



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN**

**"PUBLICACIÓN DE LA SÍNTESIS INFORMATIVA DE LA
DELEGACIÓN IZTAPALAPA EN INTRANET CON XML".**

TESINA

**QUE PARA OBTENER EL TÍTULO DE
LIC. MATEMÁTICAS APLICADAS Y COMPUTACIÓN**

PRESENTA

CARLOS ISAIAS BAUTISTA MONROY.

ASESOR: LIC. MARÍA DEL CARMEN VILLAR PATIÑO

FECHA: ABRIL DE 2004.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

***Quiero dedicar este trabajo con
todo mi cariño a las personas más
importantes de mi vida:
a Rosa, Isaías, Isaac y
especialmente
a Marianne.***

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Carlos Bautista

Menroy

FECHA: 30 de abril de 2004

FIRMA: 

AGRADECIMIENTOS

Quiero agradecer profundamente a la lic. Ma. del Carmen Villar, por el tiempo otorgado, los consejos ofrecidos, la paciencia brindada y sobre todo por la confianza para este trabajo.

También agradezco a la Delegación Iztapalapa, en especial a Carmen y Sandra por su ayuda.

Doy gracias a mis amigos, Bere, Israel, Enrique y Juan, por el apoyo y confianza que tuve de su parte dentro y fuera de la Universidad.

Gracias a esa energía que lo puede todo.

ÍNDICE

	Página
INTRODUCCIÓN	5
I. INTERNET	
I.1. Antecedentes	7
I.1.1. La red ARPA	7
I.1.2. Organizaciones Científicas	7
I.1.3. Extensión de la red	7
I.2. Origen de la World Wide Web	8
I.2.1. Historia	8
I.2.2. Intranet y Extranet	11
I.2.3. Características de la WWW	12
I.2.4. W3C	13
I.3. Lenguajes de Marcado	14
I.3.1. Hipertexto	14
I.3.2. Marcado de formato	15
I.3.3. Marcado Generalizado	16
I.4. SGML Standard Generalized Markup Language	17
I.4.1. Historia	17
I.4.2. Características	17
I.4.3. Definición de Tipos de documento	18
I.4.4. Unidades de contenido: las entidades	22
I.5. HTML HyperText Markup Language	23
I.5.1. Definición	23
I.5.2. Cualidades	23
I.5.3. Versiones de HTML	24
I.6. Limitaciones de HTML	26
I.7. HTML su uso con otras aplicaciones	28
I.7.1. CGI (Common Gateway Interface)	28
I.7.2. CSS Style-Sheets	29
I.7.3. JavaScript	29
I.7.4. Shockwave	30
I.7.5. Java	30
I.7.6. Active X	31
I.7.7. VRML	32
I.7.8. XML	32
II. XML	
II.1 Antecedentes	33
II.1.1. Historia	33
II.1.2. Definición	34
II.1.3. Objetivo	35
II.1.4. Importancia	35
II.2 Utilización	36
II.2.1. Editores	36
II.2.2. Navegadores – Browsers	37
II.2.3. Parsers XML	37
II.2.4. Aclaraciones	37

II.3	Estructura y conceptos básicos	38
II.3.1.	Gramáticas y representación	39
II.3.2.	Estructura	40
II.3.3.	Tipos de Documento	43
II.3.4.	Buena formación y validez	44
II.3.5.	Información sintáctica	45
II.4	Relación de XML con SGML y HTML	48
II.4.1.	Presentación – Procesamiento	48
II.4.2.	Paso de SGML a XML	48
II.4.3.	Integración de XML con HTML	49
II.5	XML y las bases de datos	50
II.5.1.	Manipulación de bases de datos en Web	50
II.5.2.	Integración con Manejadores de B.D.	50
II.5.3.	XML como una base de datos	52
II.6	Herramientas de apoyo	54
II.6.1.	DSSSL	55
II.6.2.	DTD – Document Type Definition	59
II.7	Tecnologías derivadas y complementarias a XML	65
II.7.1.	XML Document Object Model y Java	65
II.7.2.	Parser/Dom	65
II.7.3.	XML Schemas	65
II.7.4.	Xlink/Xpointer (Lenguaje de enlace)	66
II.8	Aplicaciones	67
II.8.1.	WEB	67
II.8.2.	EDI – Electronic Data Interchange	69
III.	APLICACIÓN DE LA SÍNTESIS INFORMATIVA	
III.1	Delegación Iztapalapa	73
III.2	Situación actual de la Síntesis Informativa	75
III.2.1.	Definición	75
III.2.2.	Origen y Objetivo	75
III.2.3.	Síntesis Informativa en Intranet	75
III.3	Planteamiento del problema	81
III.3.1.	Puntos de conflicto	81
III.3.2.	Limitaciones con el uso de HTML	81
III.4	XML en el diseño de la Síntesis	85
III.4.1.	Utilización de XML	85
III.4.2.	Planteamiento de la solución	85
III.4.3.	Estructura del Site	89
III.5	Desarrollo de la Síntesis	90
III.5.1.	Generación de la estructura y actualización	90
III.5.2.	Uso del Sistema de Actualización para la página	95
III.5.3.	Descripción del resultado	118
	CONCLUSIONES	131
	BIBLIOGRAFÍA	133

INTRODUCCIÓN

En la actualidad Internet se ha convertido en una herramienta indispensable en la vida comercial, de servicios e incluso de índole personal y ha contribuido al desarrollo de un mejor medio para la distribución de información con fácil acceso.

Uno de los servicios de Internet, la World Wide Web (WWW o simplemente Web), a través de las llamadas "Páginas Web", ha expandido su éxito en la sociedad actual. Dentro de este auge, se difundió rápidamente la idea de crear páginas para intercambiar u ofrecer información con usuarios de la Web, gracias a la facilidad para su creación con el Lenguaje de Marcado de HiperTexto mejor conocido como HTML.

El antecedente para desarrollar la presente tesina es precisamente la influencia y el gusto generado por Internet y el uso de HTML, que ha llegado mas allá de la simple creación de Páginas para la Web. Dentro de éste ámbito, la creación y difusión de Sitios en Internet cada vez más elaborados ha ido creciendo rápidamente y de la misma forma han surgido diversas tecnologías para conseguirlo. XML, el Lenguaje de Marcado eXtensible, es una de esas tecnologías, que motivó principalmente la exposición del actual trabajo gracias a la divulgación generada con buenos comentarios, con un futuro prometedor y de gran alcance.

En la primera parte de este trabajo, se presentan los antecedentes de lo que hoy conocemos como Internet, su desarrollo y evolución, así como el surgimiento de servicios tales como Intranet y uno de los más importantes: la World Wide Web. Se muestra también, cómo es que los lenguajes de Marcado jugaron un papel muy importante dentro de la Web para poder intercambiar información de manera fácil y rápida, especialmente HTML, que actualmente es el lenguaje internacional para creación de documentos conocidos como "Páginas Web" dentro de la WWW e igualmente importante por ser su antecesor, el lenguaje SGML; explicando sus orígenes y crecimiento con el fin de comprender igualmente las complicaciones a las que han llegado dichos lenguajes y la evolución tomada de ambos en la descendencia del Lenguaje de Marcado eXtensible, XML.

Dentro del segundo capítulo se estudia a XML, su surgimiento como una tecnología alterna y a su vez complementaria a HTML, así como sucesora de SGML. Se analiza su forma de utilización, estructura y conceptos básicos para su comprensión. De igual forma, se menciona la relación que guarda con HTML.

Se expone la importancia que posee XML al ser un estándar por parte del Consorcio de la World Wide Web que permite elaborar etiquetas propias, con lo cual se pueden crear documentos *inteligentes* que ayudan a procesar toda una diversidad de información y simplificar las transacciones Business-to-Business (B2B) en la Web. Finalmente se mencionan algunas herramientas de apoyo, tecnologías derivadas y complementarias que tiene XML.

La aplicación final basada en XML se expone en el capítulo III, explicando cómo XML ofrece una solución para los requerimientos de la Síntesis Informativa en la Delegación Iztapalapa, al mantener una estructura tipo árbol y ser un lenguaje inteligible que funciona de forma adecuada para el dinamismo en la publicación de noticias.

Dentro de este último capítulo, se introduce primeramente al lector en el contexto sobre el cual trabaja la Delegación Iztapalapa para comprender la importancia de generar una Síntesis Informativa, la cual consiste en la recopilación de noticias de los principales periódicos dentro del Distrito Federal, así como cubrir su distribución al interior de la Delegación. Finalmente, en este apartado se describe el planteamiento tomado como solución para publicar y distribuir la Síntesis Informativa a través de la Intranet de la Delegación, el desarrollo de la aplicación generada y la programación llevada a cabo, explicando la utilización integral de la aplicación.

I.1. ANTECEDENTES

I.1.1. La red ARPA

Los orígenes de Internet se remontan a los años sesenta. Era la época de la guerra fría entre las dos superpotencias: Estados Unidos y la Unión Soviética. Nuevos impulsos en la informática de aquel tiempo eran realizados principalmente por iniciativas militares.

En el Departamento de Defensa de Estados Unidos se comenzó a reflexionar sobre la forma de cómo se podrían proteger mejor los importantes datos militares. Incluso en caso de un ataque nuclear, los datos no deberían ser destruidos. Como solución, entró en consideración solamente el proceso de datos electrónicos. Los mismos datos deberían establecerse en diferentes computadoras alejadas unas de otras. Todas las computadoras entrelazadas podrían enviarse en un lapso corto de tiempo, el estatus actual de los datos nuevos o modificados. Cada computadora debería tener la capacidad de comunicarse a través de varias formas con las demás computadoras. De tal manera se compondría una llamada "red" con la capacidad de trabajar también si una computadora individual o cierta línea llegara a ser destruida por un ataque del enemigo.

La Agencia de Proyectos de Investigación Avanzada (ARPA, Advanced Research Projects Agency), que forma parte del ejército de los Estados Unidos, realizó el proyecto planeado. En los primeros años la red fue por tanto llamada ARPANET (Red de la ARPA). A finales de 1969 los primeros cuatro ordenadores ya estaban conectados. Tres años más tarde se incrementó la cifra en 40 conectados.

I.1.2. Organizaciones científicas

Sin embargo el principio de las computadoras entrelazadas no era interesante solamente para propósitos militares. Se detectó rápidamente que también las instituciones académicas se beneficiarían de la ARPANET. Para los científicos era sin embargo mucho más importante el tener acceso a los datos de otras computadoras y no la sincronización de los mismos. Debido a la configuración abierta de la ARPANET, no había inconvenientes para el uso de la red de esa manera. A partir del comienzo de los años setenta, los científicos conectados podían llamar datos de otros ordenadores y poner a disposición los propios datos a otras instituciones.

Con la evolución de los ordenadores, en especial de la aparición de computadoras personales más accesibles, el número de computadoras entrelazadas se incrementó. Se trataba de tipos muy diversos de computadoras con sistemas operativos no compatibles y con diferentes accesos a la red. Ordenadores grandes de diversas marcas, ordenadores UNIX y un poco más tarde también computadoras personales se integraron a la red. Algunos tenían una línea arrendada, es decir, una conexión permanente; otras se conectaban con la ayuda de líneas telefónicas y módems.

Para traer los diferentes requisitos bajo un común denominador, nació la necesidad de desarrollar un nuevo protocolo para la red. El protocolo no debía depender de ningún sistema de computadoras, ni de la velocidad de conexión. Así fue como se originó el uso del protocolo TCP/IP. Después de la introducción de este protocolo, las transmisiones de datos tenían la característica de ser enviadas de forma uniforme y con un esquema estandarizado.

I.1.3. Extensión de la red

La conexión del mundo académico a la red requirió una separación entre la sección militar y la civil, puesto que los militares deseaban proteger sus propios intereses. Así fue como al principio de los años ochenta una nueva red de datos militares, la MILNET, se separó de la ARPANET y con ello la red fue cedida para las investigaciones científicas.

En la parte civil de la red el número de computadoras conectadas subió drásticamente en el curso de los años ochenta. Un papel muy importante jugó en ello la Fundación Nacional de Ciencias de los Estados Unidos (NSF, National Science Foundation). Esta organización creó un sistema de unión o entrelace que interconectó a los centros científicos más importantes del país. Las computadoras individuales de las universidades o pequeños centros de cómputo se podían conectar a un centro de sistemas para llegar a otras redes. Así se formó una red más extensa. Poco tiempo después se asentó el término "INTER-NET". La designación "ARPANET" fue abandonada a finales de los años ochenta.

El sistema de unión o entrelace por medio del cual las redes pequeñas se convirtieron en una red total, recibió la designación de "Backbone" (espinas dorsal de la red). Al mismo tiempo había en Europa progresos similares. En el viejo continente, sin embargo, se buscó el camino de los estándares ISO. Los europeos no querían saber nada del modelo TCP/IP, que no estaba nominado como estándar ISO. En 1986 fue creada la organización RARE (Réseaux Associés pour la Recherche Européenne), esta organización debería coordinar todas las iniciativas para el establecimiento de un sistema de red sobre toda Europa. La RARE fundó para ello un proyecto llamado Cooperación para Interconexión de Sistemas Abiertos de trabajo en Red en Europa (COSINE, Cooperation for an Open Systems Interconnection Networking in Europe).

El avance triunfante del protocolo TCP/IP no podía ser parado en adelante. Bajo la presión del éxito en los Estados Unidos finalmente surgió en Europa una red de datos con la capacidad de ser compatibles con múltiples protocolos, entre ellos el TCP/IP. A esta red se le designó el nombre de "EuropaNet". Diferentes redes nacionales, por ejemplo la Red de Investigación Alemana, fueron asociadas a ella.

Entretanto existe una línea transatlántica que une a Europa con los backbones en los Estados Unidos. La coordinación del tráfico de Internet en Europa es hecha por RIPE (Réseaux IP Européens). También en Europa existe desde 1992 un sistema de unión llamado Ebone (Europäischer Internet-Backbone). En otros continentes existen también progresos similares.

Lo que se entiende hoy en día como "Internet", no es una red homogénea, sino una red de muchas redes pequeñas, limitadas territorialmente o de redes limitadamente organizadas. Todas esas redes están unidas a los Backbones y de tal manera a toda la red. También los proveedores de Internet comerciales están conectados a Backbones.

1.2 ORIGEN DE LA WORLD WIDE WEB

1.2.1. Historia

A finales de los años ochenta en Ginebra, Suiza, Tim Berners-Lee (informático británico en la Organización Europea de Investigaciones Nucleares, "CERN"), inició con ayuda de algunos colegas una iniciativa para desarrollar un nuevo sistema de intercambio de información entre científicos por Internet.

En principio se pensó como un medio para la distribución de la información entre equipos de investigadores geográficamente dispersos; concretamente se dirigía a la comunidad de físicos vinculados al CERN. En su primera propuesta, Berners-Lee exponía las desventajas del uso de sistemas incompatibles e inconexos:

"En el CERN, una diversidad de datos está ya disponible: informes, datos experimentales, datos personales, listas de direcciones de correo electrónico, documentación informática, documentación experimental y muchos otros conjuntos de datos están girando continuamente en discos de ordenadores. Es sin embargo imposible, "saltar" de un conjunto a otro de una manera automática [...]. Usualmente, tendremos que utilizar un método de consulta distinto en un ordenador distinto con interfaces distintas. Una vez que se ha localizado la información, es difícil guardar sus conexiones o hacer una anotación privada que se pueda después encontrar rápidamente."

La conclusión era que "hay un enorme beneficio potencial en la integración de una variedad de sistemas, de un modo que permita a los usuarios seguir conexiones que apuntan de un elemento de información a otro".

Se pretendía pues, que los recursos disponibles en formato electrónico que residían en computadoras distintas conectadas a la red fuesen accesibles para cada investigador desde su terminal de un modo transparente y exento de dificultades, sin necesidad de aprender a utilizar distintos programas. Además, debería posibilitarse el salto entre elementos de información conexos. Los recursos existentes deberían integrarse en una red hipertextual distribuida, gestionada por ordenadores.

Se trataba de hacer visible entre otros tipos, documentos científicos en línea, en la cual debía jugar un papel importante el formato de textos y el ligamento de imágenes. Lo principal era la idea de hacer capaz el funcionamiento de hipertextos, de tal manera que los documentos pudieran contener enlaces a otros documentos, aún cuando estos se encontrasen en otras computadoras de la red. Pero más importante aún, era la obligada necesidad, de que el nuevo sistema de intercambio y representación de la información, debería manejarse como un estándar, donde no importara el tipo de computadora que generase o recibiera información.

En esa era de creación de aplicaciones informáticas, se pretendía hacer disponibles electrónicamente todas las ofertas de información imaginables (artículos, fotografías, textos, esquemas, etc.), para que se transmitieran por la red. Pero regularmente, se hizo necesario enlazar este tipo de objetos a textos aclaratorios o complementarios. Ninguna revista, por muy orientada a la imagen que esté, puede renunciar totalmente al texto. Esto se aplicaba también a las publicaciones multimedia y de hipertexto.

Aunado a lo anterior se tenía un problema más, el de publicar un cierto documento, en especial científico y saber que cualquier persona que lo revisara dentro de la red, lo recibiría en la forma básica de como fue editado. Para documentos científicos ha sido de gran importancia, manejar textos con distintos tipos de letra, tamaños, formatos, espacios, etc. De esta forma, se buscó además, la forma de conservar una estandarización hacia la publicación y recuperación de los documentos, para conservar la integridad durante su tratamiento.

Anders Berglund, uno de los primeros en adoptar un lenguaje especial de marcas, el "Lenguaje Generalizado y Estandarizado de Marcas" ¹ (SGML, Standard Generalized Markup Language), aconsejó a Tim Berners-Lee que utilizara una sintaxis de ese tipo de lenguaje, que parecía cumplir con los requisitos que demandaba el intercambio de información. El nacimiento del SGML se remonta al año 1969, cuando se formó en la empresa IBM el grupo de diseño dirigido por Charles Goldfarb, que tenía por objetivo el diseño de un sistema general que permitiera la interacción e intercambio de documentos. Fue así, hasta finales de los ochenta, cuando Tim Berners-Lee aplicó las normas del SGML a su proyecto de transmisión de datos por Internet.

A partir del tratamiento que hicieron con las normas de SGML, desarrollaron una versión de hipertexto basados en la idea del Lenguaje de Marcas, el cual, tenía una gran facilidad de uso y entendimiento para diseñar documentos tipo hipertexto y multimedia. A diferencia de SGML, este nuevo lenguaje era más sencillo, entendible y menos rígido, lo cual tuvo como consecuencia, que fuera rápidamente adoptado y aprendido para el fin perseguido por Berners-Lee. Finalmente fue desarrollado el "Lenguaje de Mercado de Hipertexto" ² (HTML, HyperText Markup Language), como solución para publicar las investigaciones de muy diversas fuentes y autores que se producían en el CERN.

Las dos columnas del proyecto tendrían que ser formadas por el nuevo formato de archivo HTML y por el nuevo protocolo para Internet HTTP (Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto), un protocolo ya estándar para la transmisión de datos de hipertexto a través de la red. Un nuevo software para los usuarios debería ser capaz de representar los archivos en línea que serían escritos con HTML. Se crearon entonces, los primeros visualizadores con la misión de manejar esos tipos de archivos.

Todo el proyecto de Tim Berners-Lee fue posteriormente bautizado como "World Wide Web" que significa Amplia Red Mundial (también se conoce como Web o por sus iniciales: WWW). Al mismo tiempo se comenzaron a configurar los llamados servidores para la red, que son las computadoras mayores que contienen la información y los recursos que los usuarios de la red necesitan. Los servidores deberían apoyar el nuevo protocolo HTTP.

¹ SGML se trata a fondo en el apartado "SGML Standard Generalized Markup Language" del capítulo I.

² HTML se trata a fondo en el apartado "HTML HyperText Markup Language" del capítulo I.

Los primeros programas para los usuarios fueron desarrollados. Los programadores de software fueron contagiados con la idea de la WWW y pronto se comenzaron a generar los primeros programas informáticos (conocidos como navegadores Web) que permitían recuperar el contenido de los archivos utilizando el protocolo HTTP.

Un joven llamado Marc Andreessen desarrolló el primer navegador para superficies gráficas, el navegador "Mosaic" y posteriormente nuevos programadores que trabajaron sobre Mosaic inventaron nuevas características que se implementaron en el navegador sin que estuvieran estandarizadas. Marc Andreessen por su parte abandonó el proyecto Mosaic y fundó una nueva compañía de software: "Netscape".

En la segunda mitad del año 1993, periódicos influyentes como el "New York Times" publicaron los primeros artículos sobre la fiebre en la comunidad de Internet. La cantidad de usuarios creció, así como la cantidad de gestores de servidores que instalaron el software HTTP gratuitamente convirtiéndose en servidores Web. Miles de personas con acceso a Internet comenzaron a escribir HTML para la WWW. Se comenzaron a escribir no sólo artículos científicos, sino también cuestiones personales o de interés común, entre otros muchos temas.

De esta manera surge un nuevo concepto: las "Páginas Web". Los usuarios que tenían el acceso a Internet, comenzaban a revisar los artículos que se publicaban en los servidores Web y que podían ser fácilmente visualizados gracias a las páginas Web escritas en lenguaje HTML. Este lenguaje jugó el papel más importante dentro de la WWW, debido a la simplicidad que posee para el desarrollo de páginas para publicaciones en Internet. Pronto se generaron diversos tipos de documentos con diferentes contenidos, permitiendo a la gente compartir información y trabajar en colaboración

La cantidad de soluciones de software y tecnologías en el campo de la WWW actualmente es tan extensa, que no puede ser totalmente abarcado. Entre más voluminoso sea el mercado de tecnologías, es mucho más importante que existan estándares reconocidos como el de HTML, pues al fin y al cabo, los datos son importantes y no el software. Muchos usuarios comienzan a entender esto y por ello la presión a las compañías de software ha aumentado, para que ellas se orienten a los estándares. Esto es válido para formatos de documentación como HTML, así como para soluciones de comercio electrónico.

La World Wide Web es el servicio más joven en Internet. La facilidad de la Web permite que aún usuarios con muy poca experiencia se puedan mover dentro de la gran oferta de información que se ofrece. Las páginas de la Web aparecen casi al instante en la pantalla de la computadora que las solicita. Una persona que solicita información en Internet, no necesita conocer los nombres de los archivos o comandos especiales. Si cuenta con la conexión para Internet, sólo necesita la ayuda del ratón de la computadora y de los enlaces resaltados en las páginas, para lograr ver la información.

El diseño de la World Wide Web sigue el modelo Cliente-Servidor: un paradigma de división del trabajo informático en el que las tareas se reparten entre un número de clientes que efectúan peticiones de servicios y un número de servidores que las atienden. En la Web, las estaciones de trabajo son clientes que demandan hipertextos (como las páginas HTML) a los servidores.

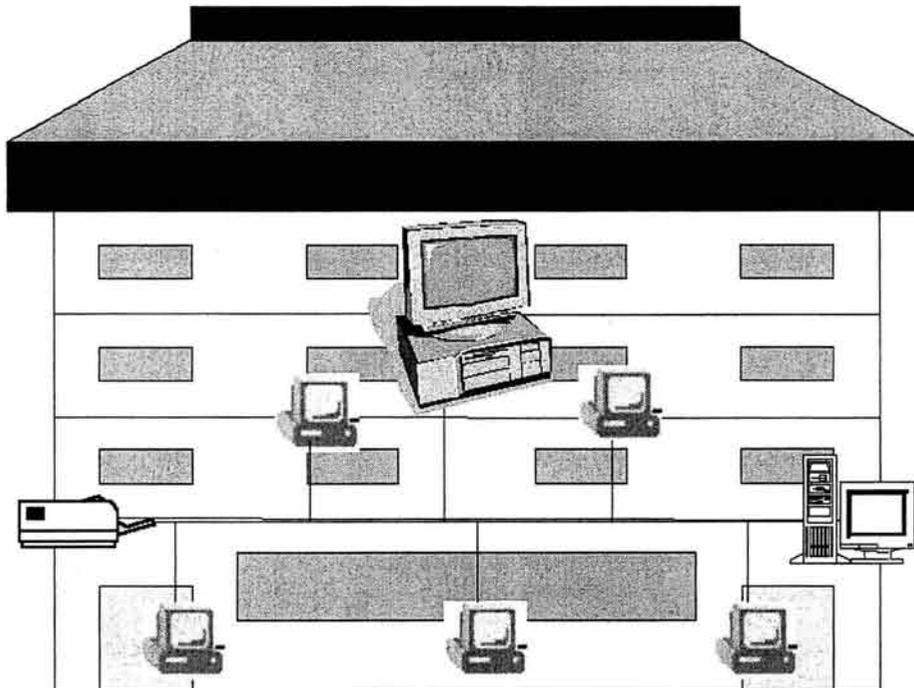


esquema 1.1 Modelo Cliente-Servidor.

I.2.2. Intranet y Extranet

En la actualidad el modelo de interconexión de computadoras no es de aplicación general, es decir, que una red de computadoras no necesariamente publica información para ser consultada en la WWW. Este tipo de redes puede publicar, acceder a la información y a las aplicaciones corporativas a través de navegadores Web convencionales, como si trabajara sobre Internet pero de forma privada, total o parcialmente aislada del mundo exterior.

Este tipo de red es conocido con el nombre de Intranet. Una Intranet puede considerarse como una red Internet local o interna para una organización. El acceso está limitado a los usuarios autorizados de la organización. La intranet puede estar limitada a una única ubicación o puede extenderse a más de una sede de una organización, es una herramienta de gestión que permite una potente difusión de información y mecanismos de colaboración entre el personal.

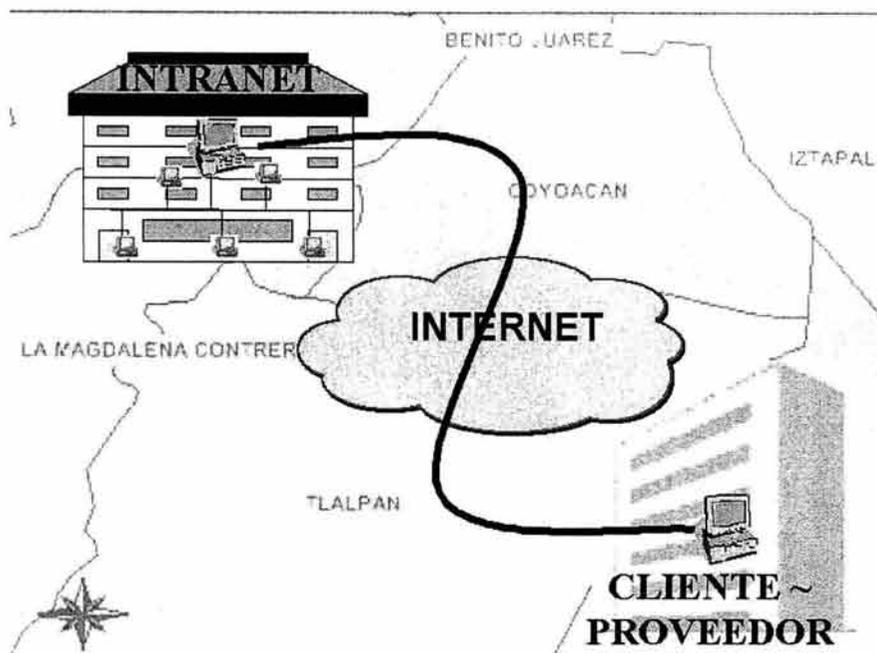


esquema 1.2. Modelo de Intranet

Se puede entonces definir una Intranet como la implantación o integración en una red local o corporativa de tecnologías avanzadas de publicación electrónica basadas en Web en combinación con servicios de mensajería, suministración de recursos, acceso remoto y toda una serie de facilidades cliente/servidor proporcionadas por la pila de protocolos TCP/IP. Su propósito fundamental es optimizar el flujo de información con el objeto de lograr una importante reducción de costos en el manejo de documentos y comunicación interna.

Una Extranet es una red de una organización, realizada empleando la tecnología Internet, diseñada para permitir el acceso de personas externas a la organización, tales como clientes o proveedores.

Los tres tipos de redes (Internet, Intranet y Extranet) están basados en el mismo modelo funcional de información: el intercambio de Páginas Web entre servidores y clientes mediante el uso de Navegadores Web.



esquema 1.3. Modelo de Extranet

El acceso a la información de una empresa desde una Extranet es mayor que el que puede tener cualquier usuario desde Internet, pero es menor que el concedido a los empleados que se conectan a la Intranet corporativa. Una organización puede mantener simultáneamente una Intranet, una Extranet y una presencia pública en Internet.

1.2.3. Características de la World Wide Web

Algunas de las principales características que definen al World Wide Web y que suponen ventajas considerables para su utilización y para el desarrollo de documentos son las siguientes:

- *Estructura hipertextual de la información.* La Web está especialmente diseñada para distribuir información con una estructura no lineal. La información se presenta dividida en pequeños nodos o núcleos e interconectada a través de enlaces o puntos de unión. Se trata de conexión entre bloques informativos formando una gran red de documentos que permiten al usuario "navegar" a través de ellos construyendo su propio itinerario de lectura. El "lector" avanza en la lectura de una forma intuitiva, asociando ideas con base en sus intereses, conocimientos y la disponibilidad de la información.
- *Carácter multimedia.* Los documentos Web permiten la presentación de información en diferentes códigos (texto, gráficos, audio, vídeo, entornos virtuales, etc.) o la combinación entre ellos de forma coordinada en un mismo documento. Al hablar de estructura hipertextual y carácter multimedia, sería más propio identificarlos como documentos hipermedia
- *Integración de las herramientas de comunicación de Internet.* Otra de las principales características de estos documentos, hace referencia a la posibilidad de activar diferentes herramientas de Internet como el correo electrónico, la conexión con una computadora específica remota (Telnet), buscadores de información, etc., lo cual supone la posibilidad de enviar un mensaje electrónico con el responsable de una página en cuestión sin necesidad de salir de la pantalla; la posibilidad de acceder al servidor donde se encuentra la información o transferir directamente archivos a la computadora del usuario.
- *Diferentes niveles de acceso a la información.* Los documentos Web son accesibles a través de Internet, en principio por cualquier persona que se encuentre conectada. Existen sin embargo materiales con acceso restringido, que requieren al usuario una suscripción o clave de acceso para visualizar la información. Por otra parte, también es posible consultar documentos sin salir a Internet, accediendo a archivos que físicamente residen en disco duro de la propia computadora o en otra conectada en red local.

- *Expansión de la información.* La Web es un espacio de información en constante movimiento, cada día aparecen páginas nuevas, se actualizan páginas existentes con nuevos datos o nuevos enlaces a otros documentos.
- *Compatibilidad entre plataformas.* Los documentos Web, utilizan lenguajes de marcado para su desarrollo. Lo cual hace posible la visualización de la información desde diferentes plataformas sin necesidad de convertir o duplicar los documentos.
- *Posibilidad de cierto control por parte del usuario sobre la presentación de la información.* El usuario puede modificar algunos elementos de presentación de la información, como la fuente de texto, color, tamaño, color del fondo, tamaño de la ventana o diferenciar los enlaces por donde el usuario ya ha pasado.

1.2.4. W3C

Bajo el contexto del desarrollo y crecimiento de la Web, en octubre de 1994, Tim Berners-Lee fundó el consorcio de la World Wide Web (W3C) en el Instituto de Tecnología de Massachusetts, laboratorio para la informática (MIT/LC) en colaboración con el CERN. El consorcio de la World Wide Web fue creado para conducir la World Wide Web a su capacidad máxima desarrollando los protocolos comunes que promueven su evolución y aseguran su interoperabilidad. El W3C tiene más de 500 organizaciones miembros alrededor del mundo y ha ganado el reconocimiento internacional por sus contribuciones al crecimiento de la Web.

