundante. Estos puntos contribuyen a que la información tenga un sentido único de valor, algunas veces incluso es un valor de vital importancia. De ahí que la información contribuye al desarrollo económico, cultural y social en beneficio del mundo moderno.

Un derivado de la investigación aquí presentada, puede destacar que el resguardo, almacenamiento, seguridad y presentación de la información es uno de los mayores retos de la ingeniería de software, de tal suerte, que prácticamente, la invención de la misma rama de la ingeniería se basó en este principio que a simple vista pudiera parecer simplista pero que en realidad es una actividad de magnitudes incluso inimaginables en medida en que la información ha ido evolucionando hasta convertirse en el paradigma que es actualmente.

Una biblioteca digital es un repositorio de acervos tipificado de obras digitales que se encuentran almacenados en formatos electrónicos y que generalmente cuentan con una colección limitada de temas. El objetivo de las bibliotecas digitales es contribuir a la difusión masiva de la educación y la cultura, así como preservar el contenido de los libros sin que estos se deterioren con el paso del tiempo.

Los libros digitales hoy en día son el medio en boga para la difusión del conocimiento prácticamente de cualquier rama. Gracias a ellos, la ciencia y la tecnología actualmente se producen en cantidades industriales, por lo que la creación y generación de nuevo conocimiento se ha incrementado incluso de manera exponencial en comparación con otras épocas de la humanidad. Existen análisis que estiman que a mediados del siglo en curso prácticamente todo el conocimiento humano se encontrará inmerso en este tipo de tecnología.

La tecnología Java aunada con el lenguaje de marcado extendido (XML), y las bases de datos relacionales son las herramientas perfectas para la creación, generación, manipulación y almacenamiento de datos que en su conjunto se traducen en el sentido amplio de lo que conocemos como información. Estas herramientas administran la información de manera oportuna, y es gracias a la base de datos relacional que ésta cuenta con características únicas por ser de carácter no redundante.

Por tanto el manejo de datos a través del World Wide Web es desde finales del siglo pasado una realidad al servicio de los sistemas de cómputo.

Así, el manejo oportuno de la información puede ser aplicado para controlar y administrar sitios de dominio público como lo son los nichos de negocio empresarial, el comercio o la banca electrónica y en general la administración de aplicaciones sofisticadas que manejen información de usuarios y entidades reales.

El manejo de información a través de Internet permite la mitigación de problemas sociales, económicos, políticos, ecológicos, así como la prevención de desastres naturales o incluso la ahora llamada administración de salud social.

Hoy en día en México existen aplicaciones que se desarrollan con el objetivo de la administración de recursos humanos, transacciones empresariales y una gama de sistemas que administran información en Internet de carácter público, social, así como de gobierno.

### **Logros de la aplicación de publicación de libros digitales.**

El proyecto de tesis realizado es un proyecto prototipo que permite crear, editar y administrar libros en línea, dentro de su nicho se pueden realizar actividades de consulta, edición y publicación de un libro digital.

A continuación destacaremos los puntos medulares de la aplicación desarrollada:

- ❖ La conjunción de los lenguajes de programación Java y XML permiten una gran versatilidad ya que Java proporciona libertad en cuanto a plataforma y XML brinda mayor facilidad para el manejo de datos. Todo esto se traduce en un mejor desempeño y estandarización de documentos electrónicos.
- ❖ Se cuenta con mayor extensibilidad del sistema a posteriori debido a que los lenguajes ocupados se encuentran en una fase de expansión, además de ser cada vez mas utilizados en el desarrollo de aplicaciones para Internet.
- ❖ Permite maximizar el nivel de almacenamiento de las obras digitales aumentando la capacidad del acervo en medida de lo que se requiera.
- ❖ Brinda la facilidad de visualizar los documentos digitales respetando los derechos de autor.
- ❖ Se desarrolló bajo el paradigma de software libre, por tanto sus costos son relativamente bajos, pero dependen de la magnitud de la implementación deseada.

### Estado actual del Sistema

El sistema permite realizar consultas de las obras digitales, edición y creación de las mismas, cuenta con un pequeño motor de búsquedas especializadas que permite encontrar los detalles de la obra con mayor rapidez, y finalmente, se diseñaron componentes que mantienen un manejo eficiente y flexible para la construcción, edición y administración de las obras digitales en línea.

### Perspectivas a futuro

El proyecto resultante maneja una gran capacidad de expansión y es ampliamente eficaz dadas sus características versátiles de implementación. De esta manera, en caso de que se requiriera un cambio en las reglas de negocio, podrían realizarse prácticamente de manera transparente y por sus características modulares, se requiere poco esfuerzo por parte del personal que lo administre, solo se necesitaría agregar los módulos correspondientes, volver a compilar y levantar la aplicación para ver los cambios reflejados en las interfaces correspondientes.

La aplicación desarrollada se enfocó a la creación de libros electrónicos, pero otra de sus aplicaciones y adaptaciones podría extenderse para manejar revistas electrónicas, ya que la base de datos es capaz de soportar éstas sin necesidad de cambiar el esquema que se maneja actualmente.

XML y Java son lenguajes que están en evolución continua, y por tanto se están convirtiendo en herramientas estándares en el mundo del Internet; Por ello es fácil el mantenimiento y la expansión del Sistema.

### Último comentario.-

Si para la economía clásica de David Ricardo existió la llamada mano invisible, bien pudiera ser que en los tiempos de la globalización, esa mano invisible cuente con una poderosa pala virtual que se llama Internet y sus aplicaciones, pero lejos de servir sólo al poderío de un sistema económico, nación o credo, es una herramienta universal con una miríada de posibilidades y capacidades al servicio de las naciones.

Por lo pronto, Internet y las aplicaciones que se soportan en él, están haciendo una notable labor, y no cabe duda que aportaran el eslabón más importante en el futuro de la información más preciada por la humanidad, los libros.

---

## **APENDICE**

### **MANUAL DE USUARIO DEL SISTEMA**

## INDICE

<b>I. ANTECEDENTES .....</b>	<b>153</b>
1.1 BIBLIOTECA DIGITAL .....	153
<b>II. CONVENCIONES Y TÉRMINOS.....</b>	<b>154</b>
2.1 CONVENCIONES RELATIVAS AL TEXTO.....	154
2.2 CONVENCIONES RELATIVAS AL USO DEL RATÓN.....	154
2.3 ALGUNOS TÉRMINOS UTILIZADOS.....	154
<b>III. REQUERIMIENTOS TÉCNICOS .....</b>	<b>156</b>
3.1 DESCRIPCIÓN GENERAL DE ARQUITECTURA.....	156
3.2 REQUERIMIENTOS DE HARDWARE .....	157
<b>IV. DESCRIPCIÓN GENERAL DE LA PANTALLAS .....</b>	<b>158</b>
4.1 ÁREA DE CAPTURA Y MANEJO DE DATOS .....	158
4.2 CONTROLES DE CAPTURA.....	161
4.3 ÁREA DE BOTONES .....	161
4.4 MENSAJES DEL SISTEMA .....	163
4.5 INICIO DE SESIÓN .....	164
<b>V. DEFINICIÓN DE ROLES.....</b>	<b>165</b>
5.1 Administrador .....	165
5.2 Edición.....	165
5.3 Consulta .....	167
<b>VI. DESCRIPCIÓN DEL PROCESO DE CONSULTA .....</b>	<b>168</b>
<b>VII. DESCRIPCIÓN DEL PROCESO DE EDICIÓN .....</b>	<b>173</b>
<b>VIII. MÓDULO DE ADMINISTRACIÓN .....</b>	<b>178</b>

## ANTECEDENTES

### 1.1 BIBLIOTECA DIGITAL

Una biblioteca digital es un sistema diseñado para recopilar, almacenar y organizar acervos de información que se guardan electrónicamente, para ponerlos a la disposición de la comunidad.

Entonces podemos también definir a una Biblioteca Digital en el ámbito de una cultura informática, como una colección organizada de documentos almacenados en un formato digital, que a su vez ofrece servicios de búsqueda y recuperación de información.

Los documentos que integran el acervo de una biblioteca digital son textos, imágenes, video, sonido o combinaciones de éstos. Lo ideal para una Biblioteca Digital es que la recuperación de documentos sea en forma completa, realizando las búsquedas sobre el texto completo de los mismos y con posibilidad de recuperación inmediata.