Las metas a largo plazo de W3C para la Web son:

1. **Acceso Universal:** Para hacer la Web accesible y extenso a todos promoviendo las tecnologías que consideren las aplicaciones en cultura, la educación, la capacidad, recursos materiales y limitaciones físicas de los usuarios en todos los continentes;
2. **Web Semántico:** Para desarrollar un ambiente de software que permita que cada usuario haga el mejor uso de los recursos disponibles en la Web;
3. **Web de Confianza:** Para dirigir el desarrollo de la Web con la consideración cuidadosa para las ediciones legales, comerciales y sociales de los escritos planteados por cada tecnología.

El W3C concentra sus esfuerzos en tres tareas principales:

1. **Visión :** El W3C promueve y desarrolla su visión del futuro de la Web. Contribuciones mundiales de varios cientos de investigadores dedicados e ingenieros que trabajan para las organizaciones miembros del equipo de W3C y de la comunidad entera de la Web, permiten al W3C identificar los requisitos técnicos que deben ser satisfechos para que la Web sea un espacio verdaderamente universal de la información.
2. **Diseño :** El W3C diseña tecnologías de la Web para realizar esta Visión, considerando tecnologías existentes así como las del futuro.
3. **Estandarización :** El W3C contribuye a los esfuerzos de estandarización de tecnologías de la Web produciendo las especificaciones (llamadas también "recomendaciones") que describen los bloques de construcción de la Web, logrando que estas recomendaciones y otros informes técnicos estén disponibles libremente para todos.

I.3. LENGUAJES DE MARCADO PARA DOCUMENTOS.

I.3.1. Hipertexto

El hipertexto tiene ya una historia relativamente amplia, pues a pesar de que su nacimiento data apenas de los años sesenta, los cambios que se han experimentado en el campo de las tecnologías de la información han acelerado su desarrollo, de tal modo que actualmente hablar de hipertexto es casi sinónimo de hablar de documento electrónico. Los documentos electrónicos son cada vez más la mayoría de los documentos producidos en la actualidad y con bastante probabilidad lo seguirán siendo. Porque el hipertexto no es solamente una nueva tecnología de almacenamiento y recuperación de la información, es además, la cristalización de algunos sueños visionarios y de ciertos deseos que pretenden conjugar la amplitud y profundidad del conocimiento almacenado con la absoluta libertad de acceso a éste. El hipertexto no ha tenido un verdadero sentido mientras no ha existido un desarrollo tecnológico que lo haga posible; ha tomado sentido con la invención de la red Internet y de la World Wide Web. Es la Web la que ha actuado como amplificador de sus mejores y peores posibilidades.

El hipertexto es un tipo de texto electrónico, al mismo tiempo una tecnología informática y un modo de edición, que suponen una lectura no secuencial. Se puede entender como un conjunto de textos conectados entre sí mediante vínculos, que permite la elección del orden de lectura por parte del usuario. Se trata de un sistema cuya esencia es su capacidad asociativa entre piezas de información, por lo que algunos autores lo han llamado AIMS: Associative Information Management Systems (Sistema de Administración de la Información Asociada).

Quizás las dos tipologías más aceptadas de estos sistemas, las debidas a Jeff Conklin y a T.J. Byers³ puedan acotar de algún modo el término. Jeff Conklin enumera cuatro tipos de sistemas hipertexto:

- *Los macroliterarios*, sistemas que soportan grandes conjuntos documentales en línea.
- *Herramientas para la exploración de problemas*, que gestionan ideas sin estructurar sobre un problema determinado.
- *Sistemas de consulta o browsing systems*, similares a los macroliterarios pero a pequeña escala, enfocados a la enseñanza, el trabajo de referencia y la información pública. No permiten interacción del usuario.
- *Tecnología General Hipertexto*, sistemas generales aplicables de modo muy amplio.

La siguiente clasificación, la de Byers, es más genérica y distingue solamente:

- *Sistemas hipertexto estáticos*, que sólo permiten la búsqueda a través de enlaces ya creados.
- *Sistemas hipertexto dinámicos*, que permiten la participación del usuario mediante notas, nuevos vínculos, añadido o supresión de datos, etc.

El hipertexto consta de dos elementos básicos, el nodo o pieza de información y el enlace, que relaciona unos nodos con otros. Estos elementos constitutivos tan simples dan lugar a desarrollos muy complejos, de gran riqueza y heterogeneidad, que se corresponden exactamente con el modo en que el cerebro humano construye sus representaciones de la realidad y permiten una recuperación rica y libre mediante la llamada navegación, o travesía entre nodos relacionados mediante enlaces.

Bush, director del Departamento de Desarrollo e Investigación Científica de los Estados Unidos, consideraba que el crecimiento exponencial de la información científica hacía imposible que los especialistas se mantuvieran informados de los desarrollos de una determinada disciplina. Básicamente pensaba que el avanzar de subclase en subclase para acceder a una información especializada haciendo y deshaciendo más de una vez las rutas de acceso a cualquier información y siguiendo unas reglas rígidas, no se adapta a la forma en la que pensamos y por lo tanto hace más lenta y difícil la recuperación de información.

³ Citadas por Pedro Razquín Zazpe, Del hipertexto al multimedia interactivo. Evolución situación actual y perspectivas de los sistemas y aplicaciones multimedia,. *Revista General de Información y documentación*. Madrid, Universidad Complutense. 1997, vol. 7 n. 1

Para evitarlo ideó un sistema teórico llamado MEMEX (MEMory Extended System), que funcionara de forma similar a como lo hace la mente humana, no en forma secuencial, sino con continuos saltos o asociaciones entre unidades arbitrarias. Lo imaginó como una combinación de dos tecnologías entonces emergentes, la microfotografía y la lógica electrónica. Su característica básica era su capacidad para unir pares de elementos y podría usarse para hojear y hacer anotaciones en un gran sistema de gráficos y textos en línea, además de una gran biblioteca de documentos. A pesar de que Bush teorizara sobre la tecnología necesaria para construir este sistema, nunca se convirtió en una máquina real.

Douglas Engelbart, durante el desarrollo del proyecto Augmented promovido por el Stanford Research Institute (Instituto de Investigación de Standford), en el año 1962, planteó la tesis de desarrollar un conjunto de herramientas orientadas a aumentar la capacidad y productividad humana. Estos planteamientos llevaron al diseño de técnicas e instrumentos que hoy parecen comunes, pero que en su día supusieron la creación de nuevos entornos en la informática, como el "ratón" o "mouse" de la computadora, la distribución de la pantalla en distintas ventanas, sistemas de ayuda integrados o correo electrónico.

Pero quien de verdad acuñó el término hipertexto fue Theodor Nelson en 1965, dentro del proyecto Xanadu, que pretendía almacenar y enlazar toda la literatura mundial, accediendo a ella desde cualquier terminal. Los usuarios podrían integrar también cualquier pieza informativa en sus propios documentos, que a la vez pasarían a formar parte de la red universal. Señala Nelson que sobre la base del documento se pueden establecer relaciones con cualquier otro documento y que nuevos vínculos permiten añadir constantemente nuevas rutas de acceso a materiales ya existentes. Pero aunque los fundamentos teóricos ya están puestos y aunque ya existe el término, no hay aún un verdadero sistema de hipertexto.

En los primeros tiempos, los sistemas hipertextos que se diseñaron fueron en realidad software de generación de hipertextos, más tarde con el desarrollo de las tecnologías como bases de datos, dispositivos de almacenamiento óptico y por último redes e Internet, se fue ampliando su alcance hasta adquirir su actual omnipresencia.

I.3.2. Marcado de formato

El sistema de procesamiento de texto es una disciplina de la ciencia informática dedicada a la creación de sistemas de computadora capaces de automatizar partes de la creación de un documento y proceso de edición. El software de procesamiento de texto incluye desde sencillos procesadores de texto, hasta avanzadas bases de datos de noticias, sistemas de representación de documentos por hipertexto (como Páginas Web), entre otras herramientas de edición.

La primera ola en el procesamiento de texto automatizado llegó con la tipografía informática. Después de teclear el documento, el autor describe cómo desea que esté formateado. La computadora lo imprime con el texto descrito y el formato seleccionado.

Se denomina *reproducción* al formato de archivo que contiene la combinación de datos reales del documento y la descripción del formato deseado. Algunas famosas notaciones de reproducción son Troff, el Formato de texto enriquecido (RTF, Rich Text Format) y LaTeX.

El sistema convierte la *reproducción* en algo perceptible para el ser humano, la *presentación*. Históricamente, el medio de representación ha sido el papel impreso, ahora también lo es una pantalla electrónica.

Los sistemas de tipografía han acelerado el proceso de edición de documentos y evolucionado hasta lo que se conoce actualmente por edición de sobremesa. Los programas mas recientes, como Word y Adobe Pagemaker siguen trabajando con reproducciones, si bien proporcionan a los autores una mejor interfaz para manejarlos. La interfaz de usuario para la reproducción está diseñada para tener el aspecto de una *presentación* (el producto acabado impreso en papel).

Esta clase de tipografía recibe el nombre de WYSIWYG, por sus siglas en ingles, What You See Is What You Get ("lo que ves es lo que obtienes"). Puesto que la *reproducción* se limita a describir una *presentación*, tiene sentido la interfaz de usuario que refleje al producto acabado.

La clase de notación de tipografía que precedió al aún en uso WYSIWYG, recibe el nombre de *marcado de formato* en la cual, se marca el texto con instrucciones llamadas etiquetas o códigos, dependiendo del lenguaje de marcado en que se haga la notación. LaTeX, es un ejemplo de este tipo de lenguaje, donde marcas hechas con simbología especial y palabras clave, describen el formato del texto que limita las marcas. Como el marcado sólo utiliza caracteres propios de los teclados convencionales, se puede utilizar con editores de texto en vez de procesadores de texto especiales (como Word, que aparecieron posteriormente).

El procedimiento de marcado generalizado es válido si el objetivo es introducir documentos en la computadora, describir una reproducción e imprimirlos. Pero hacia finales de los años sesenta, los usuarios comenzaron a exigir más de los documentos. En particular, IBM hizo ver que muchos de sus sistemas eran incapaces de comunicarse entre sí porque cada uno utilizaba un lenguaje de control distinto. No podían leer los archivos de los demás sistemas, algo así como si no se pudiera leer un archivo del procesador de texto de WordPerfect en el procesador de Word.

El problema es que cada aplicación utiliza sus propias marcas para describir las entidades de los documentos. Las "marcas" son los códigos que indican al programa cómo debe tratar a su contenido, de esa forma, si se desea que un texto aparezca en cursiva, cada aplicación introduce al principio y al final del texto correspondiente una marca que le permita mostrarlo en pantalla e imprimirlo adecuadamente. Lo mismo ocurre con las tablas, los márgenes, las imágenes, los tipos de letra, los enlaces, etc.

Los usuarios del editor de texto WordPerfect conocen perfectamente este sistema de marcas, ya que las aprovechan desde las primeras versiones del programa para controlar a la perfección sus documentos. Word también permite tratar algunas de las marcas que utiliza, aunque muchas menos de las que en realidad utiliza internamente. No obstante, las marcas que utiliza cada sistema propietario son intratables directamente desde el código interno del archivo que describe el documento.

I.3.3. Marcado Generalizado

IBM solicitó a Charles Goldfarb, junto con otros dos investigadores, tratar de solucionar el problema de un marcado generalizado. El equipo identificó tres aspectos importantes. En primer lugar, que era necesario que los programas soportaran una representación genérica de documento. Es decir, las herramientas no pueden trabajar en común si no hablan en el mismo idioma. Tiene sentido que en el contexto del procesamiento de texto el lenguaje común sea una especie de lenguaje de marcado, porque el marcado se entiende bien y es compatible con los editores de texto y los sistemas operativos.

En segundo lugar, los tres comprendieron que el formato común debía ser *específico*. El equipo podía haber inventado un lenguaje simple, quizás similar a la representación de un procesador de texto estándar, pero esa representación no hubiera permitido cierto nivel de complejidad en el procesamiento.

La tercera conclusión del equipo de IBM fue que si los sistemas de computación debían trabajar de forma fiable con los documentos, éstos debían seguir algunas normas. Por ejemplo, en una carta no puede insertarse una posdata al principio del documento, ni una dirección al final.

Conocer las marcas que utiliza cada programa de tratamiento de documentos hace posible diseñar filtros que permitan pasar la información de unos formatos de marcas a otros sin perder el diseño. La forma que IBM creó para solventar este problema se basaba en tratar marcas accesibles desde documentos universales TXT, es decir, en código ASCII, haciendo así posible su tratamiento desde cualquier sistema.

En los años sesenta, IBM intentó resolver sus problemas asociados al tratamiento de documentos por diferentes plataformas a través de lo que denominó GML (Generalized Markup Language, lenguaje de marcas generalizado).

Más tarde, GML cayó en manos de ISO, que lo convirtió en un estándar oficial en los ochentas (ISO 8879), denominándose SGML (Standard Generalized Markup Language, lenguaje de marcas generalizado estándar). Esta norma de carácter general se aplica desde entonces para diseñar lenguajes de marcas específicos, cuyos ejemplos más conocidos son el HTML y el RTF.

Los lenguajes de marcas no son equivalentes a los lenguajes de programación, aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que si se ajustan a SGML se pueden controlar desde cualquier editor ASCII. Las marcas más utilizadas suelen describirse por textos descriptivos encerrados entre signos de "menor que" (<) y "mayor que" (>), siendo lo más usual que existan una marca de principio y otra de final. Existe la posibilidad de aprovechar la etiqueta para incluir atributos internos, ahorrando así otras etiquetas.

Se puede decir que existen tres usos básicos de los lenguajes de marcas: los que sirven principalmente para describir su contenido, los que sirven más que nada para definir su formato y los que realizan las dos funciones indistintamente. Las aplicaciones de bases de datos son buenas referencias del primer sistema, los programas de tratamiento de textos son ejemplos típicos del segundo tipo, y con ciertas limitaciones, el HTML es la muestra más conocida del tercer modelo.

I.4. SGML - Standard Generalized Markup Language

I.4.1. Historia

El nacimiento del SGML se remonta al año 1969. Fue entonces cuando se formó en la empresa IBM el grupo de diseño dirigido por Charles Goldfarb, que tenía por objetivo el diseño de un sistema general que permitiera la intercambiabilidad de documentos. En colaboración con Edward Mosher y Raymond Lorie, Goldfarb diseñó el lenguaje GML. Aunque originalmente esta abreviatura se formó a partir de las iniciales de sus creadores, ya desde entonces se asoció a Generalized Markup Language. El GML gozó muy pronto de una gran aceptación y dentro de la empresa IBM, a partir de entonces, empezaron a crearse en la práctica todos los documentos basándose en este sistema.

En el año 1978 el Instituto Nacional Norteamericano de Estándares (ANSI, American National Standards Institute), formó un grupo de trabajo para la especificación de lenguajes de computación en relación con el procesado de textos, con el que también colaboraría Charles Goldfarb. Una vez concluido este diseño, surgió definitivamente el estándar SGML (Standard Generalized Markup Language, Lenguaje de Marcas Generalizado Estándar).

El SGML es un estándar internacional para la descripción de la codificación electrónica de texto creado por el Dr. Goldfarb a partir de los trabajos ya mencionados y adoptado como norma ISO8879 en octubre de 1986. Con más exactitud SGML es un metalenguaje, un modo de describir formalmente un lenguaje, en este caso un lenguaje de codificación.

I.4.2. Características

Tres características de SGML lo distinguen frente a otros lenguajes de marcación:

1. *El uso de marcación descriptiva más que procesal.* Un sistema descriptivo de marcas usa códigos de marca que simplemente proporcionan nombres para categorizar partes de un documento: por ejemplo, "<parra>" indica que lo que sigue es un párrafo. Por el contrario, un sistema de marcación procesal define los procedimientos que deben realizarse en los puntos particulares de un documento: por ejemplo, llamar al procedimiento parra con parámetros para ejecutar alguna acción. En SGML las instrucciones para procesar un documento con algún propósito especial se distinguen cuidadosamente de las marcas descriptivas que aparecen dentro del documento.
2. *Su concepto de "tipo de documento".* En SGML los documentos se consideran como portadores de tipo. El tipo de documento se define formalmente según sus partes constitutivas y su estructura. Si los documentos son tipos conocidos, un programa –analizador SGML– puede usarse para procesar un documento que va a parecerse a un tipo particular, revisando que todos los elementos requeridos para ese tipo de documento estén realmente presentes y correctamente ordenados.

3. *Su independencia respecto de cualquier sistema de representación.* Un objetivo básico del diseño de SGML, fue asegurar que los documentos codificados según sus indicaciones pudieran ser transportados de una marca de hardware/software a otra sin pérdida de información.

Por lenguaje de marcación entendemos una serie de convenciones de marcas usadas para la codificación de textos. Este lenguaje debe especificar cuáles son las marcas permitidas, cuáles las necesarias, cómo una marca se distingue del texto y cuál es su significado. SGML permite hacer las tres primeras tareas, las normas de la Text Encoding Initiative ("TEI", un proyecto internacional que pretende crear un estándar de codificación que facilite el intercambio de texto en formato electrónico), permiten realizar la última.

Un texto, un documento para SGML no es una secuencia no diferenciada de datos. Según los propósitos, éste se puede dividir en unidades de diferentes tipos o tamaños. Un texto en prosa puede dividirse en secciones, capítulos, párrafos y oraciones. Un texto en verso puede dividirse en cantos, estrofas y versos. Una vez impresas, las secuencias de prosa y verso pueden dividirse en volúmenes, colecciones y páginas. Dichas unidades son usadas frecuentemente para expresar localizaciones específicas o puntos de referencia dentro del texto ("la octava oración del tercer párrafo del capítulo tres", por ejemplo), pero también pueden usarse para subdividir un texto en fragmentos de significado con propósitos analíticos (como saber si la longitud de oración de la segunda sección, es diferente de la longitud de oración de la sección que le continúa).

Además, en un texto en prosa se pueden considerar los pasajes en estilo directo o indirecto, los pasajes que emplean diferentes registros estilísticos (narrativo, polémico, comentario,...), los pasajes de narrador diferente, etc., como unidades de distintos tipos.

Estas estructuras textuales se superponen unas a otras de manera compleja e impredecible. El lector necesita estar prevenido tanto acerca de la organización física del texto o libro como de la estructura lógica del trabajo que contiene.

1.4.3. Definición de Tipos de documento

SGML clasifica todos los documentos en tipos de documento. La definición de estos tipos de documento especifica cuáles son las unidades estructurales o de otro tipo que podemos encontrar en todos los documentos que pertenecen a ese tipo.

Podemos explicar a la Definición de Tipo de Documento (DTD, Data Type Definition) como una especificación formal de las estructuras de un documento SGML.

SGML se basa en el concepto de documento, que es compuesto de una serie *de entidades*. Cada entidad puede contener unos o más elementos *lógicos*. Cada uno de estos elementos puede tener ciertas *cualidades* (características) que describan la manera en la cual debe ser procesada. El SGML proporciona una manera de describir las relaciones entre estas entidades, elementos y cualidades e indica a la computadora cómo puede reconocer las piezas de un documento.

El término técnico usado en la estandarización SGML para la unidad textual, visto como componente estructural es el *elemento*. Esta unidad textual o elemento se marca con un membrete de apertura en su comienzo y un membrete de cierre en su final. Este membrete describe la característica distintiva del elemento. A los creadores de series de mimbretes conformadas según SGML corresponde seleccionar nombres inteligibles para los elementos que se desean identificar y documentar su correcto uso en la marcación de textos. Estos mimbretes o mejor llamados "marcas", son distinguidos del texto porque se encuentran entre los signos de "menor que" (<) y "mayor que" (>), así por ejemplo, si se encuentra el texto <marca> dentro del documento, el procesador reconoce que la cadena <marca>, debe ser tratada como una marca y no como alguna parte del texto.

De la necesidad de dar a los elementos nombres indicativos de función surge el término técnico de identificador Genérico (GI, Generic Identifier). El GI identifica por tanto, el tipo de elemento ("párrafo", "lista", "poema",...). Además, pueden existir otras caracterizaciones de estos elementos llamadas atributos que los definen más específicamente.

Un atributo se usa entonces para describir información, que en algún sentido describe ocurrencias de un elemento específico, pero que en sí misma no es considerada como un elemento.

La declaración de cada elemento en la DTD, debe especificar dos parámetros de declaración de elementos: el "identificador Genérico" (GI) y el "modelo de contenido" de lo que manejará.

El parámetro de modelo define qué subelementos o cadenas de caracteres pueden aparecer en el contenido.

Por ejemplo, la declaración para un libro de texto puede ser la siguiente:

```
<!ELEMENT libro_de_texto (encabezado, cuerpo, final)>
```

Un modelo de contenido es un tipo de grupo que está constituido por una serie de elementos relacionados llamado ejemplares "tokens". Un grupo debe aparecer entre paréntesis que son los delimitadores "apertura de grupo" y "cierre de grupo". Estos paréntesis son necesarios incluso cuando el modelo de grupo incluya un solo "token".

Los ejemplares en un modelo de grupo son también identificadores genéricos (Gis). Como modelo de contenido se pueden definir también palabras especiales. Existen varias palabras de este tipo, una de ellas es #PCDATA que es la que aparece con mayor frecuencia. Es una abreviatura de "datos de carácter analizados" y significa que el elemento que está siendo definido puede contener cualquier dato de carácter válido.

Una característica específica de #PCDATA es que permite que en su interior haya elementos que puedan ser procesados que no necesariamente tienen que aparecer declarados en la DTD en la que aparecen.

Un ejemplo de uso de #PCDATA lo constituye la definición del elemento *forma* en la DTD lingüística:

```
<! ELEMENT forma
      (#PCDATA
       | (forma. nombre, objeto*)
       | objeto+)
```

En este caso *forma* presenta una estructura de #PCDATA que contiene en su interior elementos que deberán ser procesados como tales.

Es posible también introducir caracteres directamente sin que sean procesados como elementos SGML, para dar cuenta de ésto se utiliza la palabra especial #CDATA ("character data").

Un ejemplo de uso de #CDATA podría ser la siguiente definición del elemento "fórmula"

```
<! ELEMENT fórmula CDATA>
```

Una variante de #CDATA es #RCDATA ("replaceable character data") que es exactamente igual que la anterior con la única diferencia de que se permite la introducción de texto que contenga referencia a entidades o referencia a caracteres.

El uso de palabras especiales o palabras clave incluye también la posibilidad de definir modelos vacíos. Elementos para los cuales el usuario nunca especifique el contenido.

Indicadores de ocurrencia y Conectores de grupo

Los indicadores de ocurrencia indican cuántas veces puede aparecer cada GI's en el texto. Existen tres indicadores de ocurrencia en la sintaxis de SGML, convencionalmente representados por el signo más (+), el signo de interrogación (?) y el asterisco o estrella (*).

El signo más significa que puede haber más de una ocurrencia del elemento en cuestión; el signo de interrogación significa que puede haber como mucho una y posiblemente ninguna ocurrencia; el asterisco significa que el elemento en cuestión puede o bien estar ausente o aparecer una o más veces.

Los conectores de grupo indican el orden de los GI's. Hay tres conectores de grupo representados convencionalmente mediante la coma (,), la barra vertical (|) y el ampersand (&). La coma significa que los componentes en conexión deben aparecer en el orden especificado por el modelo de contenido. El ampersand indica que los componentes especificados deben aparecer pero pueden hacerlo en cualquier orden, la barra vertical indica que sólo uno de los elementos en conexión puede aparecer.

Los modelos de contenido pueden modificarse mediante la inclusión de una lista de excepciones:

```
<! -- ELEMENTS CONTENIDO (EXCEPCIONES)? ->
```

Existen dos tipos de excepciones en SGML:

1. Inclusiones. Se trata de incluir elementos que pueden aparecer en cualquier punto de la DTD que se está definiendo (por ejemplo en el elemento al que se adjunta o en cualquier otro incluido en él). Las inclusiones se declaran entre paréntesis precedido de un signo más (+). Por ejemplo, en un poema se puede permitir la introducción de anotaciones y variantes ("nota | variante") de lecturas en cualquier parte del texto. Para ello es necesario declarar los siguientes dos elementos:

```
<! ELEMENT (nota|variante) --- (#PCDATA)>
```

que se une al elemento poema en la forma de una lista de inclusión:

```
<! ELEMENT POEMA - o (TÍTULO?, (ESTROFA + | PAREADO+ | VERSO+) ) + (NOTA | VARIANTE)>
```

Así agregadas, notas y variantes pueden aparecer en cualquier punto del contenido del elemento poema incluso en aquellos (por ejemplo, TÍTULO) para las que se definiera un modelo de contenido #PCDATA.

2. Exclusiones. Se trata de identificar elementos que no pueden usarse mientras el elemento que está siendo definido permanezca abierto. Las exclusiones se declaran entre paréntesis precedido de un signo menos (-).

Si en la definición del elemento poema se quisiera evitar que notas o variantes parecieran dentro de títulos, se podría agregar una excepción de exclusión a la declaración de TÍTULO dada:

```
<! ELEMENT TÍTULO -o (#PCDATA) -(NOTA|VARIANTE)>
```

Entre las GIs y la descripción del modelo de contenido se puede tener lo que se llama la minimización de reglas aplicadas al elemento en cuestión. Estas reglas determinan si los mimbretes de comienzo y final deben estar o no presentes en cada ocurrencia del elemento, toman la forma de un par de caracteres separados por un espacio en blanco, el primero de ellos está relacionado con el membrete inicial; el segundo con el final. En cualquier caso, el guión o la letra "o" (para "opcional") han de ser dadas; el guión indica que el membrete debe estar presente y la letra "o" que puede omitirse.

El elemento estrofa o el elemento verso en un poema, por ejemplo, puede no tener elemento final pero sí el comienzo, o de una nueva estrofa o de un nuevo verso, lo que significa el final del anterior.

Como se mencionó anteriormente, un membrete incluye normalmente un identificador genérico (GI) y que puede incluir también atributos que lo caracterizan.

Un atributo es una caracterización de un elemento que se expresa como un par atributo-valor y se añade al membrete inicial del elemento al que se refiere. Los atributos se pueden usar, entre otras cosas, para:

- identificar el estatus de un documento o texto. Por ejemplo: `<memo status=procesado>`
- controlar cómo será el formato de un texto. Por ejemplo: `<linea tipo=centrado>`

- definir el tamaño o fuente de datos almacenados externamente que serán añadidos al documento. Por ejemplo: `<dibujo distanciax="120mm" distanciy "90mm">`

En una especificación de atributo hay dos partes: "nombre del atributo" y "valor del atributo" unidos por un "indicador de valor". Los valores del atributo pueden ser valores de atributo literales. Una literal es una cadena de caracteres reconocida como unidad por el sistema desde el momento en que se incluye entre delimitadores literales. En SGML existen dos tipos de delimitadores literales, el primero se representa entre comillas (") y el segundo mediante apóstrofo (').

Al igual que los elementos, en SGML los atributos se declaran en la DTD, usando una sintaxis bastante similar. Así como se especifican sus nombres y el elemento al que se vincula, es posible especificar (dentro de ciertos límites) el tipo de valor aceptable para un atributo y su valor omitido.

La declaración comienza con el símbolo `<!ESPLIST`, que introduce una especificación de la lista de valores de un atributo. Su primera parte especifica el elemento (o elementos) a los que caracterizan los atributos. Si varios elementos comparten los mismos atributos, todos ellos podrían definirse en una sola declaración. Después del nombre, hay una serie de enumeraciones, una para cada atributo que ha sido declarado, divididas a su vez en tres partes. Esas listas incluyen el nombre del atributo, el tipo de valor que puede tomar y el valor omitido, respectivamente.

```
<!ESPLIST POEMA
id ID #IMPLIED
status (anteproyecto|revisado|publicado) anteproyecto>
```

La segunda parte de una especificación de atributos puede tomar una entre dos formas, ambas ejemplificadas.

En el primer caso se usa una de las palabras clave para declarar el tipo de valor que un atributo puede tomar. En el ejemplo de arriba, la palabra clave ID indica que el atributo ID se usará para proporcionar un valor único identificable a cada instancia del poema:

Entre otras, podemos tener las siguientes palabras clave:

CDATA.	El valor del atributo puede contener cualquier dato de carácter válido, incluyendo membretes.
IDREF.	El valor del atributo debe contener un indicador de algún otro elemento.
NMTOKEN.	El valor del atributo es una muestra de nombre, lo cual significa (sin mayor precisión) una palabra única.
NUMBER.	El valor del atributo está compuesto sólo de números.

En el segundo caso, el correspondiente al atributo status, se ha proporcionado una lista de valores posibles para el atributo status. Esto significa que un analizador puede comprobar que ningún POEMA se define si el atributo status no tiene un "anteproyecto", "revisado" o "publicado" como uno de sus valores.

La última pieza de cada información en cada definición de atributo especifica el modo en que un analizador deberá interpretar la ausencia del atributo en cuestión. Esto puede hacerse proporcionando una de las palabras clave de la lista que viene a continuación:

# REQUIRED	Debe especificarse un valor.
# IMPLIED	No es necesario proporcionar un valor (como en el caso de ID anterior).
# CURRENT	Si el valor no se proporciona en la ocurrencia del elemento, deberá usarse el último valor especificado.

O como en el caso de status proporcionando un valor que será luego tomado como valor por defecto en cualquier otra aparición del elemento (en éste caso, poema).

I.4.4. Unidades de contenido: las entidades

Los aspectos de SGML presentados hasta aquí están relacionados con la codificación de las partes o elementos estructurales de un documento. SGML también proporciona un método simple y flexible para codificar, nombrar libremente y de un modo transportable partes del contenido de un documento. ISO8879 define entidad como "un conjunto de caracteres que pueden ser referidos como una unidad".

Un documento SGML distingue dos tipos de entidades:

Entidades generales.

Se usan para referirse a un texto previamente definido, para marcar instrucciones o llamadas a procedimientos en el interior de un documento, o ambas. Entre las entidades que se refieren a una cadena de caracteres o a la totalidad de un archivo podemos incluir una entidad como la siguiente:

```
<! ENTITY SGML "Standard Generalized Markup Language">
```

Define una entidad cuyo nombre es SGML y cuyo valor es la cadena "Standard Generalized Markup Language".

Un ejemplo de entidad que llama a una instrucción es la siguiente:

```
<! ENTITY página PI "nueva página; 3 espacios">
```

En donde PI indica que la entidad debe ser interpretada como una llamada a una instrucción que en este caso posiblemente querría decir "página nueva, introdúzcanse tres espacios".

Entidades de parámetro.

Las entidades de parámetro conforman un grupo especial de entidades que aparecen únicamente en la declaración SGML. La mayoría de las veces se refieren a elementos SGML que se usarán en esa misma declaración.

Declaración de entidades

La declaración de entidades generales consta del nombre de la entidad seguida de una cadena de caracteres entrecomillados que conforman su contenido.

La declaración de entidades de parámetro se distingue de las anteriores por la presencia de un signo de porcentaje (%) que precede al nombre de la entidad.

Una vez que la entidad (general o parametrizable) ha sido declarada, debe mencionarse en cualquier parte del documento. Esta mención se hace refiriendo su nombre precedido por el carácter ampersand (&) seguido de punto y coma. Este último puede omitirse siempre y cuando a la referencia de una entidad le siga un espacio o algún otro tipo de signo que pueda interpretarse unívocamente como un final.

Recapitulando todo lo dicho hasta ahora, una Definición de Documento Tipo (DTD) debe contener:

- Una declaración de elementos que especifique qué identificadores genéricos (GI) pueden aparecer en cada elemento y en qué orden.
- Una declaración de la lista de atributos que especifique qué atributos pueden aparecer para cada elemento, sus valores posibles y sus valores por defecto.
- Una declaración de entidades que defina las entidades que pueden referirse en el documento.

1.5. HTML - HyperText Markup Language

1.5.1. Definición

HTML (HyperText Markup Language, Lenguaje de Marcado de HiperTexto), se trata de un lenguaje que es definido con ayuda de SGML. Por tanto es un llamado lenguaje de marcación (Markup Language). Tiene la tarea de describir las partes lógicas de un documento. Como lenguaje de marcación, HTML contiene comandos y etiquetas ya definidas para marcar elementos típicos de un documento, por ejemplo encabezados, párrafos, listas, tablas o referencias de imágenes.

Las reglas detalladas para el HTML (los nombres de tags/elements y cómo pueden ser utilizadas) se definen usando a otro lenguaje, el SGML, el cual tiene un cierto grado de dificultad, además de que fue diseñado para colecciones masivas de documentos, tales como manuales de mantenimiento para los submarinos nucleares, etc.

El esquema de descripción de HTML se basa en un orden jerárquico. HTML describe documentos. Dichos documentos tienen cualidades globales como por ejemplo un título o color de fondo. El contenido propio se compone de elementos, por ejemplo un encabezado de primer orden. Algunos de esos elementos poseen subelementos. Un párrafo contiene por ejemplo una o varias palabras escritas en negrita, una lista numerada se compone de los puntos de la lista y una tabla contiene celdas.

La mayoría de estos elementos tienen un campo de extensión definido y fijo. Un encabezado se extiende desde el primer hasta el último símbolo. Una lista numerada se extiende desde el primer punto hasta el último punto de la lista. Una tabla se extiende desde la primera hasta la última celda. Bajo este esquema, se marcan el principio y el final de los elementos. Los navegadores que pueden visualizar archivos HTML, interpretan la combinación de comandos y representan su contenido en la pantalla.

1.5.2. Cualidades

Una de las cualidades más importantes de HTML es la posibilidad de definir vínculos o enlaces ("Hyperlinks" o simplemente "links"). Los enlaces pueden conducir a otros documentos del mismo proyecto, pero también a otras direcciones en la World Wide Web y aún a direcciones de Internet que no forman parte de la WWW.

Mediante esa simple cualidad principal HTML abre totalmente nuevos mundos. El movimiento entre computadoras remotas es reducido a unos vínculos en los navegadores modernos. En un proyecto propio se tiene la posibilidad de poner enlaces entre contenidos propios y ajenos. Este es el principio principal de la World Wide Web y por tal razón le debe su nombre a esa idea. En la era comercial de Internet los enlaces se han convertido en un objeto de comercialización.

HTML es un llamado formato de texto plano. Los archivos HTML pueden ser creados con cualquier editor de texto que pueda grabar archivos de textos planos. No existe por tanto un programa especial para crear estos archivos. Aunque actualmente existen algunos programas muy poderosos especializados para programarlo, eso no cambia nada en las principales cualidades: HTML no está ligado a ningún programa de software especial.

Los comandos de textos planos de HTML han sido diseñados para máquinas y personas. Alguien que no tenga problemas o dificultad sobre trabajar comandos en pantalla, encontrará en HTML un idioma de comandos muy simple. El idioma es inglés, sin embargo por la cantidad determinada de comandos, es fácil aún para personas sin conocimientos de inglés, adentrarse en HTML.

HTML se puede generar muy fácilmente con ayuda de algún programa. Programas de CGI⁴ por ejemplo, usan esa posibilidad. Los resultados que aparecen en la pantalla cuando se solicita alguna información en un buscador, son enviados generalmente por un programa. Eso es posible ya que HTML está en formato de texto plano.

⁴ CGI se explica en el apartado "HTML su uso con otras aplicaciones" del capítulo I.

La facilidad de uso y la particularidad de que no era propiedad de nadie, hizo a HTML el sistema idóneo para compartir información en Internet. La expansión de Internet le ha dado una posición de privilegio y ha hecho que la idea inicial se modifique considerablemente.

HTML es un lenguaje de marcación diseñado para la creación de Páginas Web. Pero los archivos HTML funcionan no tan sólo en la Web. No es ningún problema abrir un archivo HTML local en el disco duro de una computadora. Por esta razón los archivos HTML pueden ser ideales para la documentación interna o local, para distribución de CD-ROM, etc.

Con HTML y los lenguajes suplementarios como CSS⁵ y JavaScript⁶ (que también funcionan localmente), es posible realizar proyectos de muy buena calidad que no estén disponibles en la red. No importa si se desea escribir un diario para el siguiente milenio o si se desea escribir la ayuda en línea de un programa de software, HTML es el formato de archivo más distribuido del mundo. Todos los archivos HTML pueden ser visualizados con cualquier computadora en donde haya un navegador instalado.

I.5.3. Versiones de HTML

No es digno de mencionar la primera versión de HTML. La especificación de HTML 1.0. no se encuentra ni siquiera en las páginas del consorcio W3C que es el responsable de su estandarización, además esa versión no está disponible en el mercado.

HTML 1.0 contenía comandos para elementos estándares como encabezados, párrafos, referencias de imágenes y por supuesto para enlaces.

Quizás se hubieran demorado 10 años hasta la siguiente versión. Sin embargo mientras HTML 1.0 era la versión oficial, comenzó el éxito de la WWW y de los navegadores Web como Mosaic y Netscape, este último en su versión 1.1 dominaba tablas en HTML aún cuando el estándar 2.0 no había sido publicado y aún la versión 3.0 de HTML estaba lejos de declarar las tablas como estándar oficial.

HTML 2.0 se convirtió en el estándar oficial en noviembre de 1995. Esa especificación se puede encontrar aún en las páginas del Consorcio W3.

La versión 2.0 de HTML es el común denominador más pequeño. Un navegador que no interprete los comandos de HTML 2.0 no se puede llamar navegador para páginas Web. Sin embargo HTML 2.0 contiene algunos comandos que hasta el momento no son interpretados correctamente por los navegadores modernos, por ejemplo la etiqueta de enlace.

HTML 2.0 fue tomado con una gran decepción, por el hecho de que Netscape había logrado mucho más en el desarrollo de su navegador: HTML 2.0 era el estándar en el momento cuando el "Navigator" de Netscape en su abrumadora versión 2.0 apareció en el mercado. En esta versión el navegador de Netscape ya interpretaba Frames (técnica de varias ventanas) y JavaScripts que estaban lejos de ser estándar oficial de HTML.

La mayoría de los creadores de páginas Web se adentraron en las nuevas posibilidades que ofrecía Netscape y no podían entender las restricciones de HTML 2.0.

Después de largas discusiones HTML 3.2 se convirtió en el lenguaje oficial el 14 de enero de 1997. En el número de la versión se puede ver que esta versión tuvo muchos problemas para ser tomada como estándar. El Consorcio W3 desarrollaba al principio la versión 3.0, sin importarle el éxito del navegador de Netscape. HTML 3.0 contenía algunas proposiciones muy interesantes que hasta el momento son ignoradas por los navegadores modernos (por ejemplo: comandos para representar fórmulas matemáticas).

⁵ CSS se explica en el apartado "HTML su uso con otras aplicaciones" del capítulo I y se trata a fondo dentro del capítulo II en el apartado "Herramientas de apoyo" como un subconjunto de DSSSL

⁶ Javascript se explica en el apartado "HTML su uso con otras aplicaciones" del capítulo I.

Sin embargo las proposiciones eran contrarias a la nueva realidad en la WWW. En aquellos tiempos el Consorcio W3, que pertenece al mundo de las ciencias, comenzó a colaborar con productores de software comercial. HTML 3.2 es el resultado de una total revisión de las proposiciones de HTML 3.0, versión que nunca se convirtió en versión oficial. Las tablas fueron finalmente reconocidas en forma oficial, además de diversos comandos para la marcación física de texto. Los productores de navegadores, entre ellos ya Microsoft, se quedaron esperando la toma de los "frames" como estándar, pero fue en vano.

HTML 4.0 fue aprobado el 18 de febrero de 1998 oficialmente como lenguaje oficial. Con HTML 4.0 el Consorcio W3 demostró que está lista para cooperar con los productores de navegadores comerciales y que de esta manera ve su papel como gremio cooperativo de estandarización en el mercado multimillonario del WWW. Para los empedernidos científicos, este no fue un paso fácil, pero sí uno muy importante. Pues sólo un Consorcio W3 que sea respetado por los productores de software tiene una existencia segura. Muchas personas reconocen hoy en día el significado que tiene este gremio independiente después del intento de algunos productores de desarrollar e implantar su propio pseudo-estándar.

HTML 4.0 reconoce el uso de frames y el ligamento de hojas de estilo (CSS, Cascade Style Sheets) para documentos HTML. De esta manera casi todos los suplementos de HTML han sido reconocidos como estándares, o por lo menos están previstos a ser estándares de HTML.

La especificación XHTML 1.0 (recomendación del 26 de enero del 2000 del W3C) es una reformulación de HTML como aplicación XML. Por el momento se definirá al Lenguaje de Marcado eXtensible (XML, eXtensible Markup Language)⁷, como un lenguaje de marcado que surge de las especificaciones de SGML diseñado para ser ampliable, es decir, que se puedan crear etiquetas propias o elementos nuevos y principalmente, que sea conforme a las reglas de buena formación y validez, lo que significa que todos los elementos requeridos estén presentes y correctamente ordenados, de la misma forma como lo define SGML.

La situación actual, en lo que respecta al HTML, no coincide con la idea original. Por este motivo, el W3C ha sido el responsable de tomar la decisión de reformular el HTML 4.0 para adaptarse a XML (solución muy fácil), en vez de crear un nuevo HTML que volviese al redil de SGML (solución mucho más difícil de imponer) o aconsejar que se utilice el ya existente SGML (realmente mucho más complejo y difícil de utilizar).

XHTML exactamente es la reformulación de las tres definiciones de tipo de documento HTML 4.0 como aplicaciones XML⁸. Su finalidad es ser usado como lenguaje de contenidos que es a su vez conforme a XML. Como tales son fácilmente visualizados, editados y validados con herramientas XML estándar. Los documentos XHTML pueden escribirse para que funcionen igual o mejor que lo hacían antes, tanto en los documentos conformes a HTML 4.0 como en los nuevos conformes a XHTML 1.0. Los documentos XHTML pueden usar aplicaciones (scripts y applets) que se basen ya sea en el Modelo del Objeto de Documento (DOM)⁹ de HTML o XML.

Se puede decir sin lugar a dudas que el XHTML está perfectamente interrelacionado con XML y HTML, cogiendo lo mejor de cada uno, o sea, las conocidas etiquetas del HTML y la estricta normativa de XML. Matemáticamente, se podría decir que: XML + HTML = XHTML (expresión poco técnica, pero clara).

Las razones esgrimidas por el W3C para aconsejar el uso del XHTML sobre HTML son dos, principalmente:

- XHTML, ya que es una aplicación XML, ha sido diseñado para ser ampliable (de ahí el añadido de la palabra *Extensible*). Esto significa que se pueden añadir nuevas etiquetas o elementos sin alterar la Definición de Tipo de Documento (DTD) en la que está basado el análisis del documento.
- XHTML ha sido diseñado pensando en la portabilidad. Aunque hoy en día la unión de la potencia de las computadoras y de los navegadores es suficientemente para asumir las posibles diferencias y pequeños errores del código HTML, se espera que para los próximos años se produzca un aumento considerable de los aparatos que sean capaces de tratar información en código HTML, no disponiendo estos de dicha potencia. Televisores, teléfonos móviles, computadoras de bolsillo, calculadoras, hornos, tostadoras,

⁷ XML es el tema principal presentado ampliamente en el capítulo II,

⁸ La especificación oficial de XHTML se encuentra en <http://www.w3c.org/TR/2002/REC-xhtml1-20000126>

⁹ DOM se explica en el apartado "Tecnologías derivadas y complementarias a XML" del capítulo II.

etc., soportarán código HTML, siempre que esté realmente unificado y se ajuste a normas estrictas para no dar problemas que exijan soluciones complejas.

A medida que la familia XHTML evolucione, los documentos conformes a XHTML 1.0 estarán más preparados para interactuar dentro de y entre los distintos entornos XHTML. Los desarrolladores descubren constantemente nuevas formas de expresar sus ideas usando nuevas etiquetas. En XML es relativamente fácil añadir nuevos elementos así como atributos adicionales a dichos elementos a los ya existentes. La familia XHTML está concebida para acomodar estas extensiones a través de módulos XHTML y técnicas para desarrollar nuevos módulos conformes a XHTML.

Constantemente se desarrollan nuevas formas de acceder a Internet. Muy probablemente las peticiones de documentos que se visualicen en Internet se realizarán desde esas plataformas alternativas. La familia XHTML está concebida teniendo en mente la interoperabilidad con sistemas generales. A través de un nuevo mecanismo de especificación de documentos.

Por tanto, el XHTML es el siguiente paso en la evolución de Internet. Al migrar en este momento hacia XHTML, los desarrolladores de contenidos Web entran también en el mundo de XML con todos los beneficios que se esperan de él, a la vez que se asegura la compatibilidad con sistemas pasados y futuros.

1.6. LIMITACIONES DE HTML

El hipertexto, después de haber sido considerado durante muchos años como una remota posibilidad, se ha convertido en una realidad omnipresente gracias al World Wide Web. Pero esta presencia constante constituye un grave riesgo de pérdida de sus mejores cualidades. Si el hipertexto solo sirve para crear conjuntos de información miscelánea, no elaborada y superficial, o mejor dicho para repetir el desorden sin añadir nada nuevo, no merece la pena el tiempo transcurrido hasta que fuera esto posible.

El hipertexto es una herramienta documental que debe conceder valor añadido a la información, relacionando las fuentes primarias con índices o vocabularios. Existen infinidad de tareas pendientes y son muchas las posibilidades para los profesionales. Un campo ilimitado es usar un índice especializado en una determinada materia que remita a documentos completos.

HTML, tal como fue concebido, era un lenguaje para el intercambio de documentos científicos y técnicos adaptado para ser usado por no especialistas en el tratamiento de documentos. HTML resolvió el problema de la complejidad del SGML sirviéndose de un reducido conjunto de etiquetas estructurales y semánticas apropiadas para la realización de documentos relativamente simples. Además de simplificar la estructura de documentos, HTML soportaba el hipertexto. Las posibilidades de usar elementos multimedia fueron añadidas con posterioridad.

En un corto período de tiempo, HTML se hizo muy popular y rápidamente superó los propósitos para los que había sido creado. Desde los albores del HTML, ha habido una constante invención de nuevos elementos para ser usados dentro de HTML (como estándar) y para adaptar HTML a mercados verticales, altamente especializados. Esta cantidad de nuevos elementos ha llevado a problemas de compatibilidad de los documentos en las distintas plataformas.

Dada la creciente heterogeneidad de programas y plataformas, está claro que la idoneidad del HTML 4.0 "clásico" para ser usado en dichas plataformas es más que limitado.

Tras algunos años en los que el interés por este tema parecía haber disminuido, en los últimos años del siglo XXI han aparecido algunas nuevas contribuciones, que tienden sobre todo a poner de manifiesto las grandes posibilidades y las nuevas vías de investigación. Las nuevas aportaciones se refieren sobre todo al hipertexto en la Web y por tanto resaltan la necesidad de un buen diseño formal y estructural. Esto requiere el desarrollo de las llamadas herramientas de autor, que faciliten la generación de hipertextos coherentes y también desarrollen modelos formales de especificaciones de diseño de documentos.

En principio, la intención de HTML era que las etiquetas fueran capaces de marcar la información de acuerdo con su significado, sin importar cómo se mostraban en la pantalla. Lo importante era el contenido y

no la forma, o sea, que era un lenguaje de marcas orientado a describir los contenidos. En otras palabras: el título del documento, los títulos de los apartados, el autor del documento, los textos resaltados, el comienzo de un párrafo y otras descripciones, eran marcados por las etiquetas TITLE, HEAD, ADDRESS, STRONG, P, etc., dejando a cada navegador (Browser o Visualizador) la tarea de formatear el documento según su criterio.

Esto producía presentaciones diferentes, pero permitía controlar fácilmente su contenido. Si una persona o un motor de búsqueda quería conocer el título del documento, el autor de la página o las cabeceras de los capítulos, siempre buscaba en el código las etiquetas TITLE, ADDRESS o HEAD. Además, si a alguien no le gustaba la idea de dejar a cada aplicación la decisión de cómo mostrar el contenido de las etiquetas, siempre le quedaba la posibilidad de controlar el formato del documento con descripciones particulares, como es el caso de las hojas de estilo en cascada (CSS)¹⁰.

Por diversos motivos, los creadores de los navegadores fueron añadiendo más etiquetas HTML dirigidas a controlar la presentación y los usuarios las utilizaron para que sus documentos estuviesen perfectamente formateados, sin permitir diferencias importantes entre visualizadores distintos, por lo que HTML pasó a ser un lenguaje de marcas más dirigido al control de la presentación.

Ahora es más difícil que se localice al autor o las cabeceras de los capítulos de un documento, pues la mayoría de los textos se describen sin las etiquetas que corresponden a dichos elementos, sobre todo si se utilizan los editores WYSIWYG (como Microsoft FrontPage, Netscape Composer) que proliferan por doquier.

Si a esto se le agrega que para facilitar la vida a los usuarios, los analizadores sintácticos de las marcas que incluyen los navegadores permitieron saltarse algunas normas sin que el propio usuario lo notase (por ejemplo, permiten trabajar solo con la etiqueta de párrafo <P>, cuando lo correcto es que se necesite las etiquetas de principio y de final: <P> y </P>, dando como resultado que HTML ya no es un lenguaje que sigue las normas estrictas de SGML.

La mayoría de las páginas en la Web no cumplen con las especificaciones de HTML del W3C's. En algunos casos esto se debe a que sus autores desearon aprovecharse de extensiones propias; pero en muchos otros casos, simplemente por descuido. Las páginas se habrían podido fácilmente hacer con más conformidad sin ningún problema.

La razón fundamental para cumplir con los estándares del HTML es para asegurarse de que se utiliza solamente elementos y estructuras que sean probables de ser entendidos por la amplia gama de usuarios. Es como el inglés, si se habla un Inglés Británico en Estados Unidos, con palabras propias del estilo británico, muchas frases del habla inglesa Americana no tienen el mismo sentido que frases del inglés Británico. De forma semejante, si se utilizan dialectos propios del HTML, algunos usuarios no entenderán todo lo que se quiso decir. Hay sin embargo, un estándar para el HTML que los fabricantes de navegadores o browsers dicen utilizar y la validación precisa las divergencias de los documentos HTML. Al Adherirse a los estándares se maximiza la accesibilidad del trabajo a los diversos usuarios.

Otro aspecto que HTML deja atrás, es que sus documentos son documentos estables o inmóviles, por ejemplo, un documento HTML puede contener información de datos que necesiten ser actualizados o sustituidos día a día o con cierta frecuencia, por lo tanto un primer documento creado con cierta información, no será útil en un periodo de tiempo posterior porque la información que contenía ya no es actual. Aunque esto pasa a segundo término si la estructura del documento cambia con una nueva estructura y la mayoría del contenido también. Pero cuando la estructura es básicamente la misma o solo se desean modificar ciertos elementos, puede volverse una tarea difícil si el documento es extenso.

Por ejemplo, si alguna compañía desea publicar su catálogo de productos en la Web, creará un documento HTML que muestre la información deseada, como precio, modelo, tamaños, etc. De antemano, se sabe que cierta información no es estática, como el precio. Suponiendo que de un día para otro los costos dentro de su mercado se elevan y por tanto los precios también, tendrá entonces que modificar los precios que fueron publicados en su documento HTML. O quizás desea agregar imágenes de sus productos dentro del catálogo. Si su lista de productos es de gran tamaño entonces deberá hacer una búsqueda exhaustiva para

¹⁰ CSS se trata en el subcapítulo posterior. CSS se trata también en el Capítulo II: XML y su interacción con otras herramientas-

modificar todos los precios de su catálogo o agregar las imágenes. Lo cual le tomará cierto tiempo en el mejor de los casos, pero lo más importante es que este proceso se realice de forma manual y se corra el riesgo de cometer errores.

Aunque por lo general, la mayor parte de las empresas que empiezan en Internet no tienen problemas con la extracción de la información cambiante, porque es muy probable que creen sus sitios Web con herramientas modernas orientadas a Internet; por otra parte, las compañías tradicionales enfrentan retos importantes al tratar de manejar sus sitios con sistemas heredados (antiguos).

El consorcio de la World Wide Web convocó en 1999 a un taller sobre el futuro del HTML. Como la organización es quien ha mantenido y ha fomentado el estándar del HTML desde que el IETF lanzó la especificación del HTML 1.0, el W3C tiene un interés sobre HTML, tal y como lo tienen muchos de los miembros del W3C.

El propósito de este taller se planteó para que el W3C escuchara a sus miembros y se determinara qué acciones apoyan en lo mejor posible a las necesidades de sus miembros.

El HTML tiene ciertamente su pasado e importancia actual. Pero la aceptación entusiástica de XML y el hecho de que el consorcio W3C disolvió a su grupo de trabajo de HTML después de publicar la recomendación del HTML 4.0, pudo haber dejado el estado de HTML como solo cuestión de formato de datos de salida y significativo.

Dos participantes del taller que representan a ISO (International Standards Organization, Organización de Estándares Internacional) tomaron la referencia de como HTML ha constituido un estándar con una influencia tal que ISO ha dado a HTML su tratamiento de estándar.

La ISO ha estandarizado a HTML en la convicción de que el HTML persistirá por lo menos 25 años. Dada esta larga situación en la visión de ISO, el futuro de HTML tiene por lo menos un voto substancial de confianza. Por otra parte, haciendo HTML un estándar, ISO espera que el W3C siga siendo responsable del HTML. Pero la ISO es una organización conservadora que registra estándares existentes; se retira de la tarea de conducir la innovación.

Si sólo se desea exhibir texto, no hay nada malo con HTML, pero para el proceso automatizado de la Web (documentos enriquecidos de una manera que permite a programas de computadora hacer algo con ellos) HTML no es suficiente.

1.7. HTML su uso con otras aplicaciones

1.7.1. CGI (Common Gateway Interface)

El Lenguaje de Marcado de Hipertexto, HTML, ha recorrido mucho camino desde que fue creado, proporcionando a los desarrolladores Web muchas más opciones y posibilidades gráficas. Sin embargo, el HTML por sí mismo no proporciona casi ninguna interactividad aparte de la posibilidad de "saltar" de un documento a otro.

De cualquier manera, siempre existen métodos para añadir algo de interactividad adicional a las páginas, permitiendo que sus lectores puedan realizar ciertas cosas directamente desde su página Web. Por ejemplo, puede ofrecer la posibilidad de se busque cierta información contenida en una base de datos, que ofrezca comentarios sobre lo que se ha escrito o presentado, o que se seleccionen varios objetos dentro de un formulario y se envíen. Hasta ahora, este tipo de interactividad se conseguía gracias a algo llamado Common Gateway Interface (Interfaz Común de Entrada), conocido generalmente como CGI.

De la manera más sencilla posible, un programa gateway es un programa que se ejecuta en un servidor Web cuando se introducen unos datos desde un navegador. El programa es generalmente un enlace entre el servidor y algún otro programa que se esté ejecutando en el servidor, como una base de datos. La mayoría del trabajo se realiza en el servidor, sin que ocurra a la vista del usuario. Lo único que se ve es el resultado.

El CGI es una posibilidad de poner a la disposición programas en Internet que pueden ser llamados con ayuda de páginas HTML y que pueden generar código HTML y enviarlos a un navegador.

Cuando se busca en la WWW un dato de bancos, se compran productos, firma un libro de visitas o mira un contador con la cantidad de accesos a la página Web, hay que tener en cuenta que por lo general siempre existe un CGI o una interfaz como CGI detrás de todo esto.

CGI - son programas que se encuentran en un servidor en Internet y que al ser ejecutados procesan determinados datos. El procesamiento de datos ocurre en el servidor. Programas de CGI pueden almacenar en el servidor, por ejemplo, cuantas páginas han sido visitadas, o que ha escrito un visitante en un libro de visitas. Un programa CGI puede también por ejemplo generar código HTML llamando y leyendo datos ya almacenados. Esas páginas generadas "dinámicamente" son enviadas al navegador del usuario y pueden ser representadas en la pantalla.

La llamada interfaz o interfaces CGI debe ser apoyada por el software del servidor Web. A vista de una persona que alquile un espacio en un servidor Web, la interfaz CGI se encuentra a su disposición en forma de un determinado directorio. La mayoría de las veces este directorio tiene el nombre de CGI-BIN. En este directorio se pueden almacenar programas que toman funciones de CGI.

No existe ninguna prescripción de en que lenguaje deben ser programadas las aplicaciones CGI. Para que el programa pueda ser ejecutado en el servidor, debe ser compilado por el sistema operativo del servidor, o debe existir un interprete que pueda ejecutar el programa. Si por ejemplo el entorno es Windows-NT, entonces los CGI Scripts pueden ser archivos ejecutables EXE. Hoy en día la mayoría de los programas CGI son escritos en el lenguaje de Shell de Unix o en Perl.

La grandeza del CGI reside en que permite un tipo de comunicación bidireccional a través del uso de HTML.

1.7.2. CSS Style-Sheets

HTML es un lenguaje "lógico". Con HTML se define la estructura de páginas Web o sea los elementos, enlaces, referencias de imágenes, multimedia etc. Sin embargo, en un principio HTML no fue diseñado para determinar exactamente el aspecto de un elemento. De esta manera es posible definir un encabezado y determinar que el encabezamiento sea de primer nivel. Pero, dentro HTML era posible determinar el tamaño o con que tipo de letra debe ser visualizado el encabezado. Ese trabajo lo hace el navegador. Para ello utiliza una combinación de ajustes básicos que el usuario puede escoger, y la representación fija y programada de los elementos de HTML.

En los primeros tiempos de éxito de HTML, el lenguaje fue "abusado" para toda clase de formatos físicos. De pronto apareció un comando HTML para la representación de texto brillante y después otro comando con el que se podía determinar el tamaño de las letras en 7 niveles relativos. Sin embargo todas estas implementaciones en HTML son inconsecuentes.

En este punto entran en juego las hojas de estilo (CSS). Las hojas de estilo son un lenguaje suplementario que fue especialmente desarrollado para HTML. Ellas permiten dar formato a algunos elementos de HTML. Con las hojas de estilo es posible por ejemplo, determinar que todos los encabezados del primer nivel tengan 24 puntos de tamaño, sean de color rojo, que estén en fuente Helvética u otros formatos deseados. Eso es válido para cualquier clase de texto, por ejemplo que un párrafo determinado tenga un tamaño de 3 centímetros y con un color de fondo amarillo. Además las hojas de estilo permiten la posición exacta de elementos en la pantalla y también para otros medios como impresos o medios visuales. El lenguaje CSS es como HTML un lenguaje oficial. El Consorcio W3 se encarga del cuidado y desarrollo de CSS.

1.7.3. JavaScript

Entre otras cosas es posible definir formularios en HTML. Tales formularios pueden contener campos de entrada, listas de selección, botones etc. El usuario puede llenar un formulario y enviarlo por la Web. Sin embargo HTML no le permite al proveedor verificar los datos después de que el usuario haya llenado el formulario y antes de que éste envíe los datos.

Otro ejemplo: aunque con HTML es posible ligar un archivo VRML, una vez que el archivo esté en la Web, no se puede saber si el usuario posee un navegador que pueda representar VRML. En este caso sería práctico hacer depender el comando de ligamento del archivo de si el navegador del usuario puede representar VRML o no.

Netscape inventó un lenguaje de programación llamado JavaScript para esta y otras cosas útiles, pero también para cosas menos útiles. Los comandos de JavaScript se pueden anotar directamente en los archivos HTML, así como también pueden estar fuera de ellos.

JavaScript juega un papel cada vez más importante en páginas Web modernas. Nuevos desarrollos en JavaScript son el resultado de lo que ahora mismo se conoce como HTML dinámico. JavaScript permite el acceso posterior a todos los elementos de un archivo HTML durante la representación. De tal manera es posible crear nuevos efectos especiales que no pueden ser hechos con HTML normal. Un texto puede por ejemplo ser reemplazado por otro al ser tocado con el ratón, elementos pueden también desaparecer de la pantalla después que el usuario haya dado un determinado paso.

1.7.4. Shockwave

Una de las últimas adiciones es un producto llamado Shockwave, de Macromedia, Inc., creado por los desarrolladores de Director, un programa muy usado de creación de animaciones multimedia. Shockwave es lo que se conoce como "plug-in" o módulo de extensión, un programa diseñado específicamente para ser usado en otro programa ya existente. Shockwave es un plug-in diseñado para ser usado con cualquier navegador que lo soporte. Su objetivo es reproducir las animaciones Director que se encuentran en una página Web, en tiempo real. De esta manera, no será necesario que tenga que esperar a que toda la animación sea recibida y más tarde ejecutarla con un programa especial. Un usuario que posea este plug-in recibe la animación al mismo tiempo que la página Web, y su computadora la reproduce localmente. El recibir las animaciones permite que sean reproducidas mucho mejor que si hubiera que reproducirlas en un servidor y posteriormente transmitir las por Internet.

La recepción de una animación, aún siendo pequeña, puede llevar bastante tiempo, lo que provoca que la página Web en la que se encuentra se cargue muy lentamente. Macromedia ha desarrollado un nuevo método de compresión, llamado Afterburner, que reduce una animación (o película) Director a un tamaño tan pequeño que puede ser descomprimida y reproducida más rápidamente. Cualquiera que tenga un navegador con soporte para Shockwave puede ir a cualquier página que lo use y ver inmediatamente una animación. Esta nueva tecnología está todavía en fase de pruebas.

A pesar del hecho de que Shockwave es un evento fundamental para el World Wide Web, el avance más importante en interactividad y presentación de contenidos multimedia es un lenguaje de programación llamado Java.

1.7.5. Java

Java es un lenguaje de programación desarrollado por Sun Microsystems y que es independiente de los sistemas operativos, además de tener una proyección para el uso en la World Wide Web. El lenguaje se parece en su estructura y sintaxis a C/C++.

Java es apropiado para realizar animaciones, simulaciones, aplicaciones en vivo, así como aplicaciones interactivas como guías turísticas, juegos o aplicaciones de calculación. Todas esas aplicaciones pueden ser visualizadas en la ventana de un navegador que apoye Java.

Java se distingue mediante las siguientes cualidades:

- *Programación orientada a objetos.* Java es un lenguaje orientado para la programación de objetos y que dispone de una gran cantidad de objetos simples y complejos para el programador.