Además este sistema de búsqueda dentro del concepto de Biblioteca Digital se amplía, no solo a la búsqueda tradicional sino también por medio de citas referenciales de autor, título, descripción y palabras clave.

Por lo que una Biblioteca Digital no es un ente aislado ya que requiere de tecnologías específicas para compartir y enlazar todos sus recursos, los enlaces entre sus recursos deben de ser transparentes para el usuario; y todo esto con un fin específico permitir el acceso universal a diversos acervos y servicios de información que poco a poco se está convirtiendo en una meta obligada para la humanidad.



## II. CONVENCIONES Y TÉRMINOS

### 2.1 CONVENCIONES RELATIVAS AL TEXTO

- Los botones que deben de presionarse dentro de la aplicación aparecen de color naranja, por ejemplo  ACEPTAR
- Los términos importantes del sistema aparecen en **negritas** y con la palabra **Nota**.
- Los títulos de sistema aparecen con el siguiente formato **Título** (negrita y de mayor tamaño).

### 2.2 CONVENCIONES RELATIVAS AL USO DEL RATÓN

- El botón izquierdo generalmente se encuentra configurado como el principal, por lo que siempre que se mencione “botón del mouse” se hace referencia a este botón, a menos que se especifique lo contrario.
- Si en algún procedimiento se usa el secundario, debe tenerse en cuenta que se hace referencia al botón derecho.

### 2.3 ALGUNOS TÉRMINOS UTILIZADOS

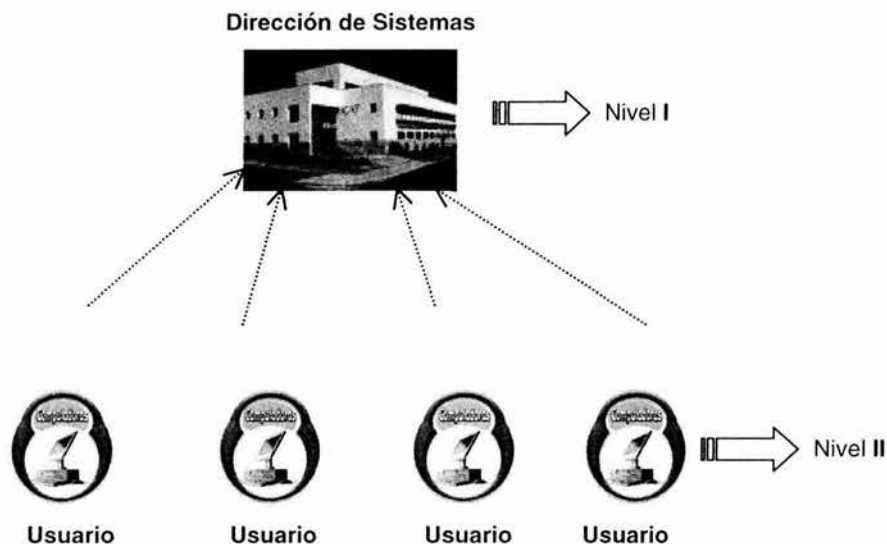
- **Botón:** Rectángulo que tiene un icono descriptivo o un título que indica el comando o acción que realizará al ser presionado.
- **Ventana:** Entiéndase como el marco que contiene información. Cada ventana posee una barra de título que identifica a la aplicación y su contenido; al estar activa, la barra del menú activo aparece en un tono de color resaltado.

- **Hacer clic:** Significa apuntar a un objeto con el mouse, después pulsar y soltar el botón izquierdo del mouse, de manera rápida
- **Hacer doble clic:** Significa apuntar a un objeto con el mouse, después pulsar y soltar el botón izquierdo del mouse, dos veces en forma muy rápida.
- **Campo bloqueado:** Significa que el campo referido presenta información que no se puede modificar o alterar directamente. Comúnmente se presenta en campos que despliegan información de una tabla o catálogo que tiene relación con dicha pantalla.
- **Valor requerido:** Significa que debe introducirse un valor en el campo especificado, como un requisito para que el registro sea almacenado. En caso de que el campo se encuentre sin información o con información errónea, el sistema presentará el mensaje correspondiente. Dichos campos tienen un \* indicando que es requerido en una nota en la parte inferior de la pantalla.

## REQUERIMIENTOS TÉCNICOS

### 3.1 DESCRIPCIÓN GENERAL DE ARQUITECTURA

Para la infraestructura de comunicaciones y de informática, se utilizan niveles para detectar las diferentes necesidades. Como se puede apreciar en la siguiente gráfica, en el primer nivel se encuentra la Dirección de Sistemas la cual proporciona el servicio del Libro Digital, y el segundo corresponde a los usuarios que hacen uso de éste.



Para realizar el envío de información del nivel I al II se utilizará Internet, debido a lo cual en los dos niveles se debe contar con una conexión a Internet mediante un ISP (Proveedor de Servicio de Internet).

## 3.2 REQUERIMIENTOS DE HARDWARE

De acuerdo a la evaluación de la arquitectura propuesta, las operaciones a realizar, el volumen de información que se maneja y el nivel de respuesta deseado. Se propone la siguiente configuración mínima de equipos para los diferentes niveles.

### **Nivel I. Dirección de Sistemas:**

- Procesador mínimo de Pentium III (min. 866 MHz).
- Disco duro de 80 GB. (Todo depende de las necesidades que se vayan presentando).
- Memoria RAM mínima de 512 MB.
- Tarjeta de red con una transferencia de datos mínima de 10/100 Mbps.

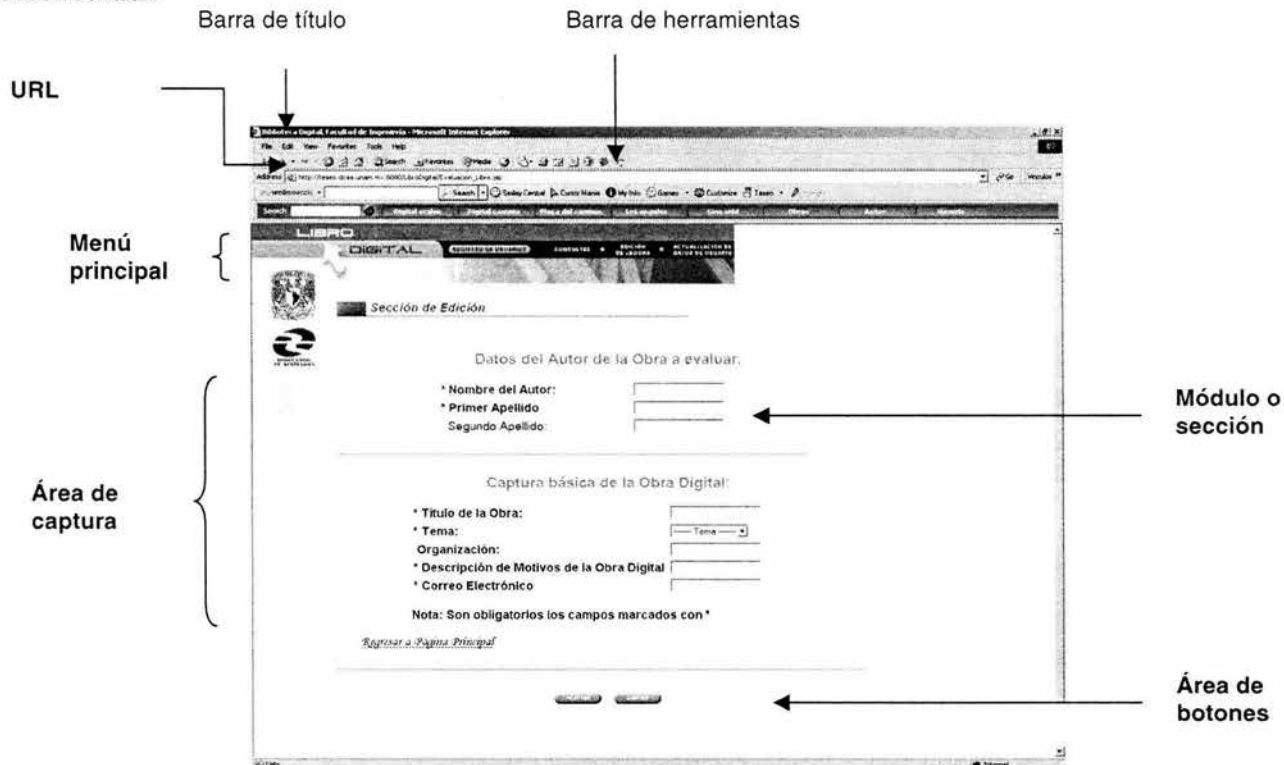
### **Nivel II. Usuarios:**

- PC con procesador mínimo de Pentium I
- Memoria RAM mínima de 64 MB
- Modem con una velocidad de transferencia de 56 Kbps