- *Administración de memoria segura.* Desde el punto de vista de programación bajo Java la administración de memoria de los trabajos no es tan crítica como en C. No existen campos de memoria de tamaños indeterminados dirigibles con indicadores, sino vectores de memoria con una longitud fija definida. La poca administración de memoria es muy importante, ya que los programas escritos en Java tienen que ser portados fácilmente a los diferentes entornos de CPU.
- *Los archivos de programas son independientes de la plataforma.* Los programas Java son compilados a código objeto como cualquier programa, pero son ligados a un determinado procesador o entorno de sistema operativo. Por tal razón los programas Java funcionan en cualquiera plataforma, siempre que esté instalado un interprete que soporte el código objeto Java (en los navegadores es la llamada "consola Java").

Los programas Java que están destinados para el uso en Internet se llaman Applets y pueden ser referenciados en archivos HTML. La interacción entre el usuario y el programa aparece en la ventana del navegador.

En el caso de un Java-Applet que es referenciado en un archivo HTML, el código del programa es cargado en la memoria principal de la computadora que hace la demanda y es ejecutado por su interprete Java.

1.7.6. Active X

ActiveX es una tecnología introducida por Microsoft para ejecutar códigos de programas en páginas Web y la pretensión de ser una alternativa o competencia para Java. ActiveX es una definición general para diferentes componentes de software. Todos esos componentes se basan en el llamado Component Object Model (COM). ActiveX no es sin embargo un estándar en Internet, sino el intento de aprovechar algunas cualidades del sistema operativo MS-Windows para páginas Web.

Con ayuda de ActiveX es por ejemplo posible leer las entradas de un formulario HTML directamente en una tabla Excel o banco de datos Access. Esto es posible con la ayuda de la interfaz OLE de Windows. El proceso puede hacerse al contrario, es decir, de una tabla Excel a un formulario HTML. Microsoft trata de buscar soluciones que puedan ser transmitidas a otros sistemas operativos como Macintosh y Unix, pero hasta el momento sin éxito.

Los componentes de ActiveX llamados ActiveX-Control, son programas o módulos de programas que se pueden ligar en archivos HTML, semejante a los Java-Applets. El código del programa es ejecutable en la memoria principal de la computadora del usuario. Normalmente es instalado en la computadora del usuario un archivo layout de ActiveX (en el directorio Windows). Con ActiveX es posible realizar animaciones, juegos etc. en páginas Web.

No existe ninguna disposición sobre en que lenguaje de programación el código ActiveX debe ser escrito. El compilador debe apoyar el Component Object Model. Muy a menudo el lenguaje de programación propio de Microsoft, VBScript, es usado para la programación.

ActiveX sólo puede ser directamente ejecutado por el navegador MS Internet Explorer. Para Netscape existe un ActiveX-Plug-in para su uso.

Actualmente esta en discusión el concepto de seguridad de ActiveX. Si un usuario permite que un ActiveX-Control sea cargado en su computadora, entonces ese programa puede ejecutarse sin restricción alguna. No existe ninguna limitante para los comandos de ActiveX, sino tan sólo una "barrera de confianza" (a conocer la pregunta del navegador si el usuario está de acuerdo con la ejecución del programa en su computadora). Desde el punto de vista de programación se puede realizar muchas cosas más que con Java-Applets, sin embargo desde el punto de vista del usuario los módulos de ActiveX son mucho más inseguros que los Java-Applets.

1.7.7. VRML

VRML (Virtual Reality Modeling Language, Lenguaje de Modelado de Realidad Virtual). El lenguaje se entiende como complemento para HTML con el propósito de hacer una experiencia tridimensional en la Web. VRML es como HTML un lenguaje estandarizado para la Web. La idea para este lenguaje surgió en la primera Conferencia de la World Wide Web en 1994 que tuvo lugar en Ginebra Suiza. Primero se había pensado integrar la descripción de los objetos virtuales 3D en HTML, sin embargo muy pronto estuvo claro que para ello eran necesarios otros esquemas de descripción. De esta manera surgió un lenguaje independiente de HTML que fue llamado VRML. VRML no describe referencias de imágenes y textos a diferencia de HTML, sino la estructura vectorial de objetos en 3D y sus dependencias.

VRML existe actualmente en la especificación 2.0. Esta versión permite crear objetos en 3D que pueden contener enlaces a otros objetos de la misma clase. De esta forma se puede realizar la navegación. Desgraciadamente la versión 2.0 no es totalmente compatible con la versión anterior, la versión 1.0, lo que hace que el software para la visualización de archivos VRML tenga problemas en ocasiones.

Para la visualización de documentos VRML existen navegadores especiales o Add-Ons para los navegadores Web existentes. En la presentación de un documento VRML no se navega haciendo clics en textos o imágenes. En lugar de ello se pueden ver escenarios en 3D. Con hardware de entrada como un ratón es posible moverse dentro los diferentes escenarios.

Los archivos VRML pueden ser incorporados en archivos HTML.

1.7.8. XML

Supongamos que alguien hubiera programado alguna vez un archivo patrón para un programa de procesador de texto como MS Word. Sigamos suponiendo que ese archivo patrón se extiende por todo el mundo y cada persona lo tuviera y sólo usara ese patrón.

Esto es exactamente lo que ha pasado con HTML. HTML es un lenguaje que es definido con ayuda de SGML y que contiene muchos comandos. Es un lenguaje que contiene todos los comandos más importantes que un autor Web necesita, pero que no le da la oportunidad de inventar comandos propios.

Esta es la diferencia entre HTML y XML. XML es definido con ayuda de SGML la madre de todos los lenguajes de marcación. Sin embargo no es una "aplicación SGML", sino un "perfil de SGML". Este es un nivel abstracto, se habla también de Marcación Generalizada.

Con ayuda de XML es posible "inventar" lenguajes propios. XML existe tan sólo para definir lenguajes de marcación. Un tal lenguaje debe tener en cuenta determinadas convenciones fundamentales. Los comandos del lenguaje deben ser definidos en XML según un esquema prescrito. Es absolutamente posible asignar nombres propios. De esta manera es posible por ejemplo crear comandos para marcar zonas del documento como glosario, como nombres geográficos o como nombres de productos en instrucciones de uso. Al definir los comandos hay que determinar exactamente que cualidades deben poseer los comandos, por ejemplo dentro de que otros comandos del lenguaje el comando puede aparecer y dentro de cuales no.

Cuando se ha definido un documento XML con la definición del lenguaje propio, entonces es posible crear los archivos en el lenguaje propio, que es especificado en la cabecera del archivo con ayuda de un comando de referencia y que señala a que tipo de XML pertenece. Pues este es el único lugar en donde el software puede darse cuenta cuales son las reglas que son válidas en el lenguaje. Este puede ser por supuesto un sitio de acceso central en Internet o Intranet.

XML mismo es un lenguaje lógico como HTML. Con XML es posible definir comandos y sus cualidades, sin embargo no es posible determinar como el software de interpretación debe formatear un comando al momento de la visualización.

II.1. ANTECEDENTES

II.1.1. Historia

La World Wide Web ha estado viviendo un proceso de cambios y revoluciones informáticas sin precedentes, en tiempos muy cortos todo esto ha desembocado en la aparición y creación de nuevos servicios para usuarios de todo tipo y con estos servicios, se han desarrollado también múltiples herramientas y tecnologías para no sólo satisfacer esa gran demanda del mercado Web, sino para hacer más poderosas y eficientes las aplicaciones utilizadas. Una de estas nuevas herramientas es la Gestión de Información Abierta (OIM Open Information Management), que significa administrar la información de forma que este abierta al procesamiento por cualquier programa y no sólo por el que lo creó. Incluye las aplicaciones de programa que no fueron creadas en el momento en el que lo fue la información.

OIM se basa en el principio de la independencia de los datos: deberían almacenarse en las computadoras en representaciones que no fueran registradas, sino exclusivamente normalizadas, incluso cuando los datos constituyan el contenido del documento. Su representación debería distinguir la información propiamente dicha de los códigos patentados de programas de procesamiento, de documento y de elementos definidos de estilos de presentación.

Hace décadas que las bases de datos empresariales establecen una separación clara entre los datos reales y los formularios de entrada e informes de salida, han logrado independencia de los datos. Pero los documentos, a diferencia de los datos empresariales, se han creado históricamente en el contexto de un estilo de presentación de salida específica. Lo que explica que la independencia de los datos de documentos sólo se haya alcanzado en fechas muy recientes. En un hecho lamentable por dos razones: en primer lugar, por que los documentos son una fuente de información para la humanidad mucho mas significativa y capaces de almacenar estructuras de información mucho mas ricas que la base de datos.

En segundo lugar la necesidad actual del OIM en documentos es ahora mayor. La demanda de "redefinición" exige que la información sea entregada en varios formatos: en papel, de forma interactiva, multimedia o hipermedia. Además, la información debe entregarse a través de diversos canales: librerías y bibliotecas tradicionales, la Web mundial, Intranets y Extranets corporativas, la ultima tendencia es que la base de datos se convierta en un documento en el que se pueda navegar, pero que después el lector pueda reutilizar como datos.

Afortunadamente, en los últimos diez años ha surgido una tecnología que introduce la característica en independencia de los datos, propia de las bases de datos, en los documentos, pero sin las limitaciones que estas imponían en cuanto a libertad estructural. Se ha denominado a esta tecnología "Lenguaje de Mercado Estandarizado y Generalizado" (SGML)¹¹, que ha sido adoptada de forma generalizada tanto por la mayoría de los creadores de documentos como para la red mundial. Con SGML los organismos de gobierno, compañías aerospaciales, editoriales, de electrónica e informática, por citar algunas; han conseguido cortar la relación de dependencia que mantenían con el software de procesamiento. SGML coexiste junto a los gráficos, multimedia y demás estándares de datos necesarios en la actualidad y actúa como marco que vincula diferentes objetos entre sí y a documentos SGML.

HTML el lenguaje de marcas basado en SGML que más se utiliza en la World Wide Web, permite insertar menús, tablas, imágenes o bases de datos en los documentos, pero no permite al usuario que maneje dichos elementos como mejor le convenga.

Como HTML se ha convertido en un lenguaje de marcas de inmensa popularidad durante estos últimos años. Los usuarios de este lenguaje se han encontrado con sus propias limitaciones, algunas de ellas se han querido subsanar con la incrustación de herramientas y tecnologías nuevas como *Active X*, *scripts*,

¹¹ SMGL se trata en el apartado "SGML Standard Generalized Markup Language" del capítulo I.

*javascrip*ts, *HTML dinámico*, *hojas de estilo en cascada (CSS)*, etc. Todo esto es insuficiente para crear una arquitectura abierta de tipo cliente/servidor que en la actualidad se requiere.

Por lo anterior, el W3C, organismo que vela por el desarrollo de la Word Wide Web, se ha replanteado crear un nuevo estándar llamado "eXtensible Markup Language", abreviado como XML y que significa Lenguaje de Marcas Ampliable o Extensible, que parte de las amplias especificaciones del SGML (Standard Generalized Markup Language).

XML es el fruto de una larga y agitada historia de Sistemas de Procesamiento de Texto¹², desarrollado por un "Grupo de Trabajo XML" (originalmente conocido como SGML Editorial Review Board) y formado bajo la auspicio del "World Wide Web Consortium" (W3C) en 1996.

Al hablar de XML, es probable que se tenga la idea de que es una especie de HTML avanzado. El mayor error consiste en considerar a XML como una versión extendida de HTML. Pero como la mayoría de usuarios en Internet, el único lenguaje de marcas que conocen es el HTML, da por hecho que XML es una ampliación de HTML.

XML fue desarrollándose por el "Grupo de Trabajo XML" desde 1996. La especificación XML 1.0 fue ratificada y publicada como Recomendación por el W3C el 10 de febrero de 1998, e interpretado como "un sistema para definir, validar y compartir formatos de documentos en la Web". El 6 de octubre del 2000, fue publicada una segunda edición de la Recomendación de XML 1.0, que no es una segunda versión de XML de la publicada el 10 de febrero de 1998. Esta nueva edición, simplemente incorpora los cambios dictados por la errata de la primera edición.

XML se basa en la simplicidad y en una estructura abierta. Por ejemplo, cuando un creador Web diseñaba un programa de compra para una tienda virtual, basándose en HTML, invertía un arduo trabajo que muchas veces no podía exportarse para su uso por otras tiendas. Quizá su diseño estaba previsto para Windows NT y no funcionaba en plataformas UNIX, o hacía uso de ciertos lenguajes de programación como Perl pero no se integraba con Java. La implantación de XML viene a eliminar este tipo de situaciones permitiendo la creación de herramientas más estructuradas. Pero sin la dificultad ni complejidad de que tiene SGML, generando un lenguaje más fácil de entender

II.1.2. Definición

Se trata de un metalenguaje que ordena, estructura y describe los documentos de las páginas Web, permitiendo una descripción más minuciosa. En un principio, no rivalizarán HTML y XML, por el contrario, se complementarán el uno con el otro trabajando juntas ambas gramáticas. Este Lenguaje de Marcas eXtensible (XML), se puede considerar como una versión abreviada del SGML (Standard Generalized Markup Language).

En primer lugar hay que olvidarse un poco de HTML (sólo un poco) para entender mejor a XML. En la situación actual, en teoría, HTML es un subconjunto de SGML especializado en presentación de documentos para la Web, mientras que XML es un subconjunto de SGML especializado en gestión de información para la Web. En la práctica, HTML está un poco dentro de XML (que a su vez es parte de SGML) y otro poco fuera de SGML. Para reconducir esta situación, el grupo W3C ha dictado reglas expresas para distinguir el HTML que sigue al pie de la letra las normas de XML, denominándolo XHTML (eXtensible HyperText Markup Language), que no es más que una reformulación de HTML 4¹³ dentro de las normas de XML.

Entonces para poder definir que representa XML, es necesario exponer lo que el W3C dice acerca de lo que XML es. Ya que este consorcio es el creador y encargado de generar y regenerar las normas que rigen al lenguaje.

¹² El capítulo I, dentro del apartado "Lenguajes de Mercado", explica los Sistemas de Procesamiento de Texto.

¹³ HTML versión 4 se explica en el apartado "HTML HyperText Markup Language" del capítulo I.

II.1.3. Objetivo

Dentro de la especificación XML que el W3C publica en Internet, se definen las metas diseñadas que persigue el lenguaje en los siguientes puntos:

- XML debe ser directamente utilizable sobre Internet.
- XML debe soportar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- El número de características opcionales en XML debe mantenerse absolutamente al mínimo, idealmente cero.
- Los documentos XML deben ser legibles para el usuario y razonablemente claros.
- El diseño de XML debe ser preparado rápidamente.
- El diseño de XML debe ser formal, conciso.
- Los documentos XML deben ser fáciles para 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 identificación de lenguajes, ISO 639 para códigos de nombres de lenguajes y también el ISO 3166 para códigos de nombres de países, para la comprensión normal de esta versión de XML”¹⁴.

II.1.4. Importancia

La virtud de XML es que los datos XML son *inteligentes*. HTML se preocupa por el aspecto de los datos, mientras que XML por su significado.

Con XML, el navegador sabe que existe un producto y está informado acerca de su modelo, vendedor, precio, entre otros datos. A partir de un grupo como este, puede comunicarnos cual es el producto mas barato o el vendedor más cercano sin que tengamos que volver al servidor.

A diferencia de HTML, con XML se pueden crear etiquetas propias que describan con precisión lo que se desea saber. Por ese motivo, las aplicaciones pueden tener acceso a cualquier fuente de información de la Web y en cualquier formato. Los servidores “de nivel intermedio” se ubican entre las fuentes de datos y el cliente, traduciéndolo todo a XML de una tarea específica.

Pero los datos XML no son solamente datos inteligentes, también son documentos inteligentes: al visualizar la información, el nombre del modelo puede aparecer con una fuente diferente a la del nombre del vendedor o con el precio mas bajo resaltado en color verde. A diferencia de HTML, el texto no tiene por qué ser presentado de forma uniforme. El texto *inteligente* XML puede controlar el contenido a transmitir.

No es necesario elegir entre información de datos o de documentos, XML integra las dos cosas. Es posible procesar o datos o documentos o ambos simultáneamente.

La grandeza de XML es que el mismo software puede procesar toda una diversidad de información. Lo que se puede hacer con un tipo de documento se puede hacer con todos los demás.

Por ejemplo, una transacción por correo electrónico, como una compra, puede requerir varios tipos de documentos. El comprador podría empezar mandando varios documentos a un vendedor:

- Nota adjunta: carta.
- Pedido de compra: formulario.
- Especificación de producto: documento extenso.

¹⁴ Esta definición se encuentra dentro de las páginas del W3C en: <http://www.w3.org/TR/REC-xml>

El vendedor a su vez respondería con varios documentos:

- Acuse de recibo: mensaje.
- Nota de agradecimiento: carta.
- Factura: formulario.

Así, para XML el momento en el que se necesitan herramientas adicionales es cuando se pretende alcanzar diferentes objetivos y no cuando se quiere trabajar con diferentes tipos de documentos.

Otra cuestión que se debe aclarar es que XML es más un sistema complejo de tratamiento de información que un simple lenguaje de descripción. XML es más una familia de lenguajes y al igual que se puede decir que HTML es un "lenguaje", para ser exactos, habrá que definir a XML como un "metalenguaje", es decir, un lenguaje capaz de definir otros lenguajes.

XML sirve para la representación digital de los documentos, éstos pueden ser extensos o cortos. Tanto una enciclopedia en varios volúmenes como un recordatorio pueden visualizarse como documentos. Un volumen en concreto de una enciclopedia también puede llamarse un documento.

XML permite pensar en una enciclopedia del modo que sea, con tal de que el trabajo por realizar se haga de la manera más eficaz. XML también permite pensar en un mensaje de correo electrónico como un documento.

Cuando se dice que se pretende representar digitalmente los documentos, se quiere decir que se desea transformarlos en algún tipo de código legible por la computadora para que esta sea capaz de almacenarlos, procesarlos, buscarlos, transmitirlos, mostrarlos, imprimirlos, o lo que sea que el usuario requiera.

Para que una computadora haga cosas útiles con un documento, se debe primero darle datos sobre la estructura del documento. EL objetivo básico es el traducir dichos documentos a un código *comprensible* por la computadora, considerando siempre lo que comprensible significa para una computadora.

Los documentos XML incluyen o pueden incluir fotografías, películas, y otros elementos multimedia, si bien no se representan los componentes multimedia en XML. Esto es, si se piensa en la representación como un proceso de traducción, similar a la traducción de idiomas, los componentes multimedia se dejarían en la lengua de origen porque no tienen traducción simple en la lengua destino (XML).

Los elementos multimedia se incluirían en su formato de origen al igual que no se haría una traducción explícita de una expresión latina en un texto en español. Por ejemplo, la mayoría de las fotos que aparecen en la Web, son archivos en formatos GIF o JPEG y la mayoría de películas tienen el formato MOV o MPEG. EL documento XML se refiere a esos archivos en sus formatos GIF, JPEG, MOV o MPEG originales. Si se transcribe un documento impreso en XML, lo más probable es que se representarían las partes de caracteres de texto en XML y las partes gráficas en los demás formatos antes mencionados.

II.2. UTILIZACIÓN

II.2.1. Editores

Para trabajar con XML no son necesarias muchas herramientas, aunque existen muchas actualmente y se están desarrollando cada vez más. La idea general, es similar al HTML. Primero, se necesita generar el documento XML y después visualizarlo o utilizarlo con algún programa. Pero además, se necesita una herramienta para verificar la validez y buena formación de los documentos.

Dado que los documentos XML, son simplemente texto ASCII, casi cualquier editor de texto puede generar un documento XML. Al igual que con HTML existen programas editores que sirven para crear páginas Web HTML; también existen editores para crear documentos XML.

Los editores son aquellos programas que nos permiten escribir los documentos XML. Dentro de estos se puede encontrar los básicos editores de texto como el Notepad de Windows, o algunos más complejos, que permiten saber si el documento XML escrito está bien formado e incluso si es válido, es decir, realizan el parseo automáticamente. Algunos de estos editores pueden soportar también XSL, DTDs, SCHEMAS, HTML, XHTML, y algunas otras herramientas¹⁵.

II.2.2. Navegadores – Browsers

El navegador o browser es el software que permite ver las páginas HTML y también XML. Por el momento el único navegador, de los que se utilizan comercialmente, que permite trabajar con XML de una forma razonable es el Internet Explorer "IE5". Hay que tener en cuenta que las versiones anteriores de éste navegador no soportan XML.

Antes de querer visualizar estos documentos, hay que asegurarse que el navegador soporte XML. También existen otros navegadores especializados que permiten trabajar con algunas aplicaciones de XML concretas.

II.2.3. Parsers XML

Los Parsers sirven para comprobar que los documentos están bien formados, que están validados por el DTD/Schema correspondiente, aplicar las hojas de estilo, etc. Es el encargado de verificar la validez de los documentos. Hay parsers escritos en múltiples lenguajes, en función de la necesidad de la aplicación que se quiera desarrollar se utilizará uno u otro.

Algunos de los Parsers más populares son:

- Parsers XML: SAX, XP.
- Parsers XSL: XT.

Los procesadores suelen ser programas o librerías de funciones (APIs), escritas en distintos lenguajes que hacen uso de estos parsers y que sirven para montar la aplicación XML.

Actualmente, el Internet Explorer 5 (IE5) se distribuye con intérpretes de XML. El intérprete de IE5 es un analizador sintáctico (parser o validador) de alto desempeño que incluye soporte completo para la Recomendación DOM del W3C. Además, contiene un procesador XSL basado en un borrador de estándar actual del W3C, un Objeto Fuente de Datos XML de C++ de alto desempeño y la habilidad de navegar archivos XML directamente.

II.2.4. Aclaraciones

La existencia de XML no implica que desaparecerá el HTML. Dado que la mayor parte de las páginas Web son documentos de texto con algunas imágenes, HTML seguirá siendo el medio más eficaz para crearlas y publicarlas, aunque muy probablemente se tenderá a utilizar la versión XHTML.

XML puede convivir con los restantes lenguajes, tales como HTML, JAVA, JavaScript, VBScript, Visual Basic, etc. En muchos de estos casos, XML hará las veces de "base de datos", los scripts servirán para realizar búsquedas selectivas y HTML permitirá que se muestre la información resultante en la pantalla.

Hasta que no se publiquen todas las recomendaciones oficiales de los estándares que complementan a la de XML, no "explotará" la publicación de páginas XML en Internet, ya que todavía se modificarán algunas cuestiones importantes:

Aunque la versión 5 del Internet Explorer ya permite trabajar con XML y existen otros visualizadores compatibles con el nuevo estándar. El respaldo definitivo se dará cuando todos los navegadores lo acepten, sobre todo Navigator de Netscape. XML se utiliza principalmente por los profesionales, debido a la

¹⁵ Se pueden encontrar un número considerable de estos editores en: <http://www.xmlsoftware.com/editors/>

complejidad de las especificaciones, pero según vayan saliendo al mercado editores WYSIWYG (editores que visualicen el resultado mientras se edita) que permitan crear documentos XML, se podrá utilizar por cualquier persona, de forma similar a como se trabaja con los editores de textos actuales.

II.3. ESTRUCTURA Y CONCEPTOS BASICOS

Aunque a primera vista, un documento XML puede parecer similar a HTML, hay una diferencia principal. Un documento XML contiene datos que se autodefinen, exclusivamente. Un documento HTML contiene datos mal definidos, mezclados con elementos de formato. En XML se separa el contenido de la presentación de forma total.

Una forma de entender rápidamente la estructura de un documento XML, es viendo un pequeño ejemplo. El documento que se utilizará se trata de un Mensaje de correo electrónico. El mensaje consta principalmente de 4 partes: Remitente, Destinatario, Asunto y Texto. A su vez las partes de Remitente y Destinatario se componen de un nombre y una dirección de Email.

Remitente Nombre:Eduardo Najera Email: lalo@myemail.com
Destinatario: Nombre: Vicente Fox Email: fox@presidencia.gob
Asunto: Saludos, Sr. Presidente
Texto de Mensaje: ¡Hola! Cordialmente lo invito al próximo juego en Dallas...

Para representar estas partes dentro del documento XML, se utilizan etiquetas que identificarán a cada elemento en un inicio (donde principia) y un final (donde termina).

Es importante aclarar antes de continuar, que éste es un ejemplo sencillo y que para crear correctamente un documento XML habrá que hacer referencia siempre a la recomendación y especificación oficial del consorcio del W3C¹⁶. Como nota, en las páginas siguientes se explica mas a fondo este tema.

Para este ejemplo una etiqueta de inicio es un texto delimitado entre los símbolos *menor que* y *mayor que* "<" y ">". Una etiqueta de fin es el mismo texto delimitado entre los símbolos *menor que* y *mayor que* "<" y ">" pero con una diagonal invertida "/" antes del texto.

Se tiene entonces la primera etiqueta que representa el mensaje. Se inicia justo donde se localiza la etiqueta <mensaje> y finaliza donde se encuentra la etiqueta </mensaje>. Las etiquetas que representan las 4 partes del mensaje son: <remite>, <destinatario>, <asunto> y <texto> y terminan en </remite>, </destinatario>, </asunto> y </texto> respectivamente. Como se mencionó anteriormente el remitente y destinatario a su vez tienen un Nombre y un Email, así, el texto ubicado entre las etiquetas <nombre> y </nombre> indica el nombre y el texto entre las etiquetas <email> y </email> indica la dirección de correo.

Las etiquetas <nombre> ... </nombre> y las etiquetas <email> ... </email> se encuentran entre las etiquetas <remitente> y </remitente>, esto indica que el Nombre y el Email pertenecen al remitente y no al destinatario, dado que el nombre del destinatario con su Email están dentro de las etiquetas <destinatario> y </destinatario>.

¹⁶ La especificación oficial (completa) acerca de la estructura, terminología y gramática de XML se encuentra dentro de la recomendación publicada por el W3C. En: <http://www.w3.org/TR/2000/REC-xml-20001006>

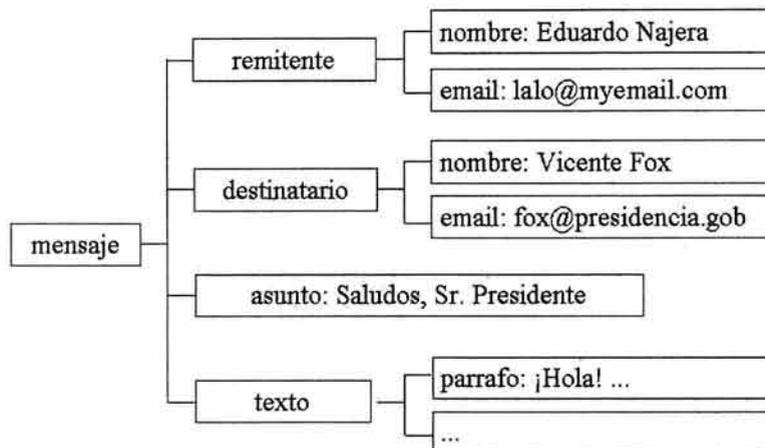
Es importante mencionar que en este ejemplo cada etiqueta de apertura debe tener una etiqueta de finalización o cierre. Por ejemplo la etiqueta <mensaje> debe terminar con la etiqueta </mensaje>. Se debe tener cuidado de no cruzar las etiquetas, es decir, no cerrar una etiqueta sin antes cerrar todas las etiquetas que se encuentren entre la etiqueta de inicio y la de fin. Por ejemplo: <mensaje> <remite> </mensaje> </remite>.

El documento queda entonces de la siguiente forma:

```
<?xml versión="1.0"?>
<!DOCTYPE MENSAJE SYSTEM "mensaje.dtd">
<mensaje>
<remite>
<nombre>Eduardo Najera</nombre>
<email>lalo@myemail.com</email>
</remite>
<destinatario>
<nombre>Vicente Fox</nombre>
<email>fox@presidencia.gob</email>
</destinatario>
<asunto> Saludos, Sr. Presidente</asunto>
<texto>
<parrafo>¡¡Hola! <énfasis>Cordialmente</énfasis> lo invito al
próximo juego en Dallas...</parrafo>
</texto>
</mensaje>
```

Las primeras dos líneas no se explican en este ejemplo, quedan señaladas solamente para familiarizarse con ellas y se explican posteriormente

Este mismo documento puede ser visto de forma gráfica, para comprender mejor la estructura del documento XML que se acaba de explicar.



II.3.1. Gramáticas y representación

Las sintaxis de los lenguajes naturales se describe por medio de una gramática, la sintaxis de XML también. La mayoría de representaciones de documentos empiezan con una cabecera que contiene información acerca del documento real y como debe interpretarse esa representación. Después le sigue la representación del documento real.

Por ejemplo HTML tiene un elemento HEAD que contiene el TITLE y los elementos META, a HEAD le sigue el elemento BODY, ahí es donde reside la representación del documento real. De forma similar, los mensajes del correo electrónico tienen "líneas de cabecera" donde se describe de donde procede el mensaje, a quién va dirigido, cómo está codificado, etc.

Un documento XML también se divide en dos partes principales; un prólogo y una instancia de documento. El prólogo proporciona información acerca de la interpretación que debe hacerse de la instancia del documento, como la versión XML de la que se trata y el tipo de documento al que corresponde. La instancia del documento viene después del prólogo. Contiene los datos reales del documento como una jerarquía de elementos.

II.3.2. Estructura

Aunque no es obligatorio, los documentos XML pueden empezar con unas líneas que describen la versión de XML, el tipo de documento, y otras cosas.

La primera, o "declaración XML", define la versión de XML usada. Hasta ahora sólo hay una, la "1.0". Además, en la "declaración XML" especificamos la codificación del documento, que puede ser, por ejemplo, US-ASCII (7 bits) o UTF-8 (código Unicode del que el ASCII es un subconjunto), UCS-2, EUC-JP, Shift_JIS, Big5, ISO-8859-1 hasta ISO-8859-7. En general y para uso con lenguajes europeos (incluyendo el juego de caracteres especiales del castellano) se utiliza UTF-7 o ISO-88591.

Además, se puede incluir una declaración de documento autónomo (standalone), que controla qué componentes de la DTD¹⁷ son necesarios para completar el procesamiento del documento.

Un ejemplo de lo mencionado es la línea siguiente:

```
<?xml version="1.0" encoding="UTF-7" standalone="yes"?>
```

La segunda, o "declaración de tipo de documento", define qué tipo de documento se está creando para ser procesado correctamente. Es decir, se declara que Definición de Tipo de Documento o DTD (Document Type Definition) válida y define los datos que contiene el documento XML.

En ella se define el tipo de documento y dónde encontrar la información sobre su Definición de Tipo de Documento, mediante un identificador público (PUBLIC) que hace referencia a dicha DTD, o mediante un Identificador Universal de Recursos (URI) precedido por la palabra SYSTEM.

Ejemplos:

```
<!DOCTYPE MENSAJE SYSTEM "mensaje.dtd">  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  
<!DOCTYPE LABEL SYSTEM "http://www.empresa.com/dtds/label.dtd">
```

Un documento XML tiene dos estructuras, una lógica y otra física. Físicamente el documento está compuesto por unidades llamadas entidades. Una entidad puede hacer referencia a otra entidad, causando que ésta se incluya en el documento. Cada documento comienza con una entidad documento, también llamada raíz. Lógicamente, el documento está compuesto de declaraciones, elementos, comentarios, referencias a caracteres e instrucciones de procesamiento, todos los cuales están indicados por una marca explícita. Las estructuras lógica y física deben encajar de manera adecuada.

Antes de poder proceder a una descripción exacta de como se van a representar los documentos, se debe tener en mente un modelo de estructuración del documento. La mayor parte de los documentos (por ejemplo los libros y revistas) se pueden dividir en componentes (capítulos y artículos) que a su vez se componen de elementos (títulos, párrafos, figuras, etc.). todos los documentos pueden considerarse de esta forma.