### 1/ DESCRIPCIÓN GENERAL DE LA PANTALLAS

#### 4.1 Área de captura y manejo de datos

La mayoría de las pantallas del sistema tienen características comunes entre las cuáles se pueden señalar:



- **Barra de título:** En las pantallas se presenta como encabezado superior izquierdo de todas el nombre del sitio y de la Facultad (Biblioteca Digital, Facultad de Ingeniería), seguido por el nombre de la caja solidaria activa, y el perfil ingresado en el inicio de sesión.
- **Barra de herramientas.** Es aquélla donde se engloba aquellos botones ya predefinidos por el browser de los cuales los más comunes son:

Botón	Descripción
	Se regresa una pantalla anterior a la actual
	Se adelanta a una pantalla ya visitada anteriormente
	Detiene la acción actual
	Actualiza o refresca la página actual
	Regresa a la página determinada en su browser como de inicio
	Imprime el contenido de la pantalla actual.

- **URL:** (Uniform Resource Locator). Es la sección donde se introduce la dirección electrónica donde se ubica el sistema.

- **Menú principal:** En esta sección de la pantalla se presenta en forma de listas desplegables las opciones principales y secundarias del sistema (Submenús) y éstos se bloquean de acuerdo al rol del usuario.
- **Módulo o Sección:** Área que identifica la sección del módulo en que se encuentra en uso.
- **Área de captura:** Área donde se presenta la información correspondiente a cada una de las secciones que componen el módulo, permitiendo tanto la captura, como la consulta de la misma según la acción solicitada por el usuario.
- **Área de botones:** Área donde se presentan los botones que permiten al usuario la acción deseada.

## 4.2 CONTROLES DE CAPTURA

Dentro de las pantallas que componen al sistema, existen diversos objetos o controles que permiten la captura y consulta de la información y que tienen las características que a continuación se mencionan.


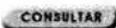



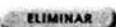
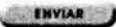



<p><b>Cuadros de edición:</b> Son objetos de captura que permiten al usuario registrar la información en él solicitada.</p>	<p>* Nombre del Autor: <input type="text"/></p>
<p><b>Ligas:</b> Las ligas permiten que el usuario navegue a otra área del sitio.</p>	<p><a href="#"><u>Regresar a Página Principal</u></a></p>
<p><b>Listas desplegables:</b> Éste control u objeto de captura muestra una serie de valores (únicos) que puede tomar un campo específico.</p>	<p><input type="text" value="Tema"/></p>
<p><b>Cuadro de selección única:</b> En este tipo de control u objeto de captura, se presentan un conjunto de opciones posibles de las cuales el usuario sólo podrá elegir una.</p>	<p>Sí <input type="radio"/> No: <input type="radio"/></p>
<p><b>Botones:</b> Este objeto realiza una acción determinada como: limpiar, aceptar, buscar, entre otras.</p>	<p><input type="button" value="LIMPIAR"/></p>

## 4.3 ÁREA DE BOTONES

Dentro de las pantallas que componen al sistema, existen diversos botones generales con los que cuenta, que a continuación se mencionan:













Botón	Acción que realiza	
<p><input type="button" value="ACEPTAR"/></p>	<p>Después de agregar:</p>	<p>Confirma la captura del registro, muestra un mensaje de confirmación.</p>



Botón	Acción que realiza	
	Después de modificar:	Realiza la modificación de un registro, muestra un mensaje de confirmación.
	Vacía los campos en los que se puede introducir información.	
	Encuentra ciertos registros de la información almacenada. El sistema solicita parámetros opcionales para realizar la búsqueda.	
	Después de ingresar los datos.	Permite capturarlos en la base de datos.
	Después de buscar:	Muestra la información de los criterios buscados.
	Permite regresar a la página previa.	
	Permite crear una página de un libro digital.	
	Permite la eliminación de un registro, muestra un mensaje de confirmación y posteriormente un mensaje de eliminación.	
	Permite enviar una petición al servidor de una solicitud.	
	Permite realizar una búsqueda de imágenes para poder agregar una imagen en la portada del libro o en una página específica.	
	Permite finalizar la sesión del cualquier tipo de usuario.	
	Permite almacenar en la base de datos la información	

#### 4.4 MENSAJES DEL SISTEMA

Dentro de las pantallas que componen al sistema, existen mensajes generales con los que cuenta, que a continuación se mencionan:

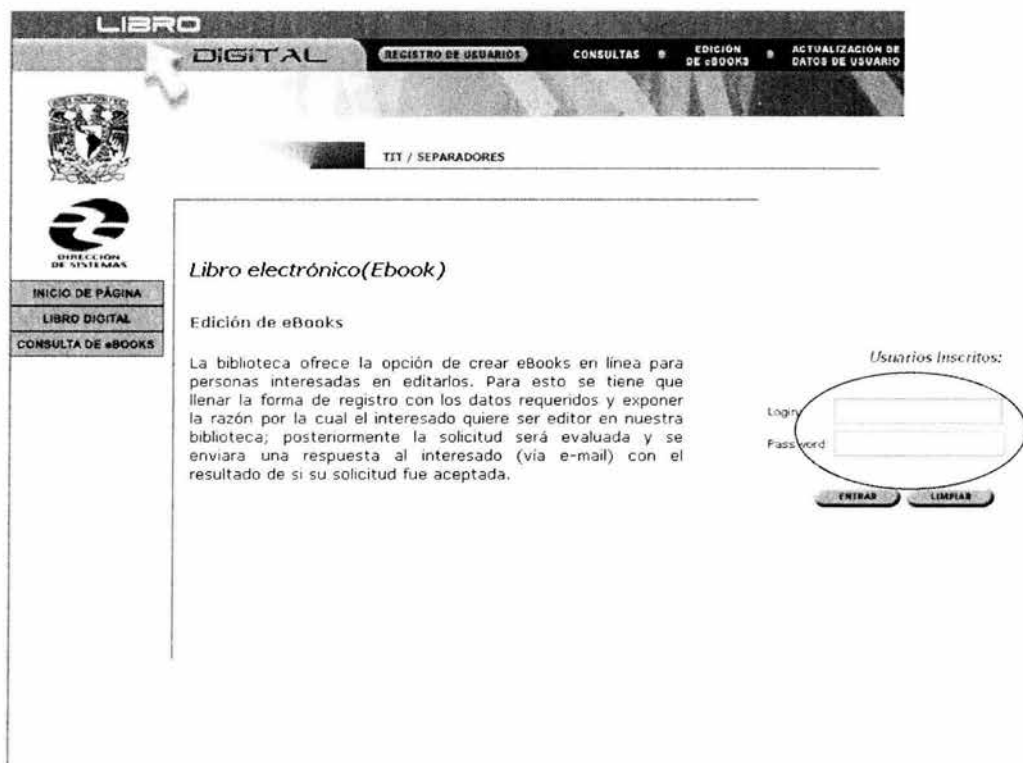
<b>Mensaje</b>	
	La transacción no fue realizada correctamente.
	Usuario no Válido!
	No es posible recargar la página.
	La transacción se realizó exitosamente.
	Asegúrese que todos los campos marcados con * estén llenos.
	Este campo sólo puede contener letras. Por favor corrija su contenido.
	El año que introdujo no es válido.
	Seleccione un elemento de la lista.
	Tipo de Archivo no válido. Sólo se aceptan imágenes formato jpg.
	Haz intentado ingresar un elemento vacío, por favor corrige su contenido.
	No se permite copiar el contenido de la obra digital
	Usted ha rebasado el número de publicaciones que le son permitidas!

### 4.5 INICIO DE SESIÓN

Para entrar al sistema de Libro Digital será necesario realizar los siguientes pasos:

- Especifique en su navegador en la sección de dirección o URL la dirección del servidor en donde se encuentra instalado el sistema.
- El sistema presentará la pantalla de Autenticación donde se debe de ingresar su clave de usuario y contraseña, posteriormente presione el botón

 para ingresar al Menú Principal.



The screenshot displays the 'LIBRO DIGITAL' web application interface. At the top, there is a navigation bar with the title 'LIBRO DIGITAL' and several menu items: 'REGISTRO DE USUARIO', 'CONSULTAS', 'EDICIÓN DE e-BOOKS', and 'ACTUALIZACIÓN DE DATOS DE USUARIO'. Below the navigation bar, there is a logo on the left and a section titled 'Libro electrónico (Ebook)'. The main content area is titled 'Edición de eBooks' and contains a paragraph of text explaining the service. To the right of the text, there is a login form with two input fields labeled 'Login' and 'Pass word', and two buttons labeled 'ENTRAR' and 'LIMPIAR'.

**LIBRO DIGITAL**

REGISTRO DE USUARIO    CONSULTAS    EDICIÓN DE e-BOOKS    ACTUALIZACIÓN DE DATOS DE USUARIO

TIT / SEPARADORES

**LIBRO DIGITAL**

DIRECCIÓN DE SISTEMAS

INICIO DE PÁGINA

LIBRO DIGITAL

CONSULTA DE eBOOKS

*Libro electrónico (Ebook)*

Edición de eBooks

La biblioteca ofrece la opción de crear eBooks en línea para personas interesadas en editarlos. Para esto se tiene que llenar la forma de registro con los datos requeridos y exponer la razón por la cual el interesado quiere ser editor en nuestra biblioteca; posteriormente la solicitud será evaluada y se enviara una respuesta al interesado (via e-mail) con el resultado de si su solicitud fue aceptada.

*Usuarios Inscritos:*

Login:

Pass word:

ENTRAR    LIMPIAR

## DEFINICIÓN DE ROLES

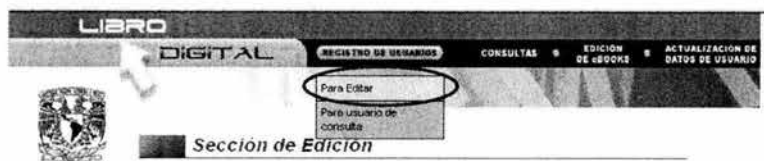
### 5.1 Administrador

El usuario de administración tiene acceso a todos los módulos del sistema, es capaz de agregar, modificar o eliminar obras digitales, usuario de consulta, edición y así mismo generar reportes de bitácoras para la gestión del sitio. Este rol no es posible darlo de alta desde el navegador, se realizará directamente en el sistema operativo.

### 5.2 Edición

El usuario de edición tiene permiso para crear y editar obras digitales hasta un máximo de tres, y con un periodo de expiración. Así mismo puede consultar obras que estén completamente terminadas con los debidos permisos asignados por el autor.

Para dar de alta a un usuario de este tipo primeramente se deberá posicionar el cursor del ratón en el menú **Registro de Usuarios** y al desplegarse el submenú se debe hacer clic en *Para Editar*.



Se desplegará un formulario para que el usuario introduzca sus datos personales:

Este sistema de información para el usuario  
de datos de los estados y municipios de  
México y sus municipios.

### Login y Password de Editor:

\* Login:   
\* Password:   
\* Confirmar Password:

---

### Información del usuario:

\* Nombre(s):  \*Apellido Paterno:   
Apellido Materno:

---

### Datos personales:

\* Pais:   
Dirección:   
C.P.:   
Nivel de estudios:   
Ocupación:   
Organización:   
Teléfono:   
e-mail:   
Solicitud:

[Regresar a Página Principal](#)

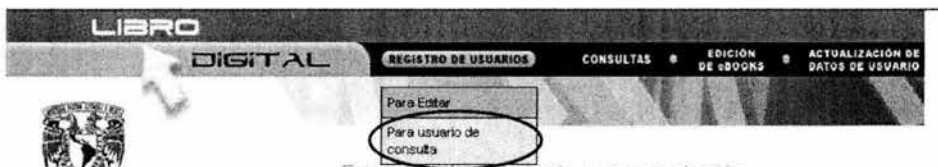
- 🔗 Login
- 🔗 Password
- 🔗 Nombre(s)
- 🔗 Apellido Paterno
- 🔗 Apellido Materno
- 🔗 Pais
- 🔗 Dirección
- 🔗 C.P.
- 🔗 Nivel de Estudios
- 🔗 Ocupación
- 🔗 Organización
- 🔗 Teléfono
- 🔗 e-mail
- 🔗 Solicitud

En el campo de solicitud se deberán exponer los motivos del interés de editar su libro en el sitio. El usuario será dado de alta pero no podrá introducir ninguna obra digital, este proceso lo podrá relizar sólo cuando exista una solicitud para editar una obra digital. (Ver sección VII).

### 5.3 Consulta

Este tipo de usuario sólo puede consultar las obras digitales.

Para dar de alta a un usuario de este tipo primeramente se deberá posicionar el cursor del ratón en el menú **Registro de Usuarios** y al desplegarse el submenú se debe hacer clic en *Para Usuario de Consulta*.



Se desplegará un formulario igual que el de edición para introducir los datos personales del usuario.

## VI. DESCRIPCIÓN DEL PROCESO DE CONSULTA

### 6.1 CONSULTA DE LIBROS DIGITALES

Cualquier usuario que requiera saber si existe algún libro digital en el sitio podrá realizar una consulta sencilla introduciendo ya sea el título de la obra o el autor de ésta.

LIBRO DIGITAL

REGISTRO DE USUARIOS CONSULTAS EDICIÓN DE LIBROS ACTUALIZACIÓN DE DATOS DE USUARIO

IT / SEPARADORES

Búsqueda por palabras:

Título: Autor:

Por Tema:

Seleccionar

CONSULTAR LIMPIAR

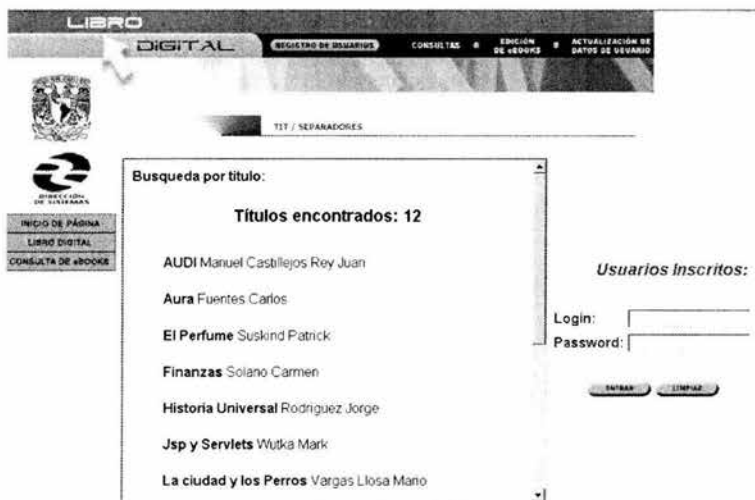
Usuarios Inscritos:

Login:

Password:

ENTRAR OLVIDAR

Si se requiere realizar una búsqueda de todas las obras contenidas en el sitio, sólo se tiene que hacer clic en el botón CONSULTAR y enseguida aparecerán los resultados de las obras utilizando el criterio seleccionado.

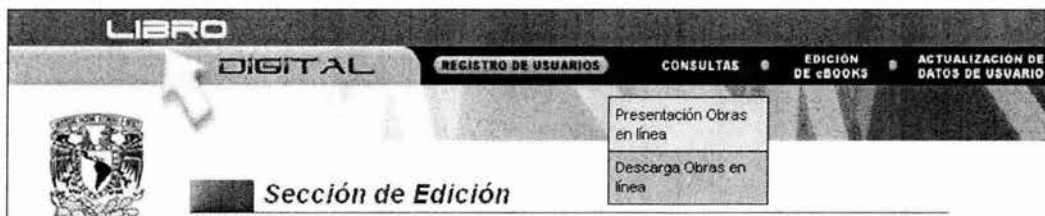


The screenshot shows the 'LIBRO DIGITAL' website interface. At the top, there is a navigation bar with the following items: 'LIBRO DIGITAL', 'REGISTRO DE USUARIOS', 'CONSULTAS', 'EDICIÓN DE eBooks', and 'ACTUALIZACIÓN DE DATOS DE USUARIO'. Below the navigation bar, there is a search box with the text 'Busqueda por título:' and 'Títulos encontrados: 12'. The search results are listed as follows:

- AUDI Manuel Castillejos Roy Juan
- Aura Fuentes Carlos
- El Perfume Suskind Patrick
- Finanzas Solano Carmen
- Historia Universal Rodriguez Jorge
- Jsp y Servlets Wulka Mark
- La ciudad y los Perros Vargas Llosa Mario

On the right side of the search results, there is a section for 'Usuarios inscritos:' with a 'Login:' field, a 'Password:' field, and two buttons labeled 'ENTRAR' and 'LIMPIAR'.