En XML, estos componentes reciben el nombre de "elementos". Cada elemento representa un componente lógico del documento. Los elementos pueden constar de otros elementos así como de frases y palabras que normalmente se consideran como el texto del documento. XML denomina este texto los "datos de carácter" de documento.

¹⁷ Las Definiciones de Tipo de Documento o DTD (Document Type Definition) se explican en el tema siguiente

Los profesionales del medio la denominan “*estructura en árbol del documento*”. El elemento que contiene a los demás (por ejemplo, libro, informe o nota) recibe el nombre de *elemento raíz*. Este nombre indica que es el único elemento que no depende de otro. Los elementos incluidos en la raíz se llaman “*subelementos*”, que pueden contener a su vez otros “*subelementos*”. Si es así, se denominan “*ramas*”. De lo contrario, se llaman “*hojas*”.

Por tanto, los elementos: capítulo y artículo son ramas (porque cuentan con *subelementos*) y en cambio los elementos párrafo y título son hojas (porque solo contienen *datos de carácter*).

En ocasiones los elementos incluyen información adicional llamada “*atributos*”. Los *atributos* describen las propiedades de los elementos. Por ejemplo, un elemento de registro CIA. Incluye un atributo de seguridad que indica el nivel de seguridad para ese elemento. Una base de datos de la CIA sólo dará acceso a determinados registros al personal autorizado que supere el nivel de seguridad establecido. No queda siempre muy claro qué aspectos de un documento deben representarse con elementos y que otros con atributos.

En la práctica, los documentos no siempre siguen a la perfección este modelo de árbol. Constan a menudo de características no jerarquizables, como remisiones o enlaces de hipertexto de un artículo a otro del árbol. XML es capaz de representar estas estructuras de hecho, XML va más allá de los enlaces de HTML.

El contenido real de un documento XML se encuentra dentro de la “*instancia del documento*”, se denomina así porque si tuviera una DTD se trataría de una “*instancia de clase*” de los documentos definidos por la DTD. Es lo mismo que decir que un individuo es una instancia de la clase “*personas*” o un memorándum una instancia de clase “*documentos de memo*”.

Puesto que una computadora no puede comprender los datos de un documento, lo primero que mira son las etiquetas, es decir, el marcado entre los símbolos de *menor que* y *mayor que* (“*<*” y “*>*”). Las etiquetas delimitan el principio y el final de varios elementos. De esta forma la computadora puede considerar los elementos como si se tratara de un árbol.

Las computadoras no saben nada de semántica, no saben lo que es un protocolo HTTP, a partir de un supermodelo, los diseñadores de tipos de documentos, deben describir las semánticas para los autores de otro modo. Por ejemplo pueden enviar un mensaje de correo electrónico, escribir un libro o hacer una secuencia de video. Lo que le interesa a la computadora es el aspecto que supone debe tener un elemento cuando esta formateado, la forma en que debe actuar, si es interactivo o que hacer después de extraer los datos, esas cuestiones se encargan de especificarlas las hojas de estilo y los programas informáticos.

Los textos están formados por caracteres. Para representar textos, se debe primero representar los caracteres que los componen y para ello, decir como representarlos en los niveles de bits y bytes. A este proceso se le denomina codificación de caracteres. Se debe decidir también que tipo de caracteres se introducirán en los documentos. Un grupo de caracteres especialmente restrictivo, solo permitirá caracteres en mayúsculas. En cambio un amplio grupo de caracteres permitirá la inclusión de ideogramas orientales y caracteres árabes.

Para los anglófonos, es suficiente con disponer de los 52 caracteres, mayúsculas y minúsculas, y de algunos símbolos de acentuación y puntuación. El grupo de caracteres ASCII de bits domina el mercado. Contiene el número suficiente de caracteres (128) para todas las letras, símbolos, signos de puntuación y demás tipos de caracteres. ASCII es tanto un grupo de caracteres como una codificación de caracteres. Define los tipos de caracteres disponibles y como deben codificarse en términos de bit y bytes.

El grupo de caracteres de XML es UNICODE, una especie de ASCII. UNICODE incluye miles de caracteres comunes de lenguas de todo el mundo. Los 128 primeros caracteres de UNICO son compatibles con ASCII. Existe una codificación de caracteres de UNICODE UTF-8 compatible con ASCII de 7 bits lo cual significa que en los niveles de bits y bytes, los 128 primeros caracteres de UNICODE UTF-8 y ASCII de 7 bits son iguales. Es una característica de UNICODE que permite a los autores utilizar editores de solo texto estándares para crear XML de forma inmediata.

En la actualidad el mundo de habla inglesa domina la Web, pero otras regiones con distinto idioma van extendiéndose en el uso de la Web. XML figura entre las tecnologías que permiten a los sitios que manejan

diversos idiomas tomar contenido de diversas bases de datos y transmitirlo a un mecanismo de traducción que genera el grupo de caracteres y la sintaxis correcta para los visitantes que no hablan inglés.

El documento XML se define por una serie de caracteres. El procesador XML empieza al principio y trabaja hasta el final. XML incluye un mecanismo que permite al texto organizarse de forma no lineal y en varias partes. El analizador sintáctico lo reorganiza en una estructura lineal.

El elemento "trozo de texto" se puede denominar como una "entidad". Una *entidad* puede consistir en un solo carácter o abarcar todos los caracteres de un libro.

Las entidades tienen nombres, para poder utilizar una entidad es necesario insertar una referencia de entidad en alguna parte del documento. Así el procesador podrá reemplazar la referencia de entidad por la entidad misma, que se denomina texto de reemplazo. Funciona más o menos como una macro de procesador de texto. Por ejemplo, supóngase que una entidad llamada "sigma" contiene el nombre de un carácter griego. Se utilizaría una referencia de la entidad cada vez que se deseara insertar el carácter sigma en el documento. Una entidad también podría llamarse "capítulo de introducción" y ser el capítulo de un libro. Se hace referencia a la entidad cada vez que se requiera que apareciera el capítulo dentro del documento.

Una de las perspectivas que más entusiasman a Theodor Nelson el hombre que acuñó la palabra hipertexto¹⁸, era pensar que el texto podría reutilizarse automáticamente en diferentes contextos. La actualización en un lugar contagiaría a todas las demás utilidades del texto. Con XML, esto se ha convertido en una realidad, gracias a la opción llamada entidad externa. A menudo se hace referencia a las entidades externas con solo decir entidades.

El documento XML, puede dividirse en muchos archivos de un disco duro, u objetos en una base de datos y cada uno de ellos se denomina entidad en la terminología XML. Las entidades pueden incluso difundirse en Internet. Al igual que los elementos XML describen la estructura lógica del documento, las entidades siguen la pista de la ubicación de los conjuntos de bytes que constituyen el documento XML. A esto se le denomina estructura física del documento

Por ejemplo, las entidades pueden guardarse en bases de datos o ser creadas de carrera por un programa de computadora. Algunos formatos de archivo como un archivo ZIP, permiten incluso a varias entidades, ocupar el mismo archivo simultáneamente. El término que admite estas posibilidades es entidad y no archivo. Además en la mayoría de los sitios Web, cada entidad permanece en un solo archivo por tanto, en esos casos, las entidades externas y los archivos son funcionalmente lo mismo. Es esta una configuración simple y eficaz pero insuficiente para grandes sitios.

El pan de cada día de las entidades no es tan excitante como reutilizar trozos de textos en Internet, pero es igual de importante: ayudan a dividir los archivos extensos para hacerlos editables, buscables, descargables y utilizables por los sistemas comunes de computadoras. Las entidades permiten a los autores partir los documentos en fragmentos que se adapten a la memoria para su edición; puedan descargarse mediante un módem lento, etc.

Sin las entidades, los autores deberían dividir los documentos en documentos más pequeños de forma arbitraria y con frágiles enlaces (como suele ocurrir con HTML). Eso complicaría la gestión del documento. Las entidades permiten fragmentar los documentos en trozos sin olvidar que forman un todo coherente que puede ser impreso, editado, y buscado como unidad cuando fuera necesario. Los objetos que no son XML se mencionan de forma muy similar y se llama *entidades no analizables sintácticamente*. Se consideran "entidades de datos" porque carecen de cualquier tipo de marcado XML, las entidades de datos incluyen los gráficos, películas, audio, texto virgen, PDF y todo lo demás que no sea XML (incluido HTML y demás formas de SGML). Cada entidad de datos dispone de una notación asociada que es simplemente una frase aclarando si se trata de un GIF, JPEG, MPEG, PDF, etc.

Aunque existan procesadores de texto XML, uno de los objetivos perseguidos al diseñar XML era que fuera capaz de crear documentos XML en editores de texto estándares. Mucha gente desconfía de los procesadores de texto, pero incluso los que si confían dependen de los editores de texto para "depurar" el documento en caso de que el procesador de texto haga o permita al usuario cometer error. La única forma

¹⁸ Hipertexto y la aportación de Theodor Nelson se trata en el apartado "Lenguajes de Marcado" dentro del capítulo I.

de que los autores tengan un acceso adecuado tanto a la estructura como a los datos del documento en editores de texto estándares, es colocando a ambos frente a frente.

Lo que representa la estructura lógica y conecta a las entidades se conoce como marcado. Un documento XML esta compuesto exclusivamente de marcado y datos de caracter ambos en UNICODE y ambos conocidos como texto XML.

La diferencia entre el marcado y los datos de caracter estriba en que el primero incluye caracteres especiales llamados delimitadores. El texto incluido entre el caracter menor que "<" y mayor que ">" o entre el signo "&" y el punto y coma ";" es el marcado. Estos cuatro caracteres son los delimitadores más comunes.

Existen entidades predefinidas que los usuarios utilizan para representar los caracteres que de otro modo se interpretarían con un significado especial, como una etiqueta de inicio o una referencia de entidad. La tabla siguiente refiere a cinco de estas entidades predefinidas (incorporadas en XML) que permiten resolver de forma precisa el problema del usuario con la interpretación.

Referencia de entidad	Carácter
&	&
<	<
>	>
'	'
"	"

II.3.3. Tipos de Documento

Los documentos XML se dividen en dos grupos, los bien formados y los documentos válidos.

Una *definición de tipo de documento* conocido como DTD (Document Type Definition) es una serie de definiciones para tipos de elemento, atributos, entidades y notaciones. Determina cuales son legales dentro del documento y en que lugar lo son. Un documento XML establece su DTD particular en la declaración de tipo de documento. Las DTD son herramientas de estandarización organizativa tan potentes como los formularios, las plantillas y guías de estilo. Una DTD muy estricta que solo permita a un tipo de elemento en un determinado lugar, es como un formulario "rellena solo los espacios en blanco". Una DTD más flexible es como una guía de estilo en cuanto que por ejemplo, puede exigir que en cada lista haya dos o más artículos, en cada informe una abstracción y que las notas a pie de página sólo aparezcan dentro de la sección "a pie de página".

Las DTD son esenciales para la estandarización organizativa, ya que posibilitan el procesamiento robusto de los documentos por software. Por ejemplo, un documento carta con un *capítulo* en el medio no tendría mucho sentido. El software de impresión de carta no imprimiría con gran fiabilidad un documento así porque no esta muy clara la definición de capítulo dentro de una carta. Un caso mas patente sería la ausencia dentro del documento de un elemento que el software que lo procesa espera descubrir. Si un programa de correo utiliza XML como formato de almacenamiento, podrá buscarnos todas las direcciones de correo electrónico procedentes de una dirección determinada. Supongase que cada mensaje almacena esta dirección en un elemento *de*. ¿Que se haría entonces para buscar las cartas que no tuvieran esos elementos *de*? Los programadores podrían escribir un código especial para "rodear el problema", pese a que de esta forma se complica la escritura del código.

HTML es una historia que tiene su moraleja. Su estructura es bastante rigurosa, definida en SGML y disponible a partir del consorcio de la Web mundial W3C. Pero parece ser que todo el mundo respeta las normas como si las hiciera la Federación mundial de lucha libre, es decir nadie les hace caso.

Los programadores que mantienen los navegadores HTML invierten muchísimo tiempo dando soporte a todas las combinaciones que tiene la gente de los elementos HTML en los documentos. Aunque HTML tenga una DTD en SGML, casi nadie la utiliza; por eso, aunque de forma no oficial, los vendedores de navegadores han sancionado que se les haga caso omiso. La programación de rodeos es cara, exige mucho

tiempo, y además de ser monótona y poco gratificante, no existe una definición clara de lo que significan estos elementos ilegales. Algunos elementos erróneos provocan fallos en el navegador, pero otros solo les hacen mostrar resultados confusos o aleatorios.

En XML, la DTD es una definición formal de los tipos de elemento, atributos, y entidades tolerados en un documento de un tipo específico.

Existe otra cuestión de carácter más sutil. Si no se detiene a pensar sobre la estructura de los documentos, se corre el riesgo de especificarlos según su formato y no según su estructura abstracta. Cuando se piensa en los documentos, se suele asociarlos con su reproducción. Esto se debe a que, antes de GML, no había medios prácticos de crear un documento sin crear una reproducción. El proceso de creación de una DTD brinda la oportunidad de pensar en los documentos en términos de estructura, como abstracciones.

Algunos documentos XML carecen de la declaración de tipo de documento. Esto no implica que no se ajusten a un tipo de documento. Simplemente, significa que declaran no ajustarse a una definición de tipo de documento formalmente definida.

Si el documento ha de servir como documento XML deberá tener una estructura, expresada mediante elementos, atributos, etc. Cuando se crea una hoja de estilos para un documento depende de los elementos de dicha estructura, de los determinados significados de los nombres de tipo de elemento y de los elementos que figuran en determinados lugares. Cualquiera que sea su forma explícita, ésta sería de factores de la que se depende, conforme el tipo de documento.

La estructura puede formalizarse en la DTD. Además de o en lugar de una DTD formal legible por la computadora; cabe también la posibilidad de redactar una descripción en prosa. Se debe para ello considerar los muchos libros HTML que son descripciones en prosa de HTML. La última alternativa consiste en recordar el tipo de documento y respetarlo haciendo gala de mucha disciplina. Aunque, para documentos extensos y complejos, requiere tal poder de concentración que para conseguirlo hay que tener una gran memoria. Por eso es recomendable. Sobre todo en documentos extensos, identificar las partes del documento con ayuda de una DTD.

II.3.4. Buena formación y validez

Todos los lenguajes tienen sus reglas sobre lo que consideran válido o no válido. XML tiene dos nociones de "correcto". La primera se limita a que el marcado sea inteligible: el equivalente XML de "pronunciar correctamente" una palabra de un cierto idioma. El documento con marcado inteligible se denomina documento bien formado. Un objetivo importante de XML es que, para facilitar su adhesión, estas reglas básicas debían ser simples. La experiencia con el marcado HTML sirvió de lección para el desarrollo de XML mucho del HTML en la Web no se ajusta incluso a las reglas más básicas de las especificaciones HTML, lo que dificulta mucho su procesamiento automático.

Como los navegadores Web seguirán mostrando documentos mal formados, los autores siguen creándolos. Al diseñar XML se decidió que los procesadores XML debían prohibir todo intento por subsanar un error de buena formación en un documento XML. Fue una decisión que levanto mucha polémica, porque muchos pensaron que era una equivocación prohibir a los implementadores XML que siguiera la mejor política de solución de errores para sus aplicaciones.

El equivalente XML de "utilizar las palabras adecuadas en el lugar adecuado" se llama validez y remite a la noción de tipos de documento. Un documento es válido si se declara conforme a una DTD en una declaración de tipo de documento y se ajusta positivamente a la DTD.

Los documentos que carecen de declaración de tipo de documento no se consideran estrictamente inválidos (no infringen su DTD) pero tampoco son válidos ya que no pueden ser validados por su DTD.

Aunque se alteraran los documentos HTML con varios títulos para poder utilizar la sintaxis XML, serían *bien formados* e inválidos (suponiendo que se convirtiera también la DTD HTML por la sintaxis XML). Suprimiendo la declaración de tipo de documento para que no pudieran declararse por más tiempo conforme a la DTD HTML, serían simplemente bien formados pero ni válidos, ni inválidos.

Es recomendable reflexionar detenidamente antes de crear un documento bien formado pero no válido. Si el documento es de un tipo y poco extenso, bastará probablemente con que sea bien formado. Pero si ha de integrarse en un sistema de información (incluso uno pequeño) o si se trata de un documento extenso, es conveniente escribir una DTD para él y validar periódicamente el documento. Cuando se toma la decisión de crear o ampliar el sistema de información, es muy importante que el documento dé garantías de compatibilidad ya que eso facilitará en gran medida la programación o escritura de la hoja de estilos y la obtención de resultados fiables.

II.3.5. Información sintáctica

Los documentos XML están formados por caracteres procedentes del conjunto de caracteres UNICODE. Cada una de las secuencias de caracteres recibe el nombre de “cadena”, para entender este concepto, se pueden considerar los caracteres como una larga cadena de texto, cada capítulo es también una cadena, lo mismo que cada palabra. Y de forma similar los documentos XML se componen de cadenas insertadas de otras cadenas.

Los lenguajes naturales como el inglés tienen una sintaxis particular. Esta sintaxis permite a la comunidad de hablantes, combinar las palabras y formar oraciones gramaticales. XML también tiene una sintaxis propia,

XML es sensible al tipo de letra. Esto significa que si la especificación XML dice que debe introducirse la palabra “ELEMENTO”, se está afirmando que debe introducirse “ELEMENTO” y no “elemento”, ni “Elemento” o “EleMenTo”, para muchos usuarios, especialmente los usuarios del habla inglesa, la correspondencia, sin tener en cuenta el tipo de letra, es mucho más sencilla, que tener que recordar un tipo de letra especial de una determinada construcción.

Por ejemplo, si un tipo de documento tiene un tipo de elemento denominado *img*, por lo general los usuarios de la habla inglesa tienden a olvidarlo y a introducir en su lugar IMG. Tienen a confundirse porque no están acostumbrados a considerar el tipo de letra como algo importante, este hecho también explica que los usuarios, que acaban de incorporarse a Internet, tiendan a escribirlo todo con mayúsculas, la mayoría de aplicaciones de SGML, incluido el HTML están diseñadas para no tener en cuenta el tipo de letra, sus defensores afirman que de esta forma se evita que todo el tipo de letra se convierta en una fuente de errores.

Sus retractores sostienen, sin embargo, que obviar el tipo de letra supone un retroceso a la época de las perforadas. Pero a la vez afirman que la sensibilidad al tipo de letra es un concepto muy complejo en un conjunto universal de caracteres como UNICODE.

El usuario debe tener cuidado con no confundir el uso de mayúsculas y minúsculas. La mayoría de los errores que se comenten en XML, se refieren al uso de las mayúsculas y minúsculas. Lógicamente XML también es sensible a la ortografía y a la escritura, por lo que no debe descuidarse ninguno de estos aspectos cuando se trabaje con XML.

Debe observarse que aunque XML es sensible al tipo de letra, eso no significa que lo imponga. En todo momento, el usuario tiene total libertad para crear sus propios nombres o textos, elegir entre el uso de mayúsculas o minúsculas, por ello, aunque las palabras clave de XML deben introducirse tal y como aparecen escritas, las cadenas que componga el usuario, pueden combinar y hacer corresponder los caracteres en mayúsculas y minúsculas como guste.

Así, cuando un usuario crea sus propios tipos de documentos, puede elegir los nombres de tipos de elementos, un determinado nombre puede venir expresado en mayúsculas “SECCION”, y en minúsculas “sección”, o en una combinación de ambas. Pero puesto que XML es sensible al tipo de letra utilizado, todas las referencias a un tipo de elemento particular, tendrá que utilizar el mismo tipo de letra. Es aconsejable crear una convención propia, simple para todas las mayúsculas o todas las minúsculas, para no tener que recordar el tipo de letra original.

Las construcciones como etiquetas, referencias de entidad, y declaraciones se denominan marcas, estas son las partes del documento que se supone tiene que entender el procesador XML. Las partes que se

encuentran entre estas marcas son las que resultan comprensibles para el ser humano y constituyen lo que se denomina datos de carácter.

El marcado toma la forma de etiquetas de inicio, de etiquetas de fin, de etiquetas de elemento vacío, de referencias de entidad, de referencias de carácter, comentarios, delimitadores de secciones CDATA, declaraciones de tipo de documento e instrucciones de procesamiento.

Su reconocimiento es muy sencillo. Empieza con los caracteres menor que "<" o el ampersand "&". El resto no son más que datos de carácter

Existe un conjunto de caracteres denominados, caracteres de espacios en blanco, que los procesadores XML tratan de forma diferente en el marcado XML. Se trata de caracteres "invisibles": espacios (UNICODE/ASCII 32), tabuladores (UNICODE/ASCII 9), retornos de carro (UNICODE/ASCII 13) y salto de línea (UNICODE/ASCII 10). Todos ellos corresponden básicamente a la barra espaciadora, tabulador y enter del teclado.

Cuando la especificación XML afirma que los espacios en blanco se permiten en un determinado punto, esto significa que se pueden introducir este tipo de caracteres como sea, en la combinación que se desee. Al igual que pueden saltarse dos líneas entre párrafos en un procesador de texto para que el documento impreso resulte legible, se pueden añadir retornos de carro en algunos puntos de XML para hacer que el archivo fuente sea legible y pueda mantenerse. Cuando después se procesa el documento. Estos caracteres se ignoran.

Cuando se utiliza XML, a menudo es necesario asignar nombres a las cosas. Las estructuras lógicas se tienen que designar con nombres de tipos de elementos, los datos reutilizables con nombres de entidades, los elementos particulares con ids, etc. En XML, los nombres tienen algunas características en común, no son tan flexibles como los datos de carácter:

Un nombre [empieza] con una letra o uno o más signos de puntuación [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", [sin que importe, en este caso la correspondencia de tipo de letra] se reservan para la estandarización de esta o de futuras versiones de esta especificación.

Esto quiere decir que no se pueden crear nombres que empiecen con la cadena "xml" o con alguna variante con otro tipo de letra "XML" o "XmL". Las letras y rayas pueden utilizarse en cualquier parte del nombre. También se pueden incluir dígitos, guiones, y caracteres de punto (.), pero el nombre no puede empezar con ninguno de ellos. El resto de caracteres, como algunos símbolos o espacios en blanco no pueden formar parte de un nombre.

Existe otra construcción sintáctica relacionada que se conoce como *name token* (señales de nombre). Los name token revisten la misma forma de los nombres con la salvedad de que pueden empezar con dígitos, guiones y caracteres de punto, así como con la cadena "xml"

Al igual que el resto en XML, los nombres y name token, son sensibles al tipo de letra en las correspondencias. Los nombres y name token no permiten los espacios en blanco, la mayoría de signos de puntuación u otros caracteres especiales. El resto de caracteres "ordinarios" se denominan caracteres de nombre.

Los datos (el texto que no se ha marcado) pueden contener casi todos los caracteres, lógicamente, en el texto principal del documento, es necesario utilizar signos de puntuación y caracteres de espacios en blanco para que resulte comprensible. Pero a veces estos caracteres también se necesitan dentro del marcado, por ejemplo, un elemento puede representar un hipervínculo y por tanto debe de contener una dirección electrónica, la cual tendrá que ir incluida en el marcado, donde normalmente sólo se permiten los caracteres de nombre.

Las cadenas literales permiten a los usuarios utilizar caracteres que no sean de nombre dentro del marcado, pero solo en contextos en que tenga sentido especificar los valores que puedan requerir estos caracteres. Por ejemplo para especificar el URL en el hipervínculo, es necesario introducir el carácter barra diagonal.

Por ejemplo: <REFERENCIA URL="http://www.document.com/ejemplo.xml">

La cadena que define el URL es la cadena literal. Esta empieza y acaba con comillas dobles. Las cadenas literales siempre están enmarcadas por las comillas dobles o las comillas verticales. Pero las comillas no forman parte de la cadena. La especificación XML dice al respecto lo siguiente:

Los datos literales son cualquier cadena entrecomillada que no contenga la marca de comillas que se utiliza como delimitador de esa cadena. Las cadenas literales se utilizan para especificar el contenido de las entidades internas, los valores de los atributos y los identificadores externos.

Se pueden utilizar tanto las comillas verticales o apóstrofes (' ') como las comillas dobles (" ") para marcar o delimitar el principio y el final de estas cadenas en los documentos XML. Pero el tipo de comillas con el que empiece la cadena tiene que ser necesariamente el mismo con el que acabe. El otro tipo se puede emplear dentro de la cadena sin que tenga un significado especial. Normalmente se utilizarán cadenas dobles cuando se quiera introducir un carácter entrecomillado en la cadena literal y las comillas verticales cuando se desea incorporar una referencia con comillas dobles.

Además del contenido, los documentos pueden tener atributos. Los atributos son una forma de incorporar características o propiedades a los elementos de un documento. Los atributos tienen nombres, al igual que las propiedades que caracterizan al mundo real. También tienen asignados valores. Por ejemplo, dos atributos posibles del elemento "personas" son el "numero de zapato" que calzan y la "edad" (los nombres de atributos) y los dos valores posibles de este atributo son "28" y "28" (respectivamente).

En una DTD, cada atributo viene definido para un tipo de elemento específico y puede adoptar un cierto tipo de valor. Los tipos de elementos múltiples pueden proporcionar atributos con el mismo nombre y a veces resulta conveniente pensar en ellos como si se tratara del "mismo atributo", aunque técnicamente no lo sean.

Los atributos también tienen una semántica, es decir, también significan algo. Por ejemplo, un atributo designado como "peso" puede referirse a los elementos "persona" (coincidencia permitida), tener asignado valores en forma de números (valores permitidos) y representar la altura de una persona expresada en centímetros (semántica).

Por ejemplo: <Persona peso="70.5 kg" edad="70">

El nombre del atributo no va entrecomillado, pero siempre el valor del atributo lo estará. Al igual que otras cadenas literales, los atributos pueden estar marcados entre comillas o entre apóstrofes. Cuando se utiliza un tipo de comillas, el otro puede usarse dentro del valor del atributo.

En ocasiones resulta práctico incorporar comentarios sobre un documento o su marcado de forma que venga ignorado por los procesos de la computadora y las reproducciones del documento. Los comentarios pueden estar dentro de un documento XML o dentro de una DTD. Pueden introducirse en cualquier lugar de la instancia o del prólogo. Sin embargo no pueden insertarse dentro de las declaraciones, etiquetas, ni de otros comentarios.

Los comentarios empiezan con los caracteres "<!--" seguidos del comentario en sí y terminan con los caracteres "-->". Por ejemplo: <!-- ejemplo de comentario puede ir prácticamente cualquier cosa aquí -->

II.4. RELACIÓN DE XML CON SGML Y HTML.

II.4.1. Presentación – Procesamiento

Inicialmente, la expectativa es que XML será utilizado como un formato de almacenamiento y HTML sea usado como formato de presentación. Se puede ejecutar un documento XML por medio de algún proceso para obtener un documento HTML. Esto proporcionaría soporte en compatibilidad hacia nuevos Navegadores y aplicaciones para HTML. Por otro lado, en la actualidad XML no ofrece algo que SGML no pueda realizar, sin embargo, XML ofrece una manera más sencilla y práctica de utilización; además, como se van desarrollando estándares a futuro, XML ofrecerá maneras de hacer enlaces (link's hacia otros elementos) más robustos y permitirá nuevos tipos de datos que podrían sobrepasar a SGML.

Esta proyección se espera por lo menos 2 o 3 años. Aunque, el uso de los navegadores nativos de XML aumentará a través de ese tiempo y surja un eventual eclipse de HTML. Todo dependerá de las herramientas creadas a futuro. Sin embargo XML está soportado por importantes vendedores de navegadores Web mientras que SGML no.

II.4.2. Paso de SGML a XML

Para la conversión de documentos SGML a XML (hablando de no convertir DTDs), la mayoría de las diferencias entre ambos lenguajes son triviales, pequeñas correcciones en sintaxis. En casos raros, no es tan simple, porque la DTD de SGML permite incluir estructura de documentos en ella que los documentos XML no permiten tener. La contrariedad "más grande" de pasar de SGML a XML es la pérdida de "las secciones marcadas", que son una manera nativa de SGML para condicionar. En la epopeya de árbol textual, el proceso real de conversión es absolutamente sencillo; es solo cuestión de almacenar el documento como XML.

Para procesar datos SGML, una aplicación requiere de ambos, una DTD y los datos. En contraste, las aplicaciones que procesan XML no requieren una DTD para procesar los datos.

SGML solamente estandariza la estructura y no prescribe ninguna ayuda para los estilos; XML se acompaña de hojas de estilo para ofrecer presentación para los datos y documentos de XML.

Si se tienen pequeños bloques de un sistema específico de información, tal como símbolos matemáticos u otros símbolos especiales para dicho sistema, SGML permite definirlos con las entidades internas. Aunque éstos fueran diseñados para pertenecer a ese sistema-específico, muchas herramientas de SGML soportan un sistema común. XML no apoya esta capacidad.

Para pasar SGML a XML, si los datos están en el SGML y se desea publicarlo en la Web, se tienen dos opciones:

- Convertirlo a HTML, preservando su formato pero perdiendo su semántica de marcado.
- Convertirlo a XML y guardar su semántica de marcado.

Cada opción consume una pequeña cantidad de costo y de esfuerzo, pero XML permite que se preserve la inteligencia que se ha construido en los documentos. SGML todavía proporciona longevidad y confiabilidad hoy tanto como lo hizo en los días sin XML. Puesto que se puede convertir fácilmente datos del SGML a XML, se puede aprovechar de cualquier uso de XML que emerja.

SGML ofrece cierto poder de validación más estricto, salvo esa excepción, la mayoría de las características que separan a SGML de XML es tan oscuro que fueron utilizadas raramente en primer lugar.

Las circunstancias sobre las cuales se recomienda utilizar SGML actualmente, son casos muy especiales. Puede haber situaciones donde las capacidades de validación de contenido adicionales de SGML y en particular, su soporte para "inclusiones", "exclusiones" y modelos de contenido variantes, serían de gran ayuda.

II.4.3. Integración de XML con HTML

Aprender a crear XML y manipular fuentes de datos de XML permitirá como autor de HTML proveer páginas Web con contenido que es más inteligente y dinámico. El marcar datos usando XML también permitirá crear fuentes de datos que se puedan acceder en un número de formas diversas para un número de propósitos diversos, haciendo posible una inter-operabilidad entre las aplicaciones y la Web.

XML se está convirtiendo en el vehículo para datos estructurados sobre la Web, complementando en gran medida a HTML, que se utiliza para presentar los datos. Al separar los datos estructurados de la presentación, los desarrolladores en la Web pueden construir un nuevo modelo de Aplicaciones Web, donde la información esté más aplicada a la realidad.

Muchas discusiones en soporte de XML han tomado fuerza de una valoración crítica de las limitaciones de HTML, las cuales han acrecentado el interés en una tecnología como XML.

Algunas motivaciones de gran alcance para preservar HTML existen, a pesar de la expansión de XML. Más allá de la motivación obvia, que es el soporte de millones de páginas Web activas, nadie puede realmente esperar que los volúmenes de páginas HTML útiles se conviertan de la noche a la mañana en irrelevantes. De hecho, HTML tiene un papel importante a jugar en el nuevo mundo de XML. Además, se encuentra también la familiarización de los usuarios con el lenguaje universal de la Web: HTML.