Se pueden realizar consultas avanzadas siempre y cuando se tenga una cuenta de edición, consulta o administrador. Una vez que se ingrese al sistema como cualquiera de estos usuarios se posiciona el puntero del ratón en la sección del menú **Consultas**, enseguida se despliega un submenú y se debe de hacer clic en **Presentación Obras en línea**.



The screenshot shows the 'LIBRO DIGITAL' website interface with the 'CONSULTAS' menu item highlighted. A dropdown menu is displayed, showing the following options:

- Presentación Obras en línea
- Descarga Obras en línea

Below the dropdown menu, there is a section labeled 'Sección de Edición'.

Posteriormente aparece una pantalla que contiene los diferentes criterios de búsqueda para realizar la consulta.



**LIBRO DIGITAL** REGISTRO DE USUARIOS CONSULTAS EDICIÓN DE EBOOKS ACTUALIZACIÓN DE DATOS DE USUARIO

**Sección de Edición**

Libro Digital:

Busqueda por campos:

Por Título:

Por Nombre del Autor: Mark por Apellido: Wutka

Por Año:

Por Editorial: Seleccionar Por Tema: Computación

Por ISBN:

CONSULTAR LIMPIAR

Se selecciona el criterio de búsqueda y debe hacer clic en el botón Consultar. Los resultados se desplegarán y se podrá hacer clic en cualquiera de las obras digitales que aparezcan como resultado de la consulta.

**LIBRO DIGITAL** REGISTRO DE USUARIOS CONSULTAS EDICIÓN DE EBOOKS ACTUALIZACIÓN DE DATOS DE USUARIO

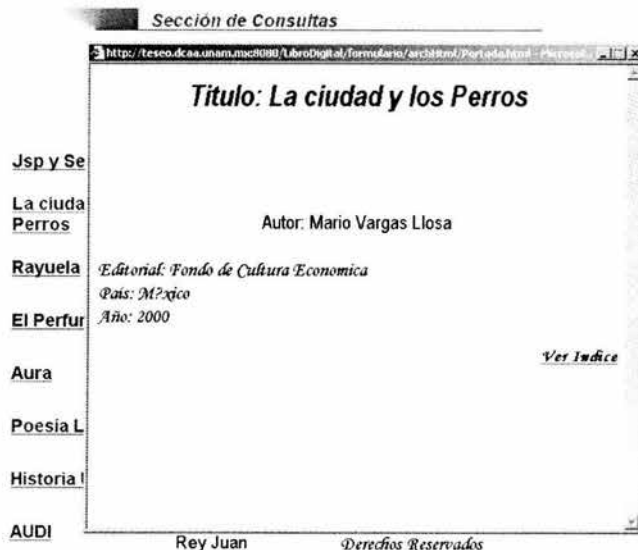
**Sección de Consultas**

Títulos encontrados: 12

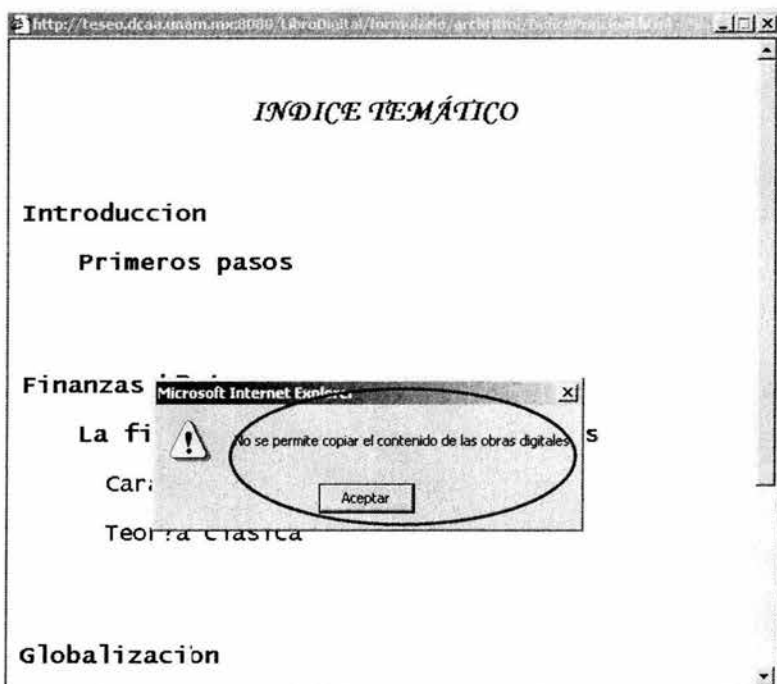
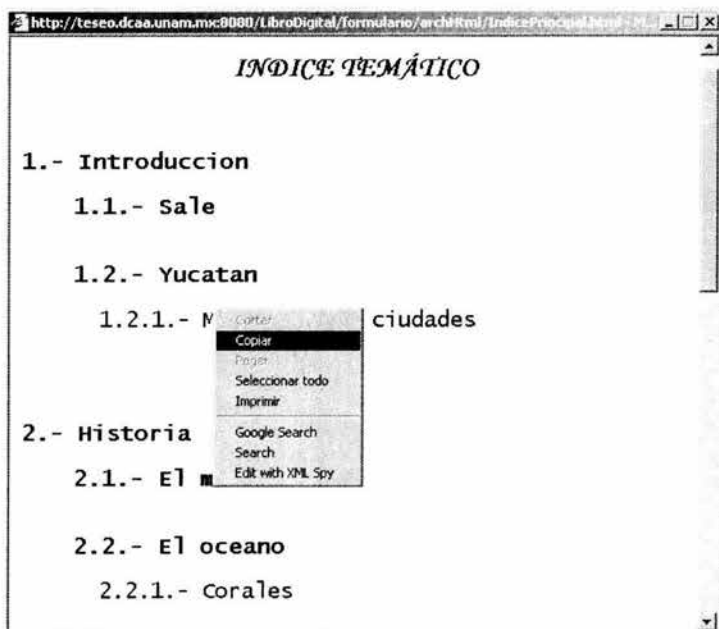
<a href="#">Jsp y Servlets</a>	Wutka Mark	México 2003 UNAM Derechos Reservados
<a href="#">La ciudad y los Perros</a>	Vargas Llosa Mario	México 2000 UNAM Derechos Reservados
<a href="#">Rayuela</a>	Cortazar Julio	México 1963 Derechos Reservados
<a href="#">El Perfume</a>	Suskind Patrick	México 1995 Derechos Reservados
<a href="#">Aura</a>	Fuentes Carlos	México 1945 Derechos Reservados
<a href="#">Poesía Literaria</a>	Sabines Jaime	México 1998 UNAM Derechos Reservados
<a href="#">Historia Universal</a>	Rodriguez Jorge	México 2000 Derechos Reservados
<a href="#">AUDI</a>	Manuel Castillejos Rey Juan	México 2002 UNAM Derechos Reservados

Se desplegará una ventana la cual mostrará:

- Título de la Obra
- Autor
- Editorial
- País
- Año



Si la obra es de interés se puede hacer clic en la leyenda "Ver Índice" y se desplegará el índice temático, y dependiendo de los derechos que el autor haya establecido se podrá o no copiar el contenido haciendo clic derecho y dando la opción copiar.



## VII. DESCRIPCIÓN DEL PROCESO DE EDICIÓN

El proceso de edición sólo podrá ser realizado por un usuario previamente registro como de este tipo. Al ingresar al sitio se desplegará una pantalla con la siguiente leyenda:

*Bienvenido: Nombre del usuario*

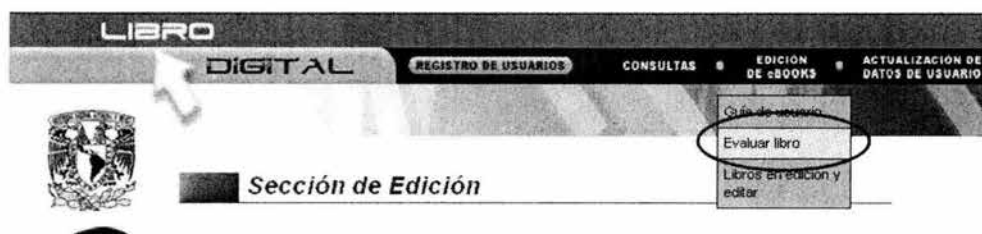
Usted ha sido aceptado por el comité de edición. Por tanto se le ha asignado una cuenta válida por un período de tiempo limitado que le permitirá elaborar una obra digital bajo los respectivos derechos de autor que usted mismo establezca.