XML es un formato basado en texto, parecido al HTML en muchos aspectos, específicamente diseñado para almacenar y transmitir datos. Un origen XML se compone de elementos XML, cada uno de los cuales consta en general de una etiqueta de inicio (<ejemplo>), de una etiqueta de fin (</ejemplo>) y de los datos comprendidos entre ambas etiquetas (el contenido). Al igual que los documentos HTML, un documento XML contiene texto anotado por etiquetas. Sin embargo, a diferencia de HTML, XML admite un conjunto ilimitado de etiquetas, no para indicar el aspecto que debe tener algo, sino lo que significa. Por ejemplo: un elemento XML puede estar etiquetado como precio, número de pedido o nombre. El autor del documento es quien decide qué tipo de datos va a utilizar y qué etiquetas son las más adecuadas.

En lugar de describir el orden y la disposición de la presentación de los datos, las etiquetas indican qué significa cada elemento de datos (si es un elemento <date>, un elemento <area>, etc.). Cualquier receptor de estos datos puede decodificar el documento y utilizarlo para sus propios fines. Por ejemplo: un particular puede utilizarlo para planificar el día, mientras que un meteorólogo puede utilizarlo como datos para un registro histórico.

En XML se puede definir un conjunto ilimitado de etiquetas. Mientras que las etiquetas HTML se pueden utilizar para mostrar una palabra en negrita o en cursiva, XML ofrece un marco de trabajo para etiquetar datos estructurados. Un elemento XML puede indicar que los datos asociados son un precio de venta, un impuesto de ventas, un índice de pluviosidad o cualquier otro elemento de datos.

Dado que las etiquetas XML son iguales en toda una organización y para todos los usuarios de Internet, existirá la capacidad de buscar y manipular datos independientemente de las aplicaciones en las que se encuentren. Una vez encontrados los datos, se pueden entregar a través de la red y presentar en un explorador de muy distintas formas, o bien se pueden pasar a otras aplicaciones para procesarlos y verlos.

Una etiqueta representa una parte de los datos. A menudo corresponde a un campo de una tabla. No obstante, esto no es necesario en absoluto. La etiqueta puede ser una columna calculada (precio por cantidad). Un archivo XML no tiene por qué representar los datos de una tabla. Es igual de frecuente que un archivo XML represente los resultados de una consulta efectuada en varias tablas. Siempre y cuando la aplicación receptora pueda descifrar los datos del archivo XML, la procedencia de los datos y su llegada al archivo XML es indiferente.

La potencia y la belleza de XML residen en el hecho de que mantiene la separación entre la interfase de usuario y los datos estructurados. HTML especifica la forma de presentar los datos en un explorador, mientras que XML define su contenido. En HTML, las etiquetas se utilizan para indicar al explorador que presente los datos en negrita o en cursiva, mientras que en XML las etiquetas sólo se utilizan para describir los datos, como el nombre de la ciudad, la temperatura y la presión atmosférica. En XML se utilizan hojas de estilo, como el lenguaje de estilo extensible (XSL) y las hojas de estilo en cascada (CSS), para presentar los

datos en un explorador. XML separa los datos de la presentación y del proceso, lo que permite mostrar y procesar los datos a su gusto con sólo aplicar distintas hojas de estilo y aplicaciones.

Esta separación entre los datos y la presentación permite una perfecta integración de los datos procedentes de varios orígenes. La información sobre clientes, los pedidos de compra, los resultados de búsquedas, los pagos de facturas, los registros médicos, los datos de catálogo y otros orígenes de datos pueden convertirse en XML en el nivel medio, lo que permite intercambiar los datos en línea con la misma facilidad con la que las páginas HTML muestran los datos hoy en día. A continuación, los datos codificados en XML se pueden distribuir por la Web hasta el escritorio. No hace falta ninguna retroalimentación para la información previamente almacenada en bases de datos o documentos de grandes sistemas. Además, puesto que se utiliza HTTP para distribuir el XML, esta función no requiere ninguna modificación.

Por ejemplo, se puede utilizar XML para describir información sobre páginas HTML o para describir los datos contenidos en reglas u objetos empresariales de una transacción de comercio electrónico, como facturas, pedidos de compra y formularios de pedido.

II.5. XML Y LAS BASES DE DATOS

II.5.1. Manipulación de bases de datos en Web

Las bases de datos se utilizan en el desarrollo de la Web de dos maneras: como interfaces a las fuentes de la información, las cuales se manipulan usando el browser como componente del sistema tradicional cliente-servidor y como una capa subyacente del sistema del cual se genera dinámicamente el contenido que compone una página Web.

Considérese un producto de base de datos, utilizado por el sistema de organización interno de un fabricante de juguetes. Puede que este desee poner la base de datos a disposición de los usuarios de Internet para dar a conocer sus productos y precios. En lugar de tener a alguien en el departamento de diseño Web encargado de volver a marcar los datos, se podría establecer una conexión entre su servidor Web y su base de datos mediante las características propias de los servidores Web que permiten este tipo de conducción de datos.

Permitiría a los diseñadores confeccionar una bonita lista de productos utilizando una hoja de estilos. La base de datos podría insertar imágenes de los juguetes, el sitio Web consistiría esencialmente en una consulta de los datos en la base de datos. Según se fueran retirando o añadiendo juguetes y estos cambios aparecerían automáticamente al visitar la página Web. Es un sistema que da una mayor libertad al administrador de la página Web, ya que le permite actualizar su aspecto sin alterar en modo alguno la base de datos o la instalación que la conecta al servidor Web.

Las expectativas sobre XML también lo convierten en una herramienta indispensable para el intercambio de información de base de datos. Desde siempre, las bases de datos han intercambiado información utilizando formatos simples de archivo de tipo, un solo registro por línea, con puntos y comas entre cada campo, resulta insuficiente para la nueva información orientada a objeto producida por la base de datos. Los objetos deben tener una estructura interna con enlaces entre ellos. XML es capaz de ello, puesto que puede utilizar elementos y atributos que proporcionen un formato común para transferir registros entre base de datos. Una base de datos podría crear un documento XML representando todos los juguetes fabricados que se pudiera cargar directamente en otra base de datos o bien dentro de la compañía o en las páginas de cliente. Es una manera muy innovadora de comprender los documentos, porque en muchos casos las personas nunca tendrán acceso a ellos. Serán documentos creados por y para el software de la computadora.

II.5.2. Integración con Manejadores de B.D.

XML es capaz de interactuar con sistemas que no suelen estar relacionados con la documentación. La mayor estructuración de los documentos, supone su mayor integración con otros datos estructurados de una organización. Pueden seguirse las mismas técnicas para su creación (como en el caso de software de

escritura, de informes o interfaces gráficas de usuario personalizadas) y utilizarse el mismo software para su lectura (como hojas de cálculo y software de base de datos). Una de las aplicaciones más populares de XML será sin duda la edición de base de datos en la Web.

Supongamos que un vendedor decide vender una gama de juguetes a través de una base de datos. Se pondría en contacto con el fabricante para organizar la venta. Una vez llegados a un acuerdo, sobre el formato del mensaje de petición de producto basado en XML, se concretaría el tipo de documento XML, si bien es muy posible que exista de antemano el tipo de documento XML normalizado para esa tarea específica. Una vez elegido, los pedidos de productos podrían enviarse automáticamente desde la computadora de comprador hasta la del vendedor. Este tipo de comercio electrónico lleva años practicándose, si bien XML posibilita que sea fácilmente normalizado, altamente extensible y conectado a las tecnologías soporte Internet. La fácil disponibilidad del software y de los estándares facilitara el uso del comercio electrónico por las organizaciones más pequeñas.

Se hará posible la integración de bases de datos distribuidas en los navegadores que admitan XML, pudiéndose modificar el contenido y la estructura de esta. Actualmente implantado en amplias redes nacionales, sin embargo, se limitan las posibilidades al establecerse una Intranets/Extranet y con amplias bases de datos que sólo permiten la visualización de los datos en el navegador.

Con XML nace una nueva generación de software de servidor Web, que permite al desarrollador Web añadir un nuevo servidor "de nivel intermedio" al modelo Web.

En el modelo tradicional Web, el cliente que utilizaba un navegador (como Internet Explorer o Netscape), interactuaba directamente con las fuentes de datos de los servidores remotos y mantenía la conexión durante toda la sesión interactiva. Cada petición recibía respuesta en HTML y podía ser directamente visualizada por el navegador del cliente. Mantener la conexión entre el cliente y el servidor era imprescindible.

En un nuevo modelo Web de tres niveles, la información que se ajusta al perfil del cliente es extraída de una sola vez de la base de datos remota por el software de nivel intermedio, o bien como documentos XML o mediante una conexión equivalente a una base de datos. A partir de ahí, la interacción con la base de datos remota puede suspenderse. La conexión a los servidores remotos se da por concluida. Una vez que toda la información requerida por el cliente ha sido organizada por el software del nivel intermedio, se envía en XML al cliente. En ese momento, también se descartará la solicitud de una mayor interacción entre el cliente y el servidor del nivel intermedio.

Los datos enriquecidos XML son entregados, para que puedan ser utilizados directamente por lenguajes de guión como JAVASCRIPT y por las aplicaciones de cliente. Se da por finalizada la interacción entre el cliente y el servidor del nivel intermedio. A partir de ahí, toda la operación estará basada en el cliente, lo que implica un uso más eficiente de la Web y una experiencia más satisfactoria para el cliente.

Otro aspecto es que se podría aplicar un manual de diferentes grados (iniciación, intermedio y avanzado) con sus diferentes idiomas, etcétera. Esto hará que este manual se pueda personalizar por los usuarios y extraer la información requerida de un capítulo determinado, con una ordenación y formatos concretos.

Por último, habrá una interacción entre la información requerida y las preferencias del usuario de la aplicación. Con XML vendrá una segunda generación de robots que permitirá una mayor precisión de la búsqueda requerida. Actualmente podemos encontrar aplicaciones de medios de comunicación como los periódicos personalizados. Por otro lado, se impulsaran de los motores de búsqueda con XML y la creación de buscadores de lenguaje natural.

Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido Web en XML consigue que la estructura de la información resulte más accesible.

Si imaginamos que se está buscando información sobre gastos militares en países de América Latina. Si realiza una búsqueda empleando sólo la palabra "armas", los resultados incluirán todas las ciudades con una "Plaza de Armas", que sin duda es el caso de cualquier comunidad con más de mil habitantes.

Pero si los desarrolladores de sitios Web crearan un DTD que incluyera etiquetas para distinguir entre "plaza de armas" y la palabra "armas", tal y como la emplean los ministerios de defensa, entonces los resultados de la búsqueda se restringirían adecuadamente, tardarían menos y ofrecerían un mejor listado de datos y fuentes.

La inexactitud del contenido de la búsqueda es apenas frustrante para un usuario casual, pero es realmente costoso para profesionales. Cuando una computadora da una respuesta errónea al requerimiento de información, la falla puede ser por varias razones, como corto circuitos y errores de programación. Pero el principal punto de error de la mayoría de las búsquedas de contenido es los datos mismos.

Actualmente, las bases de datos usan listas de palabras claves e índices para acelerar la búsqueda. En vez de buscar por los datos que está buscando, se supone que se construyen búsquedas que constan principalmente de palabras claves y datos indexados. Para hacer que una búsqueda trabaje como debería, los sistemas deben cambiar la forma como ellos representan y manejan los datos. Es necesario que el esquema actual de fila/columna/índice supere las limitaciones de las computadoras de hoy. Pero los sistemas futuros mantendrán todo el contenido de la búsqueda en memoria y la buscarán en grandes arreglos de supercomputadoras, de manera que no serán necesarios los índices y las palabras claves.

Lo anterior también vale para las bases de datos B2B (de intercambio entre empresas), en donde el término "precio" puede tener múltiples significados. Pero si se amarra una etiqueta específica de XML a un botón o una interacción particular en la interfaz del usuario, entonces las bases de datos proporcionarán información específica.

Con esto en mente, los arquitectos de soluciones de hoy deben estructurar los datos de una forma que trabaje con la forma de búsquedas de hoy, pero que además tome ventaja de los servidores ultraeficientes del futuro. Las soluciones de hoy darán la vía a bases de datos jerárquicas con esquemas flexibles. Esta transición está ya en vía con la rápida aceptación de XML, porque tiene todas las cualidades necesarias para permitir un sistema de recuperación de contenido. Los datos XML son jerárquicos, un arreglo más realista que una estructura de base de datos relacional de dos dimensiones. Éste crea orden a través de reforzar las reglas estructurales, pero XML solamente permite cambios a la estructura.

II.5.3. XML como una base de datos

Existe un tipo especial de datos que interesa mayoritariamente a los grandes editores Web, se trata de los metadatos: información sobre datos. XML esta en el origen de normas sobre metadatos como el CDF, para la descripción de canales "canales Web en expansión", MCF y el marco de metacontenidos de Netscape, para la búsqueda de información sobre sitios Web y la plataforma para la selección de contenidos de Internet que permite la filtración de material innecesario en pantallas de computadoras, basándose en descripciones externas de contenido. Son aplicaciones denominadas metadatos, porque sirven para describir otros recursos de información. La etiqueta "contiene escenas violentas", en una cinta de video es un ejemplo perfecto de metadatos, es un dato que describe el contenido de la cinta, es decir, es un dato sobre datos.

Aunque todavía no se ha generalizado el uso del XML, sobre todo por la falta de definición de las normas asociadas, ya se pueden gestionar los datos de una base de datos XML con cierta facilidad.

Una base de datos XML no es diferente de cualquier documento XML, ya que XML permite definir cualquier tipo de información. La diferencia entre documentos XML que tratan diversos tipos de contenidos, no es más que la distinta organización de sus elementos, gestionados a través de la colección de etiquetas adecuadas a cada caso. Las bases de datos más sencillas son aquellas que se pueden expresar en forma de tabla, por lo que se suelen conocer como "tabulares". Un ejemplo de bases de datos tabular sería

Base de Datos: Discos

Número	Título	Artista	Precio	Formato
1	En Vivo	Paul McCartney	\$ 150.00	Compact Disc
2	El Tri Sinfónico	El Tri	\$ 75.00	Cassette
3	Lo mejor de hoy	Varios	\$ 150.00	Compact Disc

El documento XML, que establece esta base de datos, es el siguiente. (discos.xml)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DISCOS>
  <DISCO>
    <NUMERO>1<NUMERO>
    <TITULO>En vivo<TITULO>
    <ARTISTA>Paul McCartney<ARTISTA>
    <PRECIO>150.00<PRECIO>
    <FORMATO>Compact Disc<FORMATO>
  </DISCO>
  <DISCO>
    <NUMERO>2<NUMERO>
    <TITULO> El Tri Sinfónico <TITULO>
    <ARTISTA> El Tri <ARTISTA>
    <PRECIO>75.00<PRECIO>
    <FORMATO> Cassette <FORMATO>
  </DISCO>
  <DISCO>
    <NUMERO>3<NUMERO>
    <TITULO> Lo mejor de hoy <TITULO>
    <ARTISTA> Varios <ARTISTA>
    <PRECIO>150.00<PRECIO>
    <FORMATO> Compact Disc <FORMATO>
  </DISCO>
</DISCOS>
```

Para introducir los datos de discos.xml en una determinada página Web con código HTML hay varios sistemas. El más cómodo que se puede utilizar desde el navegador Internet Explorer 5 supone dos pasos: insertar los datos del archivo y mostrar los campos en la forma más adecuada.

En estos casos se suele recurrir a las tablas como sistema para mostrar los datos. El documento HTML que nos ayuda para mostrar los datos, es el siguiente.

```
<html>
<head>
<title>Ejemplo</title>
</head>
<body>
<h1>Discos</h1>
<xml id="ejemplillo" src="discos.xml"></xml>
<table datasrc="#ejemplillo" border="0" cellspacing="4" cellpadding="2">
<caption>LISTA EJEMPLO</caption>
<thead>
<tr bgcolor="#cc99cc">
<th>Núm</th>
<th>Titulo</th>
<th>Artista</th>
<th>Precio</th>
<th>Formato</th>
</tr>
</thead>
<tbody>
<tr bgcolor="#cccccc">
<td align="center"><span datafld="numero"></span></td>
<td align="right"><span datafld="titulo"></span></td>
<td><b><span datafld="artista"></span></b></td>
<td> <span datafld="precio"></span> </td>
<td> <span datafld="formato"></span></td>
</tr>
</tbody>
</table>
</body>
</html>
```

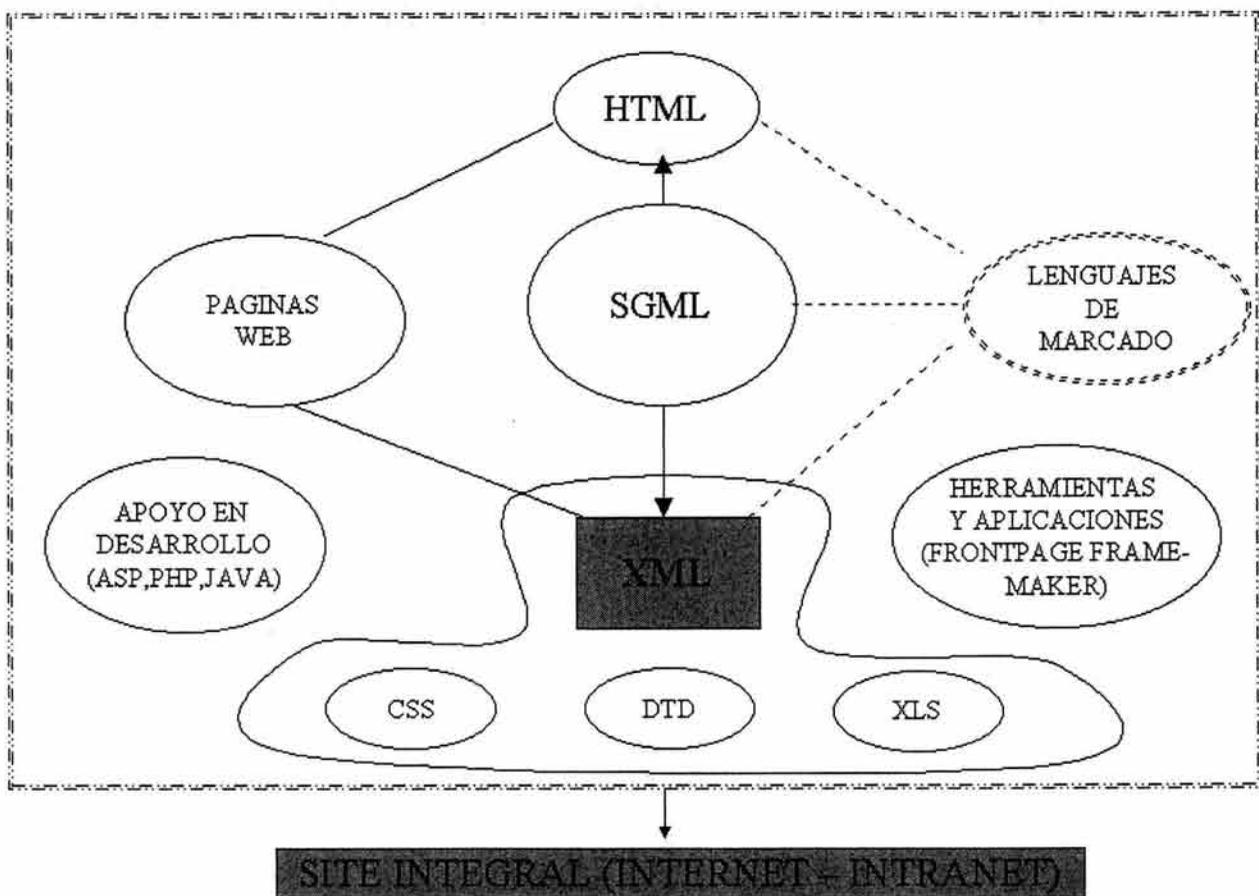
II.6. HERRAMIENTAS DE APOYO.

Se puede suponer que XML constituye la capa más baja dentro del nivel de una aplicación, sobre el que se puede montar cualquier estructura de tratamiento de documentos, hasta llegar a la presentación. El tratamiento que se le da a los documentos XML puede ser para presentar su información de forma aceptable y que el usuario final visualice un resultado a través de "lenguajes de hojas de estilos", También puede tener soporte de una DTD (Definición de Tipo de Documento) para establecer la estructura y los elementos que forman el documento XML.

Pero los lenguajes de hojas de estilo no son la única herramienta de apoyo para utilizar y manejar un documento XML. Mediante Lenguajes de programación como Java, Perl, Visual Basic o cualquier otro lenguaje en el que existan parsers de XML se pueden crear aplicaciones que lean archivos XML y los presenten de alguna forma deseada.

Como ya se ha dicho, en principio un documento XML no son más que datos, y que como tal se pueden utilizar en múltiples aplicaciones informáticas y con objetivos muy diferentes, por lo que es posible que se necesite convertirlo a un formato más comprensible para los usuarios como puede ser HTML, RTF, PDF, etc. Esto es lo que permiten los lenguajes de hojas de estilo.

La integración de XML puede simplificarse bajo el siguiente esquema, donde la combinación de las herramientas y programas para Internet (tales como lenguajes de programación), aplicaciones para desarrollo de páginas, los mismos Lenguajes de Marcado junto con sus tecnologías y complementos, ayudan a componer Sitios Web integrales, donde la manipulación de la información resulta sencilla para las organizaciones y el resultado para los usuarios es igualmente claro.



esquema 2.1. Integración de XML.

Existen tres tipos de lenguajes de hojas de estilos: DSSSL, XSL y CSS, aunque las hojas de estilo más utilizadas para poder visualizar documentos XML son las dos últimas.

II.6.1. DSSSL (Document Style Semantics and Specification Language)

El Lenguaje de Especificación y Semántica de Estilo de Documentos, DSSSL, es un estándar basado en SGML que regula las normas de presentación de documentos de marcas para la Web, del que se puede decir que XSL y CSS son un subconjunto, siempre cumpliendo las normas del XML; o siendo más específicos, podemos decir que se derivan del DSSSL-0 que es un perfil de DSSSL en el que se han eliminado algunas funcionalidades y se han añadido nuevas capacidades para hacerlo más indicado para el trabajo con documentación en línea, mucho más cercano a los documentos de la Web.

Mediante DSSSL se puede: transformar un documento SGML/XML en otro SGML/XML o convertir un documento SGML/XML a un formato de presentación.

Tanto los archivos SGML/XML como las DSSSL son archivos de texto, por lo que para realizar las acciones anteriores es necesario disponer de un motor DSSSL. El más conocido de estos motores es el Jade de James Clark. Por esta razón, DSSSL es más completo pero comparado con XSL o CSS, es menos sencillo en su uso.

XSL (*eXtensible Style Language*)

XSL que en español se traduce como Lenguaje de Estilo Extensible, es un lenguaje de hojas de estilo para documentos XML.

Con XSL se puede transformar un documento en otro XML con un estilo, Convertirlo en otro formato de salida como puede ser RTF, PDF mediante la utilización de los objetos de formateo.

El W3C ha dividido recientemente esta especificación en tres partes¹⁹: XSL, XSLT, y XPath:

- La XSL el lenguaje de objetos de formateo.
- La XSLT la parte de la especificación que describe el lenguaje que nos permite convertir un documento XML en otro XML (en particular en HTML).
- El XPath es un lenguaje mediante el cual podemos hacer referencias a partes de un documento XML

XSL no solo permite especificar cómo queremos presentar los datos de un documento XML, sino que también sirve para filtrar los datos de acuerdo a ciertas condiciones. Se parece un poco más a un lenguaje de programación.

La aplicación de una hoja de estilo XSL a un documento XML puede ocurrir tanto en el origen (por ejemplo, un servlet que convierta de XML a HTML para que sea mostrado a un navegador conectado a un servidor de Web), o en el mismo navegador (como es el caso del Internet Explorer 5).

Básicamente, XSL es un lenguaje que define una transformación entre un documento XML de entrada, y otro documento XML de salida.

Una hoja de estilo XSL es una serie de reglas que determinan cómo va a ocurrir la transformación. Cada regla se compone de un patrón (*pattern*) y una acción o plantilla (*template*).

De este modo, cada regla afecta a uno o varios elementos del documento XML. El efecto de las reglas es recursivo, para que un elemento situado dentro de otro elemento pueda ser también transformado. La hoja

¹⁹ La recomendación oficial de XSL se encuentra en: <http://www.w3.org/TR/xsl/>

de estilo tiene una *regla raíz* que, además de ser procesada, llama a las reglas adecuadas para los elementos hijos.

Algunas de las realizaciones extra de XSL incluyen:

- Generación de etiquetas de texto para datos.
- Composiciones complejas utilizando elementos HTML.
- Aparición de datos en más de una ocasión a lo largo de la presentación.
- Acceso a la información almacenada en los atributos de los elementos.
- Reordenación de los datos.
- Comportamientos dinámicos.

CSS (*Cascading Style Sheets*)

Las Hojas de Estilo en Cascada o CSS, pueden entenderse como una descripción del formato en el que se desea que aparezcan las entidades definidas en un documento. Las CSS se utilizan con gran aceptación en documentos HTML, y se aplican también a documentos XML.

CSS es eficaz para describir formatos y presentaciones, pero no sirve para decidir qué tipos de datos deben ser mostrados y cuáles no deben salir en la pantalla. Esto es, CSS se utiliza con documentos XML en los casos en los que todo su contenido debe considerarse sin mayor problema.

El siguiente cuadro establece claramente cuales son las diferencias entre XSL y CSS, que como se mencionó anteriormente, son las hojas de estilo más utilizadas con XML.

XSL	CSS
Utiliza notación XML	Utiliza su propia notación.
El árbol objeto del formato puede ser totalmente diferente del árbol fuente.	El árbol objeto es casi siempre el árbol fuente.
La herencia de propiedades de formato se da sobre el árbol destino del formato.	La herencia de las propiedades de formato se da sobre el árbol fuente.
Independiza el contenido de un documento de su formato permitiendo transformaciones muy sofisticadas, todo ello sin modificar el documento XML original, que podrá ser utilizado en otros destinos simplemente cambiando sus hojas de estilo.	La estructura de un documento XML es idéntica a la estructura de su presentación, es decir, todos sus componentes de aparecer en la presentación (pueden no aparecer), lo harán en el orden en que se definen dentro del documento XML.
Los cambios en la presentación no afectan a la estructura del documento XML original.	La mayoría de los cambios necesarios en la estructura de la presentación llevarán consigo asociados cambios del contenido del documento XML original.
Esta dirigido a operaciones de formato complejas, que suelen tener fases de transformación del documento XML	Está mas orientado a operaciones de formatos sencillas para el tratamiento de documentos <i>en línea</i> .

Por tanto, un elemento muy importante a tener en cuenta a la hora de elegir el lenguaje a utilizar para visualizar un documento en particular, es saber si la estructura de dicho documento es susceptible de ser mostrada. Además hay que tener muy en cuenta el navegador a utilizar, pues no todos los navegadores actualmente admiten ambos tipos de hojas de estilo.

Las hojas de estilo CSS han sido ampliamente utilizadas conjuntamente con HTML (CSS fue desarrollado con el modelo HTML en mente) y existen grandes cantidades de implementaciones en funcionamiento en la actualidad, al contrario que XSL que aún no ha alcanzado el mismo nivel de madurez, no existiendo todavía ningún navegador que lo implemente al cien por ciento. Actualmente la utilización de CSS esta íntimamente ligada con el desarrollo de XHTML.

Las Hojas de Estilo en Cascada o CSS son un estándar desarrollado por el Consorcio W3C²⁰ para hacer que las páginas Web tuvieran mayores capacidades a la hora de ser visualizadas, sin necesidad de añadir a HTML estándar nuevas etiquetas de formato.

El W3C se refiere a estos documentos como “hojas de estilo en cascada” porque se pueden utilizar múltiples estilos para controlar la apariencia de las páginas y el navegador sigue las reglas en “cascada” para determinar la prioridad y resolver los conflictos.

Las ventajas del uso de hojas de estilo CSS son tres principalmente: posibilidades adicionales para el formato-presentación (utilización de márgenes, sangrías, diferentes tamaños, colores de fondo, etc.), un mayor control sobre el documento (reduce el desorden de etiquetas dentro de un documento y permite el uso de variaciones de diseño a través del uso de clases) y mayor facilidad en la presentación de los documentos (Permite modificar el formato de cada una de las páginas Web que componen un sitio sin necesidad de modificar individualmente cada una de las etiquetas que las constituyen).

Una hoja de estilos CSS consiste en un conjunto de reglas de estilo que se aplican a diferentes elementos componentes de un documento XML o HTML que se quiere visualizar.

La definición de un estilo o regla consiste en dos elementos principales:

- El *selector*. Es el elemento al cual se va a aplicar el estilo
- La *declaración*. Describe las propiedades que conforman el estilo.

El *selector* más simple dentro de un documento XML es cualquiera de los elementos componentes definidos por el usuario (en el caso de estar trabajando con HTML sería cualquiera de sus etiquetas). Este selector estará seguido por una lista de propiedades que se aplicarán a dicho elemento (etiqueta), englobadas dentro de un par de llaves “{” y “}”.

Cada propiedad se identifica por un par del tipo:

Nombre_propiedad: valor

Y las diferentes propiedades a aplicar a un mismo elemento (etiqueta) se separan por puntos y coma “;”. Por tanto, la estructura básica de una regla de estilo CSS sería la siguiente:

```
Etiqueta {
  Nombre_propiedad_1: valor_1;
  Nombre_propiedad_2: valor_2;
  Nombre_propiedad_3: valor_3;
  ...
  Nombre_propiedad_n: valor_n;
}
```

²⁰ La recomendación oficial de XSL se encuentra en: <http://www.w3.org/TR/REC-CSS1>

Cuando más de una etiqueta comparte la misma definición de estilo, la forma de expresarlo es escribir dichas etiquetas separadas por comas en la cabecera de la definición del estilo común. El orden en que se coloquen los elementos no tiene importancia. La estructura a utilizar por tanto, es de la siguiente forma:

Etiqueta1, Etiqueta2, ..., EtiquetaN { ... }

Cuando se requiere aplicar un estilo a un elemento siempre y cuando este elemento ocurra en el interior de otro, se debe situar en el *selector* del estilo el elemento de jerarquía mayor (el predecesor del elemento al que se quiere aplicar el estilo) y posteriormente el elemento destino del estilo. Representado por la siguiente estructura:

Elemento_Superior Elemento_Inferior { ... }

Si además se requiere especificar que el elemento al que se va a aplicar el estilo debe ser un descendiente directo de otro (un elemento hijo) se debe indicar escribiéndolo como la estructura siguiente:

Elemento_Padre > Elemento_hijo { ... }

Los comentarios en las hojas de estilo CSS son muy similares a los que se utilizan en el lenguaje de programación C, con los símbolos juntos de diagonal y asterisco para iniciar el comentario y asterisco con diagonal para indicar el fin del mismo. La siguiente estructura muestra su uso:

```
Etiqueta {
  Nombre_propiedad_1: valor_1; /* comentario en esta línea */
  ...
  Nombre_propiedad_n: valor_n; /* comentario en esta otra línea */
}
```

CSS1 recoge dos de los efectos tipográficos más utilizados, como son el remarcar la primera línea de un párrafo o elemento (first-line) o remarcar la primera letra (first-letter). Para ello hace uso de dos pseudo-elementos denominados first-line y first-letter.

La siguiente tabla muestra algunas de las propiedades que se pueden especificar dentro de los estilos, que como se menciono anteriormente, se componen del nombre de la propiedad por una parte y por otra de los posibles valores que dicha propiedad puede tener²¹.

PROPIEDAD	DESCRIPCIÓN	POSIBLES VALORES
De Posición: <ul style="list-style-type: none"> position top, left, right display 	Define el posicionamiento de los diferentes elementos respecto a los demás. Coordenadas donde situar un elemento (en números enteros). Indica cómo debe mostrarse un elemento dentro de un contenedor	absolute, relative, fixed, static, inherit. none, normal, inline, block, list-item, table, ...
De Tipo De Letra <ul style="list-style-type: none"> font-size font-weight font-style font-family font 	Tamaño de la letra (en entero o porcentaje). Peso de la letra (en entero: Normal = 400, Bold = 700). Estilo del tipo de letra (negrita, itálica, etc). Familia de tipos de letra genérica (serif, roman, etc). Genérico para definir los anteriores en una sola	xx-small, x-small, smal, medium, large, x-large, xx-large, smaller, larger (respecto del anterior definido). normal, bold, bolder, lighter. normal, italic, oblique. serif, sans-serif, cursive, fantasy, monospace. font-family, font-style, font-weight, font-size

²¹ La lista completa de propiedades del estándar de CSS se encuentra en el documento "Cascading Style Sheet, level 1" de la página del W3C en la dirección: <http://www.w3.org/pub/www/tr/wd-css1.html>.

	propiedad.	
De Color y Fondo <ul style="list-style-type: none"> • color • background-color 	Color del elemento. Color de fondo de un elemento.	
De Texto <ul style="list-style-type: none"> • word-spacing • letter-spacing • text-decoration • vertical-align • text-transform • text-align • text-indent 	Espaciado entre palabras de texto. Espaciado entre letras (en entero o porcentaje). Estilo especial para el texto. Alineación vertical. Efectos sobre un grupo de texto (conversión a mayúsculas, etc.). Alineación horizontal de elementos de texto. Sangrado del texto. Valor entero	none, underline, overline, line-through. baseline, sub, super, top, text-top, middle, bottom. capitalize, uppercase, lowercase, none. left, right, center, justify.
De Cuadro <ul style="list-style-type: none"> • margin • margin-left • margin-right • margin-top • margin-bottom • padding • padding-left • padding-right • padding-top • padding-bottom • border-style • border-width • border-color • border • width • height 	Valores para los diferentes márgenes de la página. Valores para márgenes de la izquierda de la página. Valores para márgenes de la derecha de la página. Valores para márgenes superior de la página. Valores para márgenes inferior de la página. Espacio entre el margen y el elemento. Espacio entre el margen izquierdo y el elemento. Espacio entre el margen derecho y el elemento. Espacio entre el margen superior y el elemento. Espacio entre el margen inferior y el elemento. Estilo de borde del elemento. Anchura del borde (entero o porcentaje) Color del borde. Define los valores anteriores de borde. Anchura de la zona del elemento. Altura de la zona del elemento.	none, dotted, dashed, solid, double, groove, ridge. thin, medium, thick. border-width, border-style.

En cualquier momento se puede utilizar alguna de las siguientes unidades de medida, que se especifican con los correspondientes símbolos:

- pt puntos
- pc picas
- mm milímetros
- cm centímetros
- in Pulgadas
- px Píxeles

II.6.2. DTD – Document Type Definition

Las DTD no son obligatorias, es decir, en XML a diferencia del SGML es posible crear documentos XML sin DTD. Esto da lugar a dos tipos de documentos XML: Documentos XML bien formados -aquellos que respetan la sintaxis del XML- y Documentos XML validos -aquellos que además de bien formados se ajustan a la DTD que define su estructura-.

Crear una Definición del Tipo de Documento (DTD) es como crear un lenguaje propio de marcado, para una aplicación específica. Por ejemplo, se podría crear una DTD que defina una tarjeta de visita. A partir de esa DTD, se tendría una serie de elementos XML que permitirían definir tarjetas de visita. La DTD define los

tipos de elementos, atributos y entidades permitidas, y puede expresar algunas limitaciones para combinarlos.

Los documentos XML que se ajustan a su DTD, se denominan "válidos". El concepto de "validez" no tiene nada que ver con el de estar "bien-formado". Un documento "bien-formado" simplemente respeta la estructura y sintaxis definidas por la especificación de XML. Un documento "bien-formado" puede además ser "válido" si cumple las reglas de una DTD determinada. También existen documentos XML sin una DTD asociada, en ese caso no son "válidos", pero tampoco "inválidos", simplemente "bien-formados" o no.

La DTD puede residir en un directorio externo, y quizá compartido por varios de documentos. O bien, puede estar contenida en el propio documento XML, como parte de su declaración de tipo de documento. Veamos un ejemplo:

```
<!DOCTYPE etiqueta[
<!ELEMENT etiqueta (nombre, calle, ciudad, pais, codigo)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT calle (#PCDATA)>
<!ELEMENT ciudad (#PCDATA)>
<!ELEMENT pais (#PCDATA)>
<!ELEMENT codigo (#PCDATA)>
]>
<etiqueta>
<nombre>Fulano Mengano</nombre>
<calle>Avenida central No. 27</calle>
<ciudad>D.F.</ciudad>
<pais>México</pais>
<codigo>09343</codigo>
</etiqueta>
```

La declaración del tipo de documento comienza en la primera línea con "<!'" y termina con "']>". Las declaraciones DTD son las líneas que empiezan con "<!ELEMENT" y se denominan declaraciones de tipo elemento. También se pueden declarar atributos, entidades y anotaciones para una DTD.

En el ejemplo anterior, todas las declaraciones DTD que definen "etiqueta" residen dentro del documento. Sin embargo, la DTD se puede definir parcial o completamente en otro lugar.

Por ejemplo:

```
<?xml version="1.0"?>
<!DOCTYPE coche SYSTEM "http://www.sitio.com/dtd/coche.dtd">
<coche>
<modelo>...</modelo>
...
</coche>
```

Declaraciones Tipo Elemento

Los elementos son la base de las marcas XML, y deben ajustarse a un tipo de documento declarado en una DTD para que el documento XML sea considerado válido.

Las declaraciones de tipo de elemento deben empezar con "<!ELEMENT" seguidas por el identificador genérico del elemento que se declara. A continuación tienen una especificación de contenido.

Por ejemplo:

```
<!ELEMENT receta (titulo, ingredientes, procedimiento)>
```

En este ejemplo, el elemento <receta> puede contener dentro elementos <titulo>, <ingredientes> y <procedimiento>, que, a su vez, estarán definidos también en la DTD y podrán contener más elementos. Siguiendo la definición de elemento anterior.

Este ejemplo de documento XML sería válido:

```
<receta>
<titulo>...</titulo>
<ingredientes>...</ingredientes>
<procedimiento>...</procedimiento>
</receta>
```

Pero no lo siguiente:

```
<receta>
<parrafo>Esto es un párrafo</parrafo>
<titulo>...</titulo>
<ingredientes>...</ingredientes>
<procedimiento>...</procedimiento>
</receta>
```

La especificación de contenido puede ser de cuatro tipos:

EMPTY

Puede no tener contenido. Suele usarse para los atributos.

```
<!ELEMENT salto-de-pagina EMPTY>
```

ANY

Puede tener cualquier contenido. No se suele utilizar, ya que es conveniente estructurar adecuadamente los documentos XML.

```
<!ELEMENT descripcion ANY>
```

MIXED

Puede tener caracteres de tipo datos o una mezcla de caracteres y sub-elementos especificados en la especificación de contenido mixto.

```
<!ELEMENT enfasis (#PCDATA)>
<!ELEMENT parrafo (#PCDATA|enfasis)*>
```

Por ejemplo, el primer elemento definido en el ejemplo (<enfasis>) puede contener datos de carácter (#PCDATA). Y el segundo (<parrafo>) puede contener tanto datos de carácter (#PCDATA) como sub-elementos de tipo <enfasis>.

ELEMENT

Sólo puede contener sub-elementos especificados en la especificación de contenido.

```
<!ELEMENT mensaje (remite, destinatario, texto)>
```

Para declarar que un tipo de elemento tenga contenido de elementos se especifica un modelo de contenido en lugar de una especificación de contenido mixto o una de las claves ya descritas.

Modelos De Contenido

Un modelo de contenido es un patrón que establece los sub-elementos aceptados, y el orden en que se acepta. Un modelo sencillo puede tener un solo tipo de sub-elemento:

```
<!ELEMENT aviso (parrrafo)>
```

Esto indica que <aviso> sólo puede contener un solo <parrrafo>.

```
<!ELEMENT aviso (titulo, parrrafo)>
```

La coma, en este caso, denota una secuencia. Es decir, el elemento <aviso> debe contener un <titulo> seguido de un <parrrafo>.

```
<!ELEMENT aviso (parrrafo | grafico)>
```

La barra vertical "|" indica una opción. Es decir, <aviso> puede contener o bien un <parrrafo> o bien un <grafico>. El número de opciones no está limitado a dos, y se pueden agrupar usando paréntesis.

```
<!ELEMENT aviso (titulo, (parrrafo | grafico))>
```

En este último caso, el <aviso> debe contener un <titulo> seguido de un <parrrafo> o de un <grafico>. Además, cada partícula de contenido puede llevar un indicador de frecuencia, que siguen directamente a un identificador general, una secuencia o una opción, y no pueden ir precedidos por espacios en blanco.

Indicadores de frecuencia	
?	Opcional (0 o 1 vez)
*	Opcional y repetible (0 o más veces)
+	Necesario y repetible (1 o más veces)

Para entender esto, se muestra el siguiente ejemplo.

```
<!ELEMENT aviso (titulo?, (parrrafo+, grafico)*)>
```

En este caso, <aviso> puede tener <titulo>, o no (pero sólo uno), y puede tener cero o más conjuntos <parrrafo><grafico>, <parrrafo><parrrafo><grafico>, etc.

Los atributos permiten añadir información adicional a los elementos de un documento. La principal diferencia entre los elementos y los atributos, es que los atributos no pueden contener sub-atributos. Se usan para añadir información corta, sencilla y desestructurada.

```
<mensaje prioridad="urgente">
<de>Juan Perez</de>
<a>Jon Bozak</a>
<texto idioma="inglés">
Hello Jon, How are you?
...
</texto>
</mensaje>
```

Otra diferencia entre los atributos y los elementos, es que cada uno de los atributos sólo se puede especificar una vez, y en cualquier orden.

En el ejemplo anterior, para declarar la lista de atributos de los elementos <mensaje> y <texto> se haría lo siguiente:

```
<!ELEMENT mensaje (de, a, texto)>
<!ATTLIST mensaje prioridad (normal | urgente) normal>
<!ELEMENT texto(#PCDATA)>
<!ATTLIST texto idioma CDATA #REQUIRED>
```

Las declaraciones de los atributos empiezan con "<!ATTLIST", y a continuación del espacio en blanco viene el identificador del elemento al que se aplica el atributo. Después viene el nombre del atributo, su tipo y su valor por defecto. En el ejemplo anterior, el atributo "prioridad" puede estar en el elemento <mensaje> y puede tener el valor "normal" o "urgente", siendo "normal" el valor por defecto si no se especifica el atributo.

El atributo "idioma", pertenece al elemento texto, y puede contener datos de carácter (CDATA). Es más, la palabra #REQUIRED significa que no tiene valor por defecto, ya que es obligatorio especificar este atributo. A menudo interesa que se pueda omitir un atributo, sin que se adopte automáticamente un valor por defecto.

Para esto se usa la condición "#IMPLIED". Por ejemplo, en una supuesta DTD que defina la etiqueta de HTML:

```
<!ATTLIST IMG URL CDATA #REQUIRED
ALT CDATA #IMPLIED>
```

Es decir, el atributo "URL" es obligatorio, mientras que el "ALT" es opcional (y si se omite, no toma ningún valor por defecto).

Los atributos CDATA (character data) son los más sencillos y pueden contener casi cualquier cosa. Los atributos NMTOKEN (name token) son parecidos, pero sólo aceptan los caracteres válidos para nombrar cosas (letras, números, puntos, guiones, subrayados y los dos puntos).

```
<!ATTLIST mensaje fecha CDATA #REQUIRED>
<mensaje fecha="15 de Julio de 1999">
<!ATTLIST mensaje fecha NMTOKEN #REQUIRED>
<mensaje fecha="15-7-1999">
```

Los atributos enumerados son aquellos que sólo pueden contener un valor de entre un número reducido de opciones.

```
<!ATTLIST mensaje prioridad (normal | urgente) normal>
```

Existe otro tipo de atributo parecido, llamado de notación (NOTATION). Este tipo de atributo permite al autor declarar que su valor se ajusta a una notación declarada.

```
<!ATTLIST mensaje fecha NOTATION (ISO-DATE | EUROPEAN-DATE) #REQUIRED>
```

Para declarar las notaciones, se utiliza "<!NOTATION", con una definición externa de la notación. La definición externa puede ser pública o un identificador del sistema para la documentación de la notación, una especificación formal o un asistente de la aplicación que contenga objetos representados en la notación.

```
<!NOTATION HTML SYSTEM "http://www.w3.org/Markup">
<!NOTATION HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

El tipo ID permite que un atributo determinado tenga un nombre único que podrá ser referenciado por un atributo de otro elemento que sea de tipo IDREF. Por ejemplo, para implementar un sencillo sistema de hipervínculos en un documento:

```
<!ELEMENT enlace EMPTY>
<!ATTLIST enlace destino IDREF #REQUIRED>
<!ELEMENT capitulo (parrafo)*>
<!ATTLIST capitulo referencia ID #IMPLIED>
```

En este caso, una etiqueta <enlace destino="seccion-3"> haría referencia a un <capitulo referencia="seccion-3">, de forma que el procesador XML lo podría convertir en un hipervínculo, u otra cosa.

XML hace referencia a objetos (ficheros, páginas Web, imágenes, cualquier cosa) que no deben ser analizados sintácticamente según las reglas de XML, mediante el uso de entidades. Se declaran en la DTD mediante el uso de "<!ENTITY"

Una entidad puede no ser más que una abreviatura que se utiliza como una forma corta de algunos textos. Al usar una referencia a esta entidad, el analizador sintáctico reemplaza la referencia con su contenido. En otras ocasiones es una referencia a un objeto externo o local.

Las entidades pueden ser:

- Internas o Externas
- Analizadas o No analizadas
- Generales o Parámetro
- Entidades generales internas

Son las más sencillas. Son básicamente abreviaturas definidas en la sección de la DTD del documento XML. Son siempre entidades analizadas, es decir, una vez reemplazada la referencia a la entidad por su contenido, pasa a ser parte del documento XML y como tal, es analizada por el procesador XML.

```
<!DOCTYPE texto[
<!ENTITY alf "Alien Life Form">
]>
<texto><titulo>Un día en la vida de un &alf;</titulo></texto>
```

Las entidades externas obtienen su contenido en cualquier otro sitio del sistema, ya sea otro archivo del disco duro, una página Web o un objeto de una base de datos. Se hace referencia al contenido de una entidad así mediante la palabra SYSTEM seguida de un URI (Universal Resource Identifier)

```
<!ENTITY intro SYSTEM "http://www.miservidor.com/intro.xml">
```

Evidentemente, si el contenido de la entidad es un archivo MPG o una imagen GIF o un fichero ejecutable EXE, el procesador XML no debería intentar interpretarlo como si fuera texto XML. Este tipo de entidades siempre son generales y externas.

```
<!ENTITY logo SYSTEM "http://www.miservidor.com/logo.gif">
```

Se denominan entidades parámetro a aquellas que sólo pueden usarse en la DTD y no en el documento XML. Se pueden utilizar para agrupar ciertos elementos de la DTD que se repitan mucho. Se diferencian las entidades parámetro de las generales, en que para hacer referencia a ellas, se usa el símbolo "%" en lugar de "&" tanto como para declararlas como para usarlas.

```
<!DOCTYPE texto[
<!ENTITY % elemento-alf "<!ELEMENT ALF (#PCDATA)>">
...
%elemento-alf;
]>
```

También puede ser externa:

```
<!DOCTYPE texto[
<!ENTITY % elemento-alf SYSTEM "alf.ent">
...
%elemento-alf;
]>
```

II.7. TECNOLOGÍAS DERIVADAS Y COMPLEMENTARIAS A XML.

II.7.1. XML Document Object Model y Java

El empleo de tecnología XML, al ser un estándar internacional y público, no ata a una plataforma o sistema de desarrollo concreto. Lo mismo se puede usar Perl bajo UNIX para generar documentos XML a partir de una base de datos, como usar Python en Windows NT para servir documentos HTML4 a navegadores Web a partir de un documento XML.

Dicho esto, habrá que decir que JAVA se posiciona como una opción interesante al momento de desarrollar aplicaciones usando XML. Por ejemplo, a partir de fuentes de datos en XML, se puede escribir un servlet que analice sintácticamente el XML, y que genere un árbol DOM (Document Object Model). Una vez generado el árbol DOM, se puede ir extrayendo la información que contiene e ir generando un documento HTML de acuerdo con ciertas reglas de formato, de modo que pueda ser visualizado por un navegador Web.

II.7.2. Parser/Dom

Siguiendo con el proceso que se desarrolla en el interior del navegador, después de recoger la información de todos los documentos que definen la información XML, se genera internamente una estructura que organiza a los elementos que describen las etiquetas en forma de árbol jerárquico, lo que facilita el control de dichos elementos.

En caso que se detecte algún error incompatible con las estrictas normas XML, el navegador interrumpe el proceso y muestra dicho error en la pantalla.

Todo este proceso se puede realizar gracias al analizador (parser) interno que incluye cada navegador, que en la mayoría de los casos se relaciona directamente con el estándar DOM (Document Object Model, modelo de objeto de documento), que entre otras cosas permite acceder a cada nodo del árbol a través de scripts.

II.7.3. XML Schemas

Un "schema XML" es algo similar a un DTD, es decir, que define qué elementos puede contener un documento XML, cómo están organizados y qué atributos y de qué tipo pueden tener sus elementos.

La ventaja de los schemas con respecto a los DTDs son:

- Usan sintaxis de XML, al contrario que los DTDs.
- Permiten especificar los tipos de datos.
- Son extensibles.

Por ejemplo, un schema nos permite definir el tipo del contenido de un elemento o de un atributo, y especificar si debe ser un número entero, o una cadena de texto, o una fecha, etc. Los DTDs no nos permiten hacer estas cosas.

El siguiente cuadro muestra un ejemplo de un documento XML, y su schema correspondiente:

```
<documento xmlns="x-schema:personaSchema.xml">
<persona id="fulano">
<nombre>Fulano Mengano</nombre>
</persona>
</documento>
```

Como se puede ver en el documento XML anterior, se hace referencia a un espacio de nombres (namespace) llamado "x-schema:personaSchema.xml". Es decir, se le esta diciendo al analizador sintáctico XML (parser) que valide el documento contra el schema "personaSchema.xml".

El schema sería algo parecido a esto:

```
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
<AttributeType name='id' dt:type='string' required='yes'/>
<ElementType name='nombre' content='textOnly'/>
<ElementType name='persona' content='mixed'>
<attribute type='id'/>
<element type='nombre'/>
</ElementType>
<ElementType name='documento' content='eltOnly'>
<element type='persona'/>
</ElementType>
</Schema>
```

El primer elemento del schema define dos espacios de nombre. El primero "xml-data" le dice al analizador que esto es un schema y no otro documento XML cualquiera. El segundo "datatypes", permite definir el tipo de elementos y atributos utilizando el prefijo "dt".

ElementType

Define el tipo y contenido de un elemento, incluyendo los sub-elementos que pueda contener.

AttributeType

Asigna un tipo y condiciones a un atributo.

Attribute

Declara que un atributo previamente definido por AttributeType puede aparecer como atributo de un elemento determinado.

Element

Declara que un elemento previamente definido por ElementType puede aparecer como contenido de otro elemento.

Tal como se ha visto, es necesario empezar el schema definiendo los elementos más profundamente anidados dentro de la estructura jerárquica de elementos del documento XML. Es decir, se tiene que trabajar "de dentro hacia fuera".

Visto de otra manera, las declaraciones de tipo ElementType y AttributeType deben preceder a las declaraciones de contenido element y attribute correspondientes

II.7.4. Xlink/Xpointer (Lenguaje de enlace)

La cuestión de los enlaces e hipervínculos es tan importante para los documentos XML que el W3C ha publicado las especificaciones que las controlan fuera de las descripciones del DTD, no conformándose con crear una sola norma, ya que existen dos: XLink y XPointer.

XLink (anteriormente conocido como XLL, Extensible Linking Language) define la forma en la que los documentos XML deben conectarse entre sí. XPointer describe cómo se puede apuntar a un lugar específico de un determinado documento XML. Resumiendo, XLink determina el documento al que se desea acceder y XPointer marca el lugar exacto de dicho documento.

A los diseñadores de HTML les puede parecer que esto es una complicación sin sentido, ya que están acostumbrados a las pocas posibilidades que brinda la etiqueta A, pero si pensamos en las muchas variaciones que pueden tener los vínculos, se comprende la solución adoptada.

Las especificaciones de los hipervínculos para XML permiten cosas como: adherirse a cualquier etiqueta, hacer referencia a un lugar concreto de un documento determinado a través de su nombre o localización, ser descritos en documentos externos, ser procesados de formas distintas (automáticamente, activándolo, etc.), ser multifuncionales (permitir varios saltos), etc.

Al contrario de lo que ocurre con HTML, en XML existen dos tipos básicos de hipervínculos: simples y extendidos. Aunque no se explican todas las posibilidades de los hipervínculos XML, sí debe quedar claro que los enlaces XML son más variados que los que proporciona la sencilla, útil y conocida etiqueta A del HTML.

XLink es una aplicación XML que intenta superar las limitaciones que tienen los enlaces de hipertexto en HTML. Es una especificación que todavía está en desarrollo, y que todavía no tiene "rodaje" en el mundo real.

Los enlaces en HTML tienen una serie de limitaciones, que los hacen bastante pobres. Por ejemplo, sólo tienen dos extremos, la terminación origen y la destino, y son unidireccionales.

Si se crea un enlace en página Web personal que lleve hasta la página principal de la UNAM, se está creando un "vínculo" entre la página personal y la de la UNAM-.

```
<a href="http://www.unam.edu.mx/">U.N.A.M.</a>
```

Este vínculo es unidireccional, porque un visitante cualquiera que entre en la página de UNAM, no tiene forma de saber qué página (así como otros cientos de miles) enlaza con ella. El objetivo de XLink es crear enlaces entre recursos, de una forma en la que HTML no es capaz.

Por ejemplo, un estudiante podría hacer anotaciones a los apuntes que un profesor tiene disponibles en la red, por el método de insertar un enlace desde los apuntes (a los que no tiene acceso de escritura o modificación) y su propia página de anotaciones y comentarios.

O bien, se pueden vincular dos páginas cualesquiera, por ejemplo enlazando el texto de una noticia en un diario on-line, con el texto de la noticia equivalente en el diario competidor. Y no hay porqué quedarse en enlaces con dos extremos. Los enlaces extendidos permiten mucho más que eso.

II.8. APLICACIONES DE XML

II.8.1. WEB

La respuesta a sobre si ya se puede utilizar XML en la Web es un rotundo sí, aunque dado que es una normativa nueva, no todas las herramientas relacionadas con Internet son capaces de trabajar con XML, pero muchas tecnologías han ido actualizándose o por lo menos considerándolo y adoptando sus recomendaciones.

En realidad, con un editor de textos ASCII (el Bloc de notas, por ejemplo) y con un visualizador que incluya un parser adaptado a XML (Microsoft Internet Explorer 5, por ejemplo), ya se pueden crear y ver documentos XML.

No obstante, existen muchas nuevas herramientas que facilitan un poco la labor del diseñador, algunas de ellas gratuitas, como el "Microsoft XML Notepad", que aprovecha los motores internos del Explorer para trabajar.

Existen otros varios navegadores que admiten XML, como Amaya, HotMetal o HyBrick, aunque casi cada día están saliendo nuevos productos (analizadores, visualizadores, motores, editores, DTDs, etc.). Aunque la versión 1.0 de XML es ya definitiva, no pasa lo mismo con las demás normativas que le acompañan, que poco a poco van pasando del estado de "borrador de trabajo" al de "recomendación"

Este es otro de los motivos por los que XML no termina de generalizarse, ya que las empresas se piensan mucho invertir en productos que no están soportados por estándares definitivos.

Las necesidades de intercambio de información se habían venido cubriendo con estrategias anticuadas como los archivos delimitados por una coma, representaciones registradas y no normalizadas. El incremento cada vez mayor de intercambio de información en tiempo real, aumenta la necesidad de un formato estándar y que cumpla con el principio de independencia de los datos.

Hace décadas que las bases de datos, sobre todo las empresariales, han logrado la independencia de los datos. Pero los documentos, a diferencia de los datos empresariales, se han creado históricamente en el contexto de un estilo de presentación de salida específico, lo que explica que la independencia de los datos de documentos solo se ha alcanzado apenas en tiempos muy recientes.

Los primeros integradores de sitios, adquirirían la información de sitios asociados en HTML y escribirían con dificultad algunas rutinas de extracción para dar formato y desplegar la información de sus sitios Web. Esa rutina es tediosa y sujeta a errores debido a que los pequeños cambios en el formato interrumpían las rutinas de extracción de datos que dependían de la información que se encontrara en cierto lugar de la página Web. Es decir, es como un documento con estilo de presentación específica (no dinámico). En XML se describen los datos en lugar de su presentación, lo cual genera una importante contribución.

Aunque el contenido estático era importante en los primeros sitios Web, los sitios de comercio electrónico actuales y en general aquellos que ofrecen grandes volúmenes de contenido, manejan información dinámica en tiempo real. Por ejemplo, para comprar un boleto de avión, se requiere información en tiempo real que muestre todos los datos pertinentes. Estos sitios dependen de métodos de extracción de datos dinámicos para recuperar ese tipo de información, porque cambia en forma constante.

La información que manejan las empresas es uno de sus principales activos. Pero lo normal es que esa información esté fragmentada: el departamento de compras tiene sus propios contactos, trata con una serie de personas que tienen números de teléfono, dirección, cifras de facturación, etc., donde el departamento de investigación y desarrollo trata con algunas de esas personas, pero no dispone de algunos de esos datos, aunque sí saben qué materias primas utilizan, cuáles son sus procedimientos de trabajo, etc.

La información dentro de la empresa está en muchos tipos distintos de soporte: uno de ellos es el formato digital que puede ser entendido y procesado por las computadoras. Esa información, a su vez, está en las computadoras personales de los empleados, unas veces interconectadas entre sí y otras veces no. Existen además documentos corporativos, mensajes de los clientes, bases de datos, páginas Web, etc.

El reto ahora está en interrelacionar toda esa información para hacer rendir todo su potencial y ponerlo a trabajar para aumentar los beneficios y reducir los costos. Por causa de la inexistencia de un formato de intercambio universal de la información, las empresas posponen la automatización de muchas tareas que podrían prescindir de la intervención humana. Como resultado de ello, los departamentos de sistemas ocupan más tiempo extrayendo, re-escribiendo y actualizando datos para servir a necesidades específicas. XML se perfila como una buena solución al dilema del intercambio universal de información.

A pesar de que la Web naciera en un laboratorio de física, para la comunicación entre físicos, nunca estuvo orientada hacia el desarrollo de un avanzado sistema para la comunicación de fórmulas matemáticas. El mercado para las matemáticas es más complicado de lo que pueda parecer en un primer momento para los no matemáticos. De hecho, no existe aun un consenso entre los matemáticos como sobre hacerlo.

Basta con decir que los intentos de hacer matemáticas en la Web, o son insuficientes o son excesivos. El W3C está actualmente trabajando sobre un nuevo lenguaje basado en XML, llamado MathLab. MathLab, consigue situarse en un punto intermedio.

Al ser XML un metalenguaje, sobre él se pueden construir sublenguajes con especificaciones e instrucciones propias. Es el caso de Commerce XML (cXML), un estándar público y libre para el comercio electrónico. cXML ayudará a construir comunidades comerciales abiertas. Compradores y vendedores reducirán el costo de sus operaciones en línea al facilitarse el intercambio de información sobre Internet.

II.8.2. EDI – Electronic Data Interchange

El intercambio de información que además se realiza entre sistemas heterogéneos, es un punto importante a cubrir y de gran importancia para el mundo de los negocios electrónicos. Hace décadas que las corporaciones llevan invirtiendo billones de dólares en la automatización de los procesos internos. Si bien estas inversiones han supuesto una mejora de la eficacia, no ha influido en absoluto en la evolución de los procesos externos.

Efectivamente, las compañías han creado focos de automatización aislados de sus socios comerciales, los vendedores y clientes. La interacción entre las compañías y sus socios comerciales es lenta e inclusive ineficaz porque todavía se basa en procesos manuales.

El Intercambio Electrónico de Datos abreviado como EDI (Electronic Data Interchange), se presenta como una solución al problema. Se define como el intercambio de datos entre sistemas heterogéneos para el soporte de transacciones.

El EDI no se limita a exportar datos de un sistema a otro, implica una interacción real entre los sistemas. Pongamos el ejemplo de dos empresas, La empresa A proveedora de una empresa B. En lugar de intercambiarse pedidos de compra, facturas, cheques en formatos o formularios impresos, ambas empresas podrán efectuar estas operaciones electrónicamente a través de la interconexión de sus sistemas.

Las ventajas son muchas: mayor rapidez en algún cambio de pedido, mejor control de inventario, información completa en tiempo real sobre pedidos e inventario para una mejora en la toma de decisiones, reducción de costos en datos de entrada manuales, etc. De hecho, las ventajas generan una cadena de beneficios de varios proveedores. Por ejemplo, supongamos a otra empresa C, como empresa proveedora de la empresa A ya mencionada. Si ambas compañías C y A implementa el EDI, la compañía B se beneficiara de las ventajas de una mayor integración con el conjunto de la cadena de proveedores.

XML es una clave importante para el EDI en Internet. XML es una notación universal (formato de datos) que permite a las computadoras almacenar y transferir datos inteligibles por cualquier sistema de computadora. XML deja intactos el contenido y la estructura, además de que separa las normas empresariales de los datos. El resultado es que cada socio comercial es libre de aplicar sus propias normas. Esta flexibilidad es fundamental para aportar una solución global al EDI.

La pieza final de la solución EDI es el almacén de datos. XML propone un solo conjunto de requisitos para la asignación jerárquica de nombres y la estructura. Exige relaciones ricas y enlaces complejos.

Combinando XML, Internet, los servicios basados en Internet y la conectividad de bases de datos, se logra tener una solución global para en *Nuevo* EDI. Juntas, estas tecnologías hacen posible el cambio y evolución no solo del EDI, sino también del mundo empresarial.

EDI se remonta a los años 70, cuando fue presentado por el TDCC, Comité Coordinador de los Datos de Transporte (Transportation Data Coordinating Committee). EL TDCC creó grupos de transacciones que los vendedores debían seguir para posibilitar el procesamiento de pedidos de compra, y facturas.

En esa época, el panorama de la tecnología era muy diferente del que se conoce en la actualidad. Y como no existían CPU potentes, transporte común y un formato de archivo que permitiera la flexibilidad, definieron un grupo de transacciones rígidas. Esas transacciones direccionaban las necesidades de contenido de datos, estructura y el proceso de gestión de datos. En otras palabras, las normas empresariales se incrustaban en el grupo de transacción.

Habitualmente EDI se crea a partir de grupos de transacciones de carácter fijo. Por ejemplo un campo de contacto incluirá el nombre de la persona, el cargo, nombre de la compañía, dirección y número de teléfono. Sin embargo, la compañía no es lo bastante flexible como para añadir o borrar campos.