**Atención:** La información que usted ingrese será bajo su propia responsabilidad. *El comité se reserva el derecho de publicación digital en este portal.*

---

México

Cuando se quiere comenzar a editar un libro es necesario ingresar al sitio una solicitud de Evaluación de un Libro para que sea evaluado por el comité. Esto se realiza posicionando el ratón en el menú de Edición de eBooks, y posteriormente se da clic en el submenú *Evaluar Libro*.



En esta sección se debe introducir los datos del autor:

- ☞ Nombre
- ☞ Primer Apellido
- ☞ Segundo Apellido

Se introducen los datos generales de la Obra Digital como son:

Título

Tema

Organización

***Descripción de Motivos de la Obra*** (Este campo es de suma importancia para la aceptación de la obra en el sitio por el Comité de éste). Se debe describir las características del libro (tipo de contenido, abstract y un aproximado de la cantidad de hojas).

Se pide introducir un correo electrónico en donde posteriormente se notificará al usuario si la obra es aceptada o no en un plazo de uno a dos días. De ser aceptada se mandará una clave de acceso para que pueda comenzar la edición del libro, con un tiempo de expiración el cual se determinará de acuerdo a lo descrito en el campo de Descripción por el autor. Es decir si éste expone que el libro que desea publicar en el sitio es extenso, el comité determinará el tiempo que crea conveniente para que la cuenta esté activa.

**LIBRO DIGITAL**

REGISTRO DE USUARIOS   CONSULTAS   EDICIÓN DE eBOOKS   ACTUALIZACIÓN DE DATOS DE USUARIO

**Sección de Edición**

Datos del Autor de la Obra a evaluar:

\* Nombre del Autor:

\* Primer Apellido

Segundo Apellido:

Captura básica de la Obra Digital:

\* Título de la Obra:

\* Tema:

Organización:

\* Descripción de Motivos de la Obra Digital

\* Correo Electrónico

Nota: Son obligatorios los campos marcados con \*

[Regresar a Página Principal](#)

ACEPTAR   LIMPIAR

Esta cuenta será única para dicho libro, y como se mencionó en el punto 4.6.2 solo se pueden tener tres, y este dato aparece cuando se inicia la sesión al sitio como editor y se va a la sección de *Libros en edición y editar*.

**LIBRO DIGITAL**

REGISTRO DE USUARIOS   CONSULTAS   EDICIÓN DE eBOOKS   ACTUALIZACIÓN DE DATOS DE USUARIO

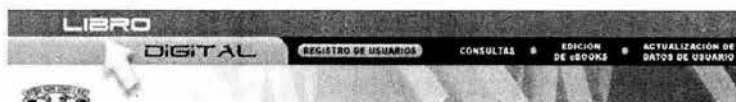
**Sección de Edición**

- Guía de usuario
- Evaluar libro
- Libros en edición y editar**

Y aparece la siguiente pantalla en la cual se puede seguir editando una obra ya comenzada o editar una nueva.



Si ya se tienen tres obras en edición el sistema no permitirá editar otra obra mas lanzando un mensaje de error "*Usted ha revasado el número de publicaciones que le son permitidas!*"

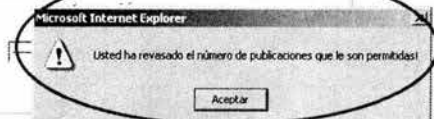


Sección de Edición



*Comenzar o continuar una edición:*

Obras Digitales que actualmente tiene usted en

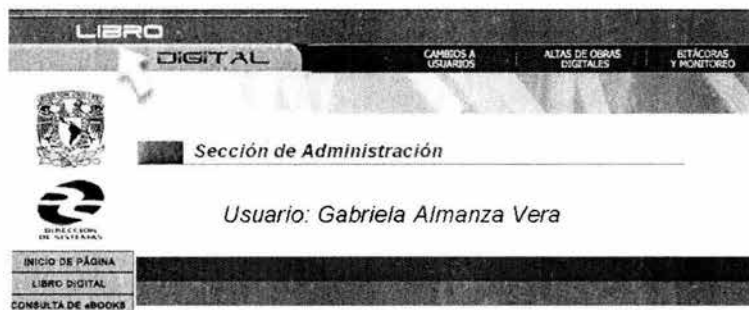


[Finalizar la sesión](#)



## VIII. MÓDULO DE ADMINISTRACIÓN

El rol definido como de administrador podrá efectuar las tareas de **Cambios a Usuario**, **Altas de Obras Digitales** y realizar las **Bitácoras y Monitoreo**.



### 8.1 CAMBIOS A USUARIOS

En esta sección se podrán realizar cambios de los datos personales de los usuarios, como son login, nombre(s), apellidos, etc.

### 8.2 ALTAS DE OBRAS DIGITALES

Aquí es donde se realizan las altas de las obras digitales para que el usuario de edición comience la captura de la obra.

**Nota.- Las obras deben ser aprobadas por el comité del sitio antes de ser dadas de altas.**

### 8.3 BITÁCORAS Y MONITOREO

En esta parte se puede monitorear y solicitar bitácoras de los diferentes usuarios, libros consultados, cantidad de libros editados.

---

## Glosario de Términos

- ACTIVE-X** Tecnología de Microsoft que facilita el uso de información compartida entre aplicaciones. Se utiliza principalmente para desarrollar aplicaciones interactivas y contenido de Web. ActiveX se ha construido sobre la tecnología OLE que se utilizó durante algún tiempo, pero expande el alcance de los objetos compartidos desde el escritorio a todo Internet. Debido a que la tecnología ActiveX es modular en cuanto al diseño, los programas pueden escribirse como aplicaciones independientes, como "objetos inteligentes" incrustados dentro de programas Visual Basic o páginas Web, o como objetos OLE tradicionales dentro de los documentos.
- ACM** *Association of Computer Machinery.* Asociación de Maquinaria Computarizada, es una de las mas grandes organizaciones educacionales en materia de computadoras y tecnología, esta dividido en diferentes grupos de interés y organiza diferentes actividades.
- APACHE** El Proyecto del Servidor Apache HTTP constituye un esfuerzo de desarrollo y mantenimiento de un servidor HTTP de código abierto para todos los sistemas operativos, como pueden ser UNIX y Windows Server. Basado en un modelo cliente-servidor multiproceso es un servidor extremadamente configurable, con soporte para hosts virtuales y con mucha capacidad de crecimiento por medio de módulos que le proporcionan nuevas funcionalidades.
- API** *Application Program Interface.* Interfaz de Programa de aplicación, parte del sistema operativo que provee a las aplicaciones una interfaz de uso común.
- APPLET** Programa escrito en lenguaje Java, diseñado para ser ejecutado desde otras aplicaciones.
- ARPA** *Advanced Research Projects Agency.* Agencia de Proyectos de investigación avanzada de defensa de los Estados Unidos.
- ARPANET** *Advanced Research Projects Agency Network.* Considerada la precursora del Internet.
- ASCII** *American Standard Code for Information Interchange.* Código estándar Americano para el intercambio de información.
- ASP** *Active Server Pages.* Es una especificación para páginas Web dinámicas con extensión .ASP que utiliza ActiveX. Cuando un navegador realiza una petición ASP, el servidor Web genera una página con código HTML y lo envía de regreso al navegador, las aplicaciones ASP son similares a las construidas bajo CGI.
- ATM** *Asynchronous Transfer Mode.* Modo de Transferencia Asíncrona, es una tecnología de red basada en la transferencia de datos en paquetes de un tamaño variable. Este tipo de paquetes son relativamente pequeños comparados con otras tecnologías mas antiguas.

- BITNET** Red de comunicaciones de las universidades del mundo. BITNET es una de las redes más antiguas y largas usada extensivamente por universidades. Una nueva versión de BITNET, llamada BITNET-II incorpora la transferencia de mensajes y archivos.
- Broker** Empresa de servicios de información que se dedica a localizar, recuperar y ofrecer información sobre cualquier tema de interés.
- BSD** *Berkeley Software Design*. Software comercial de Internet y redes basado en una versión de BSD de Unix.
- B2B** *Business to Business*. Normalmente utilizado para describir un negocio electrónico que provee servicios a otro nicho de negocio en la red.
- CERN** *Conseil European pour le Recherche Nucleaire*. Laboratorio europeo de física de partículas. Es un laboratorio de investigación ubicado en Ginebra, Suiza, con presencia en distintos países.
- CGI** *Common Gateway Interface*. Especificación para transferencia de información entre un servidor de aplicaciones en Web y un programa CGI.
- CPU** Unidad de Procesamiento Central. Es el cerebro de la computadora, donde se realizan el procesamiento de las operaciones de la máquina.
- CSS** *Cascading Style Sheets*. Una nueva característica agregada a HTML que proporciona a los desarrolladores de sitios y a sus usuarios un mejor control para el despliegado de páginas. Mediante las hojas de estilo CSS, los diseñadores y usuarios pueden crear documentos de estilo que definen la apariencia de los encabezados, ligas, así como presentación en general.
- DCL** *Data Control Language*. El Lenguaje de control de datos es empleado para asignar privilegios a usuarios sobre los objetos de una base de datos.
- DDL** *Data Definition Language*. El Lenguaje de Definición de Datos se ocupa para definir objetos dentro de una base de datos.
- DHTML** *Dynamic HTML*. Se refiere a los cambios que sufre el contenido de las páginas Web cada vez que son visualizadas.
- DLL** *Dynamic Link Library*. Archivo que almacena funciones ejecutables o datos que pueden ser usados por una aplicación en Windows.
- DML** *Data Manipulation Language*. Es el lenguaje de Manipulación de Datos. Se define como un conjunto de instrucciones usada para almacenar, extraer, modificar y borrar datos de una base de datos. Existen dos tipos de DML: procedural en éste, el usuario especifica el dato que requiere y su extracción; y el lenguaje no procedural, que indica al usuario el dato que necesita.
- DNS** *Domain Name System*. Es un servicio de Internet que traduce el nombre del dominio a una dirección IP.

- DTD** *Document Type Definition*. Definición de Tipo de Documento. Se describe por medio de etiquetas y atributos, define el contenido de un documento de tipo SGML, XML o HTML.
- DUBLIN CORE** La iniciativa Dublin Core de Metadatos (DCMI) Es una organización dedicada al desarrollo y adopción de estándares interoperables de metadatos y al desarrollo de vocabularios de metadatos especializados para la descripción de fuentes que permitan el desarrollo de sistemas de información inteligentes. <http://dublincore.org>
- EBCDIC** Código para representar caracteres como números, parecido al ASCII.
- EDI** *Electronic Data Interchange*. La transferencia de datos entre diferentes compañías ocupando Internet.
- EDVAC** *Electronic Discrete Variable Automatic Computer*. Primer computadora Digital que ocupaba un programa almacenado.
- ENIAC** *Electronic Numerical Integrator And Calculator*. La primera computadora electrónica digital operacional desarrollada por la armada en la segunda guerra mundial.
- ETHERNET** Protocolo para redes de área local. Una arquitectura desarrollada por la corporación de Xerox con la cooperación de DEC e Intel en 1976. Ethernet ocupa una topología de bus o estrella y soporta una transferencia de datos de 10 Mega bits por segundo.
- FTP** *File Transfer Protocol*. Protocolo para la transferencia de archivos.
- GML** *General Markup Language*. Lenguaje basado en XML para codificar información geográfica para poder ser almacenada y transportada en Internet.
- GUI** *Graphic User Interface*. Programa de interfaz que toma como ventaja las capacidades de los gráficos realizados por las computadoras para poder facilitar el uso de aplicaciones de desarrollo.
- GXA** *Web Service Architecture*. Arquitectura para servidores Web que se encuentra basado en SOAP, (Protocolo de acceso a objetos simple).
- HTML** *HyperText Markup Language*. Lenguaje marcado de hipertexto.
- HTTP** *HyperText Transfer Protocol*. (Protocolo de Transferencia de Hipertexto). Es la base del protocolo usado en el World Wide Web. HTTP define el formato de los mensajes así como su transmisión, y las acciones que deben tomar los servidores Web y navegadores para responder a varios comandos.
- IBM** *International Business Machines*. Es la compañía mas larga de computadoras del mundo, comenzó en 1911 como productora de tarjetas de máquinas en forma de tablas.
- 
-

- IDE** *Integrated Drive Electronics ó Intelligent Drive Electronics or Integrated Drive Electronics.* Es una interfaz IDE es para el almacenamiento de dispositivos en el cual el controlador está integrado en el disco de almacenamiento.
- IDL** Es una tecnología para objetos distribuidos que interactúan sobre distintas plataformas a través de una red. Java IDL es similar a RMI (*Remote Method Invocation*), que soporta objetos distribuidos escritos enteramente en Java. Sin embargo, Java IDL permite interactuar con los objetos sin importar si están escritos en Java o en cualquier otro lenguaje como C, C++, Perl por mencionar algunos. Para soportar la interacción entre objetos de programas separados, Java IDL proporciona una biblioteca de clases que permite una comunicación de bajo nivel entre aplicaciones Java IDL y aplicaciones compatibles con CORBA.
- IIS** *Internet Information Server.* Servidor Web de Microsoft que se ejecuta bajo plataformas Windows NT y 2003.
- ISBN** *International Standard Book Number.* Estándar internacional de numeración para libros.
- INTERNET** *Interconnected Network.* Es una red global que conecta a millones de computadoras. Más de 100 países están vinculados por medio de esta red para intercambiar datos, noticias y opiniones.
- Una red de redes a escala mundial de millones de computadoras. Al contrario de lo que se piensa comúnmente, Internet no es sinónimo de World Wide Web. Ésta es parte de aquella, siendo la World Wide Web uno de los muchos servicios ofertados en la red Internet. La Web es un sistema de información mucho más reciente (1995) que emplea la red Internet como medio de transmisión. Algunos de los servicios disponibles en Internet aparte de la Web son el acceso remoto a otras máquinas (telnet y ssh), transferencia de archivos (FTP), correo electrónico (e-mail), boletines electrónicos (news o grupos de noticias), conversaciones en línea (chat), mensajería instantánea, etcétera.
- IP** *Internet Protocol.* La IP especifica el formato de paquetes, también llamados datagramas, y le esquema de direcciones.
- ISO** *International Standard Organization.* Conformada por estándares nacionales de mas de 75 países.
- ISP** Proveedor de Servicios de Internet
- ITESM** Instituto Tecnológico de Estudios Superiores de Monterrey.
- Jakarta Tomcat** El Proyecto Jakarta es desarrollado por la *Apache Software Foundation* (ASF), crea y mantiene soluciones de fuente libre que se implementan sobre la plataforma Java para una distribución pública no lucrativa. Los productos Jakarta son desarrollados y distribuidos a través de varios subproyectos que se mantienen en constante evolución.
- JIT** *Just In Time.* Es un generador de código que convierte el código de java (*byte code*) a instrucciones en lenguaje máquina.