Las compañías no pueden darse el lujo de estancarse en el tiempo a causa de un grupo de transacciones fijo porque impide su evolución mediante la agregación de nuevos productos o servicios, el cambio a otro sistema de computadora y la mejora de procesos empresariales. Esta flexibilidad, inherente a las actuales soluciones personalizadas, necesaria para la correspondencia de datos entre dos socios comerciales es insostenible, a pesar de las ventajas visibles del EDI.

La incorporación de normas empresariales en la definición del grupo de transacción acarrea muchos problemas, debido a varias razones, como el que las normas varían de una compañía a otra, o porque están sujetas a cambios en el tiempo según la dinámica de los mercados.

El proceso actual para definir normas para grupos de transacciones podría alargarse varios años. Por tanto, no es válido en el mundo empresarial de este tiempo, que se caracteriza por cambios súbitos y una creciente competencia. Sin embargo, en su intento de impulsar como sea la creación de ontologías de industria bajo la forma DTD para XML, la labor de la normalización tradicional EDI podría acabar dando sus frutos.

Históricamente, las normas sobre la tecnología definidas y gestionadas siguiendo un método de arriba hacia abajo, como las normas EDI, están siendo sustituidas por normas de abajo hacia arriba que contribuyen a un desarrollo más independiente y distribuido. En otras palabras, las tecnologías como XML, que soportan una mayor flexibilidad y diversidad y que además, proporcionan compatibilidad entre implementaciones, están desbancando las soluciones gestionadas inflexibles como los grupos fijos de transacciones. El proceso de normalización XML está a cargo del W3C.

Afortunadamente, están surgiendo nuevas tecnologías decididas a reconfigurar el panorama EDI. Actualmente, EDI se suele implementar basándose en la relación entre dos socios comerciales (Business to Business "B2B").

Esta asociación de 2 puede ampliarse a terceros creándose así una cadena de proveedores. El nuevo paradigma es la Web de suministro. Se planea que esta se base en la utilización de XML, Internet, los servicios basados en Internet y la conectividad de base de datos para crear una red o Web de socios comerciales.

Los costos de implementación y operatividad caerán en picado, los socios comerciales implementarán soluciones válidas para todos los tamaños, su adopción será generalizada.

Las ventajas ya no serán monopolio de los socios comerciales, también afectarán a los usuarios finales. EDI se convertirá en un servicio tan común como el Correo Electrónico.

En resumen EDI revolucionará el panorama de la computación empresarial, cambiando su rumbo, que pasará de los islotes de automatización a una estructura compacta de comercio, que creará un nexo de unión entre las empresas y los usuarios finales. EDI tradicional está basado en tecnologías que ya existían en los años 70. Pero es posible construir una nueva y mejorada arquitectura EDI, que sea cimentada en las tecnologías actuales, como XML, Internet, los servicios basados en Internet y la conectividad de base de datos.

Internet proporciona la conectividad ubicua necesaria para que una Web de socios comerciales interconectados pueda crecer.

La teoría más atrevida se basa en la noción de que la Extranet redefiniría el nuevo paradigma de la computación. Lo que estos entendidos no consideraron, es que Internet por sí solo no da respuestas a las necesidades de la comunidad EDI.

La comunidad EDI se compone casi exclusivamente de grandes empresas, EDI es fundamental y necesita depender de una red. Exige además un grado de seguridad que no se encuentra en Internet. En resumidas cuentas el ahorro no es suficiente como para dar lugar a esa posibilidad. Por otro lado, la conectividad es

sólo una pequeña parte del problema, la cuestión principal es el intercambio de datos en un modelo universal.

Pero se ha logrado cambios importantes. La tecnología es ahora capaz de proporcionar servicios de marcado que soporten Internet, para direccionar a tiempo y con rendimiento la información clave. La capacidad para intercambiar datos de forma mucho más democrática y adecuada disparará el promedio de conexiones a EDI. Además, la utilización de XML ampliará la clientela de EDI, incluyendo a las personas, y a las pequeñas y medianas empresas. Este grupo de clientes es mucho más sensible en cuanto al precio, por lo que tiene tendencia a buscar una solución basada en Internet.

Las tecnologías Internet aportan seguridad, autenticación, soporte en las transacciones y sobre todo, soporte a las necesidades de EDI.

Estas tecnologías han mejorado y siguen en una ascendente evolución. Ya que ofrecen una gran variedad de aplicaciones, herramientas y servicios para la seguridad y contabilidad. La criptografía de clave publica certificación y firmas digitales garantizan la seguridad de transacciones. Por otro lado, también se ha incrementado la fiabilidad de las conexiones de Internet.

La conectividad de base de datos supone que los datos XML y las normas empresariales que interactúan con los datos, puedan comunicarse entre sistemas distintos mediante filtros de datos del nivel intermedio y agregadores.

La fuente de la inmensa mayoría de la información relacionada con EDI se encuentra en las Mainframes y las bases de datos relacionales. Estos datos se señalarán de carrera con etiquetas XML.

Tras décadas de inversiones en centros de datos corporativos, se han formado islotes de automatización dentro de compañías. Su aislamiento de los socios comerciales hace que las compañías infravaloren estos sistemas. EDI puede cambiar esto, ya que ofrece ventajas como:

- Eficacia mejorada
- Gestión de la cadena de suministro
- Datos en tiempo real y sistema métrico
- Mejor planificación
- Ejecución superior
- Sistemas de control
- Gestión de recursos
- Reducción de costos
- Mejor acceso a la información para la toma de decisiones
- Capacidad de respuesta del cliente.

Sin embargo EDI tradicional es muy difícil y caro de implementar. Debido a los problemas relativos a grupos de transacciones rígidos que incrustan las normas empresariales, el desarrollo lento de los estándares, los elevados costos fijos y su limitada implantación en el mercado. EDI aun no ha sido adoptado de forma generalizada. Afortunadamente se hace más accesible aprovechar EDI gracias a nuevas tecnologías para buscar soluciones y en consecuencia redefinir toda la industria EDI.

III.1. DELEGACIÓN IZTAPALAPA

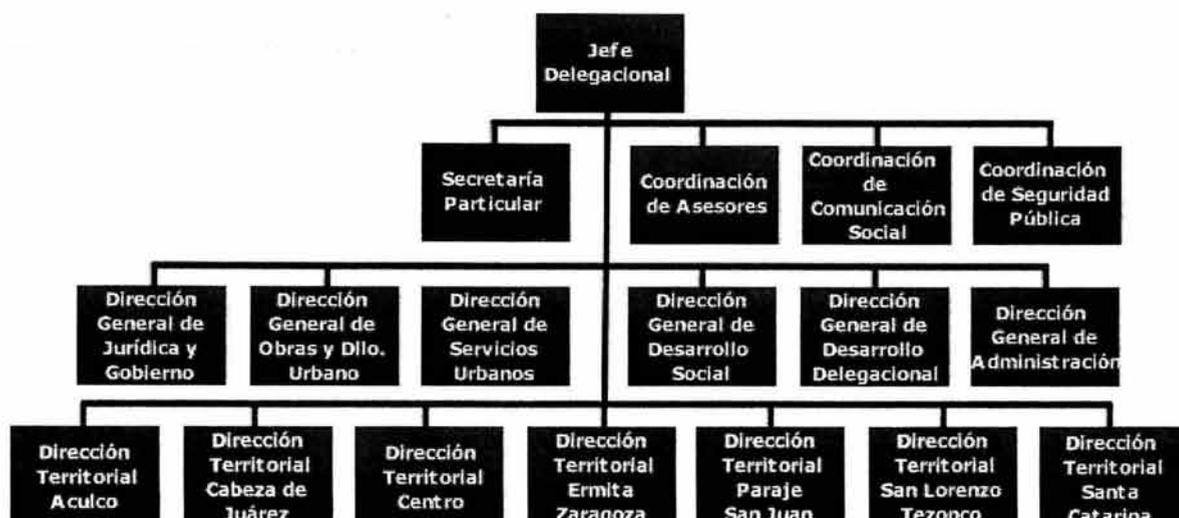
El Distrito Federal, que también se conoce como Ciudad de México, es la capital de los Estados Unidos Mexicanos, se ubica en la zona centro de la Republica Mexicana y está organizada por 16 Delegaciones Políticas, bajo la dirección del jefe de gobierno, Andrés Manuel López Obrador desde Diciembre del 2000.

La Delegación Iztapalapa es una de las Delegaciones Políticas del Distrito Federal, se encuentra situada en la región oriente de la ciudad y cuenta con una superficie aproximada de 117 kilómetros cuadrados, mismos que representan casi el 8% del territorio de la capital de la República.

Geográficamente, Iztapalapa colinda: al Norte, con la Delegación Iztacalco y el Municipio de Netzahualcóyotl del Estado de México; al Este, con los Municipios de los Reyes la Paz e Ixtapaluca del Estado de México; al Sur, con las Delegaciones Tiáhuac y Xochimilco; al Oeste, con las Delegaciones Coyoacán y Benito Juárez.



El 15 de abril del año 2003, ante el pleno de la Asamblea Legislativa del Distrito Federal tomó protesta como Jefe Delegacional en Iztapalapa el Ing. José Luis Morua Jasso. El organigrama constituido en la Delegación en esta época es el siguiente:



esquema 3.1. Organigrama de la Delegación Iztapalapa.

A diferencia de las demás delegaciones políticas, la Delegación Iztapalapa está constituida por siete delegaciones territoriales las cuales se crearon como una iniciativa por parte del anterior Jefe Delegacional, Lic. René Arce, para descentralizar las actividades de la Delegación, tener más apertura a las demandas y a las zonas de rezago de la comunidad de Iztapalapa.

III.2. SITUACIÓN ACTUAL DE LA SÍNTESIS INFORMATIVA

III.2.1. Definición

Actualmente en la Delegación Iztapalapa se encuentran varios sistemas en uso, algunos en proceso de desarrollo y otros proyectos en planeación a futuro.

Uno de los sistemas en desarrollo es el de la "Síntesis Informativa para Intranet", el cual se basa en la elaboración y reproducción de la Síntesis Informativa de la Delegación.

La Síntesis Informativa consiste en la recolección de noticias de todos los periódicos de mayor circulación en el Distrito Federal, que contengan información relacionadas con las siguientes secciones:

- Delegación.
- Gobierno del Distrito Federal.
- Partidos Políticos del Distrito Federal.
- Partidos Políticos Nacionales.
- Información General.
- Columnas, Artículos y Caricaturas.

III.2.2. Origen y objetivo

El objetivo de la Síntesis Informativa, como su nombre lo indica es informar a los altos directivos de la Delegación, sobre los sucesos que están siendo tomados en cuenta por la prensa y eventos que acontecen con relación a la Delegación, con el fin de ayudar a la toma de decisiones.

El origen de la Síntesis Informativa se establece en el año de 1990 y su elaboración está a cargo de la Coordinación de Comunicación Social, la cual se ubica dentro de las instalaciones de la Delegación.

Los pasos que llevados a cabo como base para la elaboración de la Síntesis son los siguientes:

1. Se adquieren los periódicos y se revisan los artículos
2. Si existe alguna noticia relacionada con las secciones ya mencionadas se recorta y se pega en una hoja anotando la fecha, página, sección y nombre del periódico, conforme a un formato establecido.
3. Se juntan todas las noticias agrupadas por sección a la que pertenece en un compendio por día.
4. El compendio se reproduce en fotocopias y se hace llegar al Delegado y a las principales áreas de la Delegación, incluyendo las delegaciones territoriales.

Esta forma de trabajo es la que se aplica actualmente. Aunque por otro lado, se consideró en su momento, que no sólo era importante que los altos directivos de la Delegación estuvieran en contacto directo con la Síntesis, sino que también era importante que quienes laboran en la Delegación tuvieran acceso a las noticias publicadas por los periódicos y recopiladas en el compendio diario; bajo ese esquema, se planteó la forma de como poder compartir la información a través de las diferentes áreas.

III.2.3. Síntesis Informativa en Intranet

En un principio se tomó la opción de fotocopiar la síntesis y distribuirla de esa manera a quien lo requiriera en la Delegación. Aunque la elección planteada tenía el problema de que en ocasiones no se disponía de papel suficiente para las fotocopias, o se tenían otras prioridades que atender. Debido a lo anterior, se tomó la alternativa de construir un sitio Web para la Síntesis, encontrando que a través de la red local, por medio de Intranet, se podía cumplir con el fin de compartir la información con las áreas involucradas.

Se creó entonces la Síntesis Informativa para la Intranet de la Delegación, la cual se comenzó a utilizar al inicio del año 2001.

Para poder publicar las noticias en la Síntesis de la Intranet, fue necesario digitalizar los artículos de los periódicos en archivos de imagen, esto con el fin de no transcribir el texto de la noticia y evitar alterar la emisión real del periódico. Las noticias se escaneaban a un archivo de imagen digital con formato JPG y se coleccionaban en directorios por día, al mismo tiempo se modificaba una página base en HTML que era la interfase para acceder a las noticias.

La documentación de la Síntesis solamente señala como debían almacenarse los archivos de imagen y la estructura de los directorios para guardar los archivos. Su contenido es el siguiente:

1.- Formato de imágenes: JPG

nota: No basta solo con cambiar el nombre de la extensión.

2.- Nombre: El nombre de las imágenes deberá estar escrito en minúsculas, sin acentos y sin espacios en blanco. La siguiente lista indica como deberá estar escrito el nombre de cada imagen, donde el símbolo # representa el orden en que aparecerá la imagen, siempre comenzando con el número 1.

NOMBRE DE ARCHIVO PERIODICO DE LA NOTICIA

cronica_#.jpg	Crónica
cuestion_#.jpg	Cuestión
diario_#.jpg	El Diario de México
economista_#.jpg	El Economista
eldia_#.jpg	El Día
excelsior_#.jpg	Excelsior
financiero_#.jpg	El Financiero
heraldo_#.jpg	El Heraldo de México
jornada_#.jpg	La Jornada
metro_#.jpg	El metro
mexico_#.jpg	México hoy
milenio_#.jpg	Milenio
novedades_#.jpg	Novedades
sol_#.jpg	El Sol de México
ovaciones_#.jpg	Ovaciones
prensa_#.jpg	La Prensa
reforma_#.jpg	Reforma
universal_#.jpg	El Universal
unomasuno_#.jpg	Uno Más Uno

Por ejemplo: para el periódico El Sol de México y su noticia 1, se escribe como sigue:

sol_1.jpg

Igualmente, para el periódico El Sol de México y su noticia 2, se escribe como sigue:

sol_2.jpg

ESTRUCTURA

Dentro del directorio de la Síntesis Informativa se localiza una carpeta llamada historico, en la que se encuentran las noticias de fechas anteriores, almacenadas por carpetas de mes y de día. Por ejemplo para las noticias del mes de julio se crea una carpeta llamada julio.

La carpeta julio, contiene noticias de ese mes, organizadas en subcarpetas con el nombre dia##. Por ejemplo para las noticias de fecha 3 de julio, la carpeta se llama dia03.

NOTA: Es importante escribir el dígito 0 (cero) si se trata de un día entre 1-9.

En dicho folder, se deben encontrar los documentos HTML de los 19 periódicos que sirven para mostrar las noticias. Las imágenes se encuentran en un subfolder llamado noticias.

Para actualizar el histórico, primero se crea la carpeta del día correspondiente (por ejemplo: dia11/) dentro del mes correspondiente en el folder historico, después se copian todos los archivos que se encuentran en el directorio Formatos/historicos_html/ (que se encuentra en la raíz de la síntesis informativa) al nuevo folder creado (se copia también la subcarpeta noticias).

Después se copian las noticias-imágenes que corresponden al día, dentro de la carpeta noticias del folder correspondiente.

Nota: Al copiar las imágenes a la carpeta noticias, aparece un mensaje de comprobación, que pregunta si ¿desea reemplazar el archivo existente? A lo que se debe responder que sí. Y al final en cada carpeta noticias, deberán existir 115 objetos.

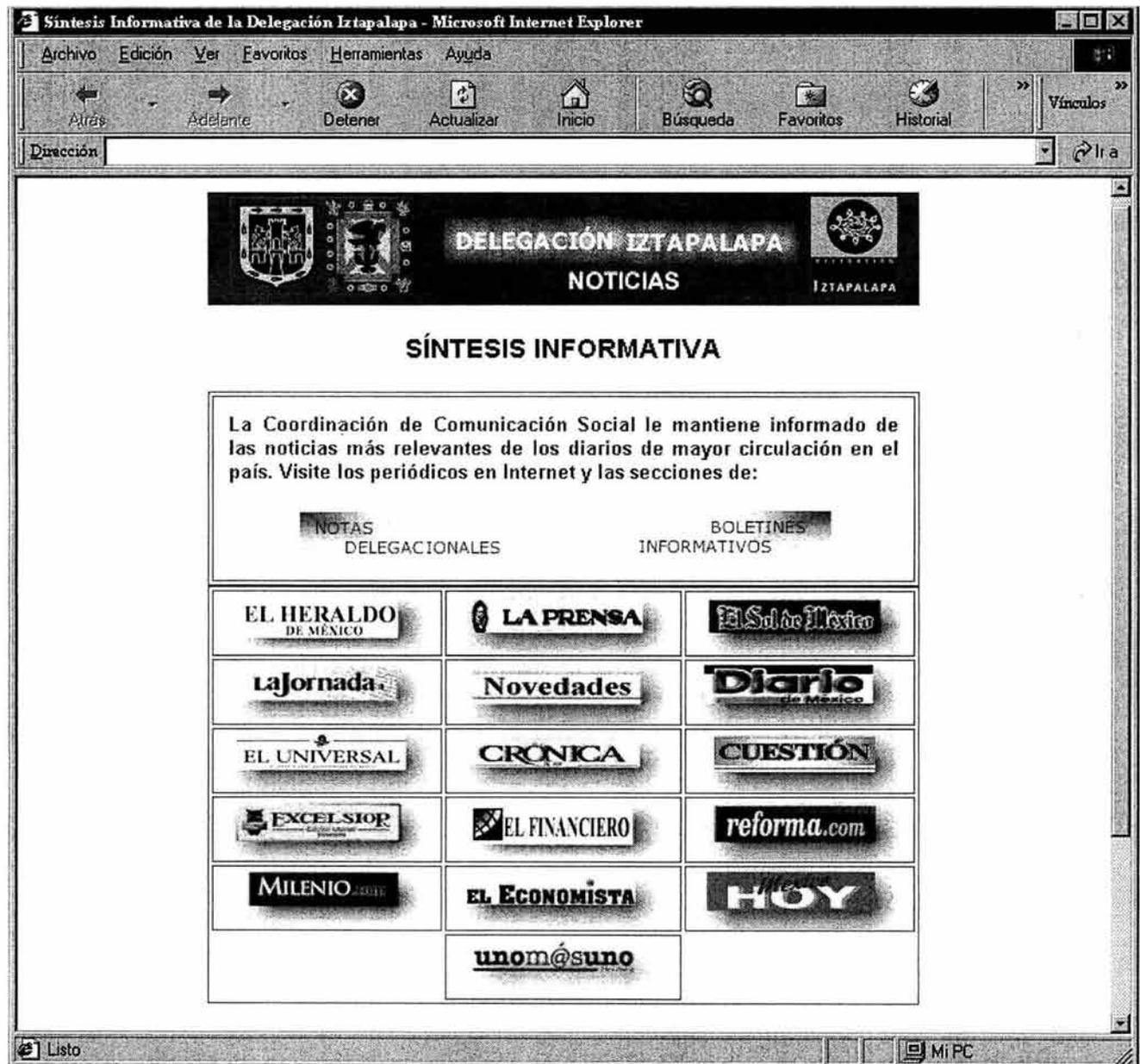
Por último, se actualiza el enlace del día nuevo correspondiente, dentro la página días.html que se encuentra en la carpeta del mes correspondiente del fólder historico. Para actualizar el enlace, se agrega un enlace en el número correspondiente, dirigido específicamente a:

dia##/heraldo_del.html

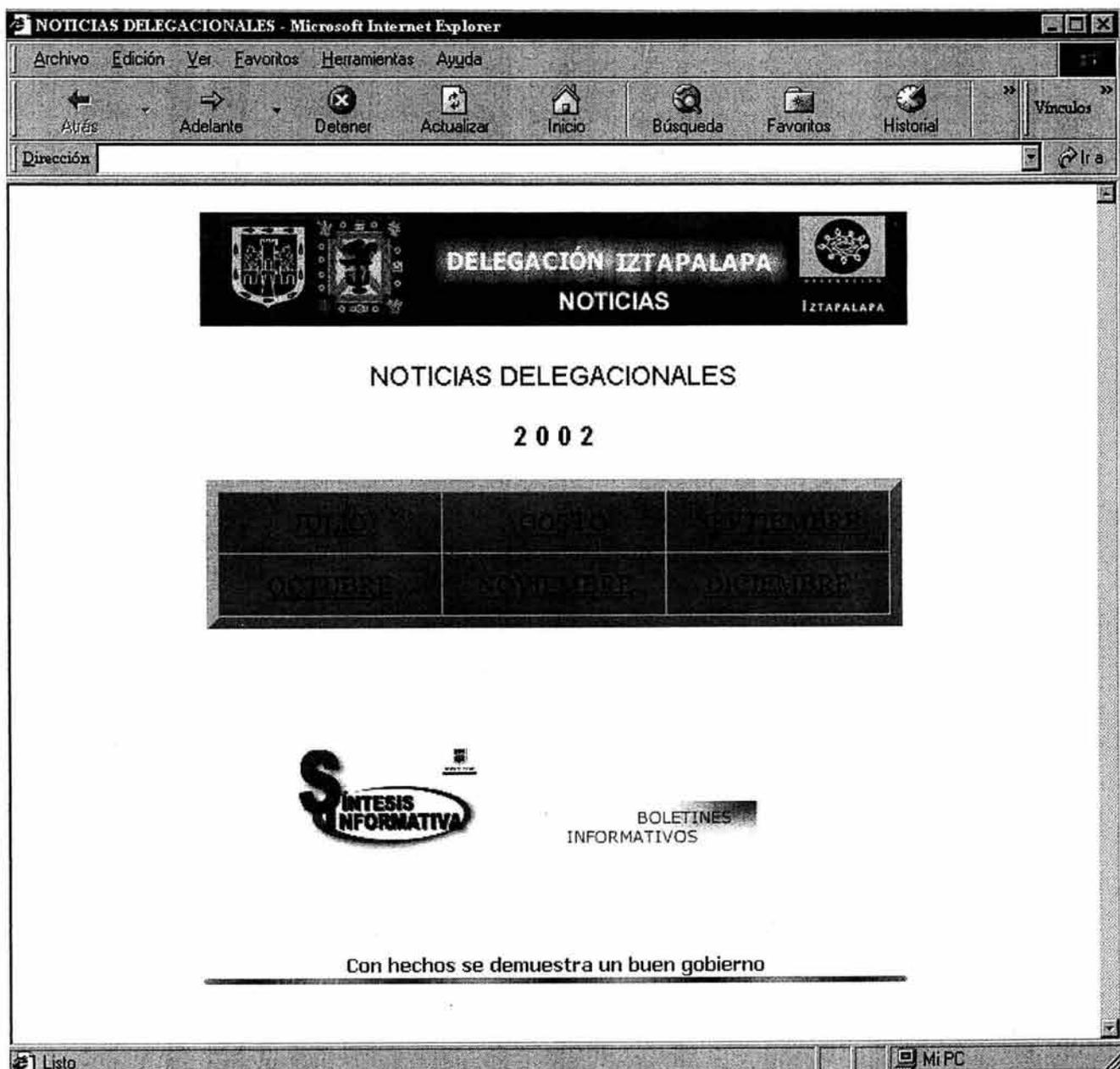
por ejemplo, para agregar el enlace de noticias del día 8 de julio, el enlace en código HTML debe ser de la siguiente forma:

 8

Ejemplo de cómo se muestra la síntesis:



esquema 3.2. Página principal, contiene enlaces a las páginas Web de los diarios contemplados dentro de la Síntesis Informativa y el enlace NOTAS DELEGACIONALES para acceder al calendario de consulta de las noticias de la Síntesis.



esquema 3.3. Página de consultas por mes.

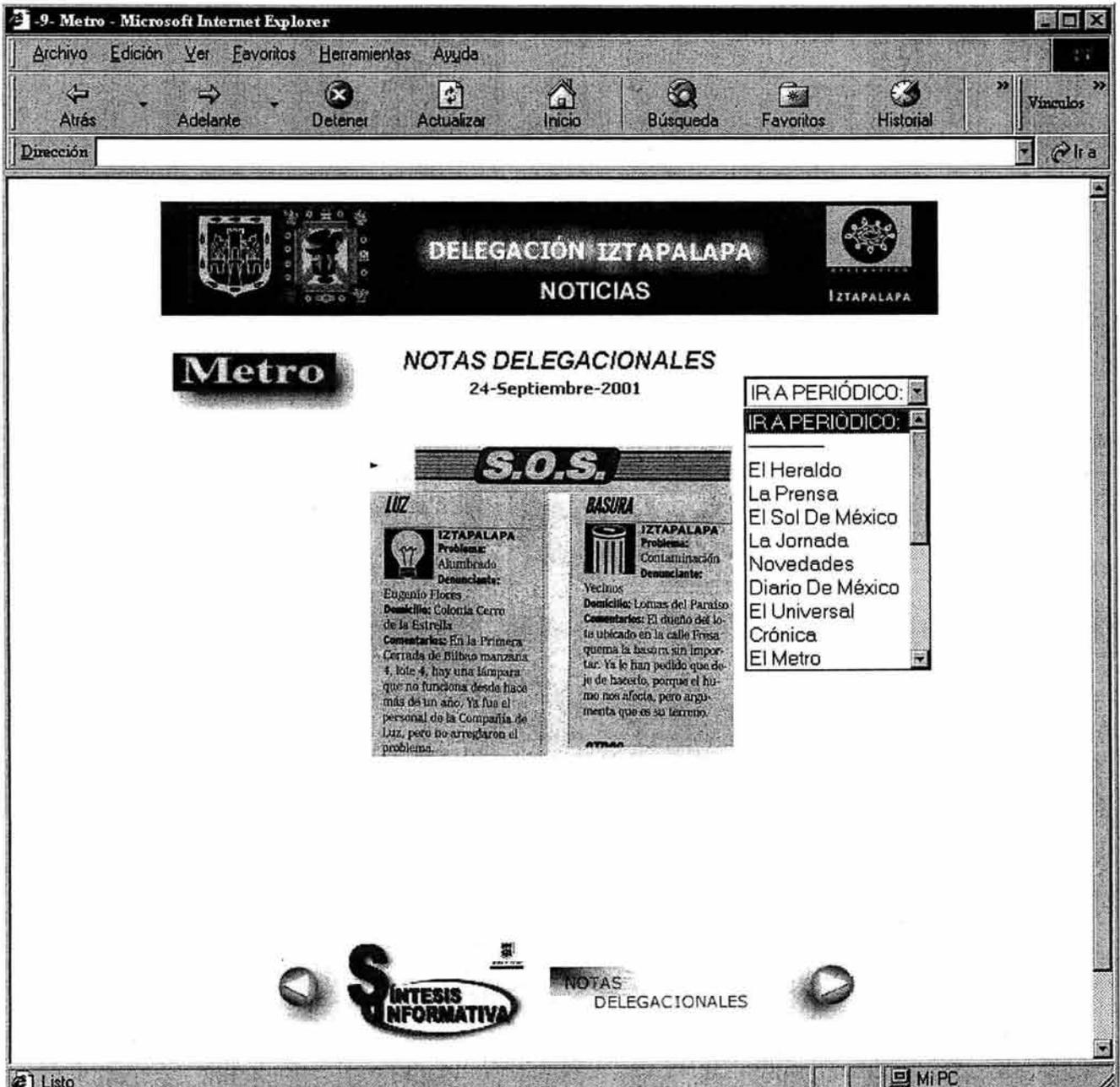
El enlace de Noticias Delegacionales de la página principal muestra la página para consulta de noticias por mes, donde al seleccionar algún mes de los presentados con enlace, se muestra la página para consulta de noticias por día.



esquema 3.4. Página de consultas por día.

Para ver las noticias de algún día del mes se selecciona el enlace correspondiente para consulta.

Por ejemplo para ver las noticias del día 24 de septiembre, se debe hacer un clic sobre el número 24. El cual contiene un enlace hacia: "dia24/heraldo_del.html", que es el archivo HTML que presenta las noticias publicadas del periódico el Heraldo de México y además es la página que inicia la secuencia de noticias.



esquema 3.5. Página ejemplo de una noticia dentro de la Síntesis.

Las noticias se muestran en orden progresivo por periódico, del 1 al 19, con ayuda de los enlaces (flechas) en la parte inferior o por medio del menú desplegable superior. Se van recorriendo los 19 periódicos y su contenido, iniciando por el periódico el Heraldo de México (heraldo_del.html).

A la Síntesis tenían acceso el jefe delegacional, el área de Comunicación Social y el área de Informática. Debido a las dificultades de comunicación con la Red Interna, no se pudo hacer llegar a las demás Delegaciones Territoriales, las cuales se encuentran fuera de las Instalaciones de la Delegación Iztapalapa.

Pero debido a la falta de interés para actualizar la Síntesis en Intranet y la falta de apoyo para su generación, se abandonó su utilización, especialmente por el manejo de la digitalización de las noticias con el uso del escáner y la generación de las páginas de enlace, no tan sencilla.

III.3 PLANTEAMIENTO DEL PROBLEMA

III.3.1. Puntos de conflicto

La Síntesis Informativa se debe hacer llegar a todas las áreas, principalmente a los directivos dentro de la Delegación y todos aquellos jefes o personal que tienen facultad para la toma de decisiones. Dado que el objetivo de la Síntesis Informativa es mantener informados a los empleados de la Delegación Iztapalapa y con ello ayudar a la toma de decisiones, es importante hacerla llegar a todas las áreas de la Delegación incluyendo las Delegaciones Territoriales en que está dividida Iztapalapa.

La distribución de la Síntesis por medio de Intranet es un recurso que cumple con el objetivo de distribuir las noticias a las áreas interesadas, pero con algunos inconvenientes debido a la estructura estática de los archivos HTML. Por ejemplo:

- Se publican los 19 periódicos contengan o no noticias.
- Se deben publicar exactamente 6 imágenes por periódico. Como plantea el funcionamiento actual: se tienen predeterminados 19 archivos de imagen con una caricatura de la Delegación, que corresponde a cada uno de los 19 periódicos y 95 imágenes en blanco.
- Cada directorio creado por día para publicar las noticias de los diarios, contiene 114 imágenes, es decir, el resultado de multiplicar 19 periódicos por 6 imágenes y además contiene 19 páginas HTML por cada periódico y un archivo imagen de la fecha de publicación.
- El nombre de archivo de las imágenes, sí indica el nombre del periódico al cual pertenece, pero no describe información acerca de la fecha de publicación.
- Si algún periódico excede en su publicación a 6 noticias, se recurre entonces a unir o “pegar” 2 o más noticias, según sea el caso, en un solo archivo, para poderlas publicar y no sobrepasar el número de archivos de imagen predeterminado.
- Las actualizaciones de los enlaces en las páginas de acceso a las noticias por día no se realizan de forma automatizada, es necesario modificar el código HTML para editar el enlace del día.
- Dentro del Site, si se desea buscar una noticia, es necesario conocer