- JSP** *Java Server Pages.* Es una tecnología implementada del lado del servidor, es una extensión de los servlets. Esta tecnología fue también desarrollada por Sun Microsystems.
- JVM** *Java Virtual Machine.* Es una plataforma independiente de la plataforma que convierte el bytecode en lenguaje máquina y lo ejecuta.
- MIDDLEWARE** Es la capa de software que existe entre la red y las aplicaciones que bajo ésta se ejecutan. Bajo esta capa se proporcionan servicios como autenticación, identificación, autorización así como seguridad. Esta capa es muy importante para la estandarización, interoperabilidad y compatibilidad de las aplicaciones empresariales en Internet.
- MSDOS** *Microsoft Disk Operating System.* Sistema operativo desarrollado por Microsoft e implementado en todos sus equipos.
- NIC** *Network Interface Card.* Tarjeta de expansión que se inserta en la computadora para que ésta se pueda conectar a una red.
- NCP** *Network Control Protocol.* Protocolo de control de red.
- NSF** *Network File System.* Sistema de archivos de red. Se desarrolló para permitir montar una partición perteneciente a una máquina remota como si fuera una partición local. Proporciona un método eficaz para compartir archivos.
- NSFNET** *National Science Foundation Network.* Comenzó con una serie de redes dedicadas a la comunicación de la investigación y de la educación. Fue creada por el gobierno de los Estados Unidos, y fue reemplazada por ARPANET.
- NVP** *Network Voice Protocol.* Fue implementado primera vez en Diciembre de 1973 y ha sido usado desde entonces para comunicaciones locales y remotas en tiempo real de voz.
- PDA** *Personal Digital Assistant.* Dispositivos o asistentes personales son computadoras de bolsillo comenzaron como organizadores personales, actualmente ofrecen una gama amplia de opciones para software y hardware, conectividad en Internet, cargan instantáneamente y ocupan una considerada cantidad de funciones y procesos. El primer PDA fue el Newton de Apple Computing que en principio fungía como agenda electrónica, funcionaba con un lápiz con el cual se escribía en pantalla.
- PHP** *Hypertext Preprocessor.* Es un lenguaje de programación concebido en otoño de 1994 por Rasmus Lerdorf. Se utiliza principalmente para la programación de CGI para páginas web, destaca por su capacidad de ser embebido en el código HTML.
- PSGML** Modo para editar ficheros sgml/html en emacs. Las versiones mas recientes (1.2.1-8 o superior) soportan mejor el código XML.

- MIT** *Massachussets Institute of Technology.* Instituto de Tecnología de Massachussets de los Estados Unidos de América.
- RDBMS** *Relational Data Base Manager System.* Sistema administrador de Bases de Datos Relacionales. Los RDBMS proporcionan el ambiente adecuado para gestionar una o más bases de datos.
- RMI** *Remote Method Invocation.* El sistema de Invocación Remota de Métodos (RMI) de Java permite a un objeto que se está ejecutando en una Máquina Virtual Java (VM) llamar a métodos de otro objeto que está en otra VM diferente. RMI proporciona comunicación remota entre programas escritos en Java. Si uno de nuestros programas está escrito en otro lenguaje, deberemos considerar la utilización de IDL en su lugar.
- RPC** *Remote Procedure Call.* El RPC de Sun es un protocolo propuesto como estándar. Su status es electivo. Es una interfaz de programación de aplicación (API) disponible para el desarrollo de aplicaciones distribuidas. Permite que los programas llamen a subrutinas que se ejecutan en un sistema remoto. El programa llamador, denominado (llamado cliente) envía un mensaje de llamada al servidor y espera por un mensaje de respuesta. La llamada incluye los parámetros del procedimiento y la respuesta los resultados.
- RTF** *Rich Text Format.* RTF es uno de los formatos nativos del Microsoft Word. Teóricamente, todas y cada una de sus opciones son representables en este formato. Teniendo en cuenta que las opciones de Word se han ido añadiendo a lo largo de los años, el RTF es un formato común pero no es muy ocupado actualmente.
- SATNET** Red enlazada de satélites ocupada para la automatización de acceso telefónico.
- SGML** *Standard Generalized Markup Language* o Lenguaje de Etiquetado Generalizado Normalizado. SGML es un estándar internacional, no propietario y abierto, que provee un método para la descripción de la estructura de documentos basándose en la relación lógica de sus partes. Provee una codificación estándar para la transmisión de documentos entre sistemas de computadoras diferentes: distintas plataformas, soportes físicos, lógicos y diferentes sistemas de almacenamiento y presentación, con independencia de su grado de complejidad.
- SMTP** *Simple Mail Transfer Protocol.* Este protocolo es un estándar de Internet para el intercambio de correo electrónico. Al momento de enviar un correo electrónico se ocupa como medio un servidor SMTP que es encargado de hacer llegar el correo a su destino, se puede comparar con el servicio postal, para hacer entrega del correo necesitamos de tres datos importantes el origen, el destino y el medio que es el servidor SMTP.
- SOAP** *Simple Object Access Protocol.* Es un protocolo utilizado para facilitar la llamada remota de funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web.

- SQL** *Structured Query Language.* El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos, se utiliza como interfaz para que los usuarios o las aplicaciones puedan interactuar con la base de datos.
- SWING** Es un paquete que permite incorporar elementos gráficos en las aplicaciones, de una manera más potente que con el AWT. Aparece en la versión 1.2 del JDK. Es no de los componentes que están incluidos en las *Java Foundation Classes*, o JFC.
- TCP/IP** *Transmission Control Protocol.* Es un protocolo que proporciona transmisión fiable de paquetes de datos sobre redes. El nombre TCP/IP Proviene de dos protocolos importantes de la familia, el TCP y el Internet Protocol (IP). Todos juntos llegan a ser más de 100 protocolos diferentes definidos en este conjunto. El TCP / IP es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local y área extensa. TCP / IP fue desarrollado y demostrado por primera vez en 1972 por el departamento de defensa de los Estados Unidos, ejecutándolo en el ARPANET una red de área extensa del departamento de defensa.
- TELNET** *Tele Network.* Conexión a un host en la que una computadora cliente emula una terminal de manera que se configura como terminal virtual del servidor.
- UCLA** University of California.
- UNAM** Universidad Nacional Autónoma de México.
- UNICODE** Código de 16 ligamentos para exponer signos en la computadora, parecido al ASCII pero posee un número mas grande de signos lo que permite el uso alfabético de todos los idiomas mundiales.
- UNIX** Sistema operativo multitarea y multiusuario.
- URI** Identificador Uniforme de Recursos.
- URL** *Uniform Resource Locator.* Dirección global de documentos y otras fuentes en Internet.
- UTSA** Universidad de Texas en San Antonio.
- W3C** El **World Wide Web Consortium** Es el consorcio de la Web (<http://www.w3.org/>), abreviadamente W3C, es una organización que produce estándares para la Web mundial o World Wide Web. Está dirigida por Tim Berners-Lee, el creador original del HTTP (HyperText Transfert Protocol o Protócolo de Transferencia de HiperTexto) y HTML (Lenguaje de Marcado de HiperTexto) sobre el que se basa la Web, y por extensión su inventor.



**WWW** World Wide Web. Es un sistema hipermedia interactivo que junta texto, imágenes, audio y video dentro de un mismo envoltorio llamado documento. La WWW se asienta sobre el protocolo HTTP y es desarrollado bajo el lenguaje de definición de documentos hipermedia HTML. Se basó desde sus inicios en los siguientes componentes: TCP/IP, tipos MIME, SGML, y Arquitecturas Cliente-Servidor así como Sistemas Abiertos.

## Referencias Bibliográficas

- ❖ Akif Mohammad, Bordead Steven y Cioroianu Andrei (2001), *Referencia para programadores Java y XML*, Madrid España, Naya Multimedia, ISBN 84-415-1364-2002.
- ❖ Bobadilla S, José (1999), *Superutilidades para WebMasters*, México, Mc.Graw-Hill.
- ❖ Damon, Hougland y Tavistock, Aaron (2002), *Guía Esencial* Madrid, España Prentice Hall, ISBN: 84-205-3332-7.
- ❖ Date, J. C. (2001), *Bases de Datos*, México, Pearson Education, ISBN: 0321197844.
- ❖ Deitel y Deitel (1998), *Cómo Programar en JAVA*, México, Pearson Education ISBN: 970-17-0044-9.
- ❖ Eckel, Bruce (2002), *Piensa en Java, 2ª Edición*, Editorial Prentice Hall, México, Madrid, España, ISBN: 84-205-3192-8.
- ❖ Pérez, Cesar (2003), *Oracle 9i, Administración y Análisis de Bases de Datos*, Madrid España, Alfaomega Ra- Ma ISBN 970-15-0845-9.
- ❖ Schmidt, Hebert (2001), *JAVA 2 Manual de Referencia*, Madrid, España, Primera edición, Mc Graw Hill.
- ❖ Sommerville Ian (2002), *Ingeniería de Software*, México, Sexta Edición Addison Wesley, ISBN 0-201-39815-X.

- ❖ Williamson, Heather (2001), *Manual de Referencia XML*, México, Mc.Graw-Hill ISBN 84-481-3267-X 2001.

### Referencias electrónicas de importancia

<http://dublincore.org/>

<http://jakarta.apache.org/tomcat/>

<http://www.redhat.com/>

<http://java.sun.com/>

<http://www.softquad.com/>

<http://www.sybase.com/home>

<http://www.xmetal.com/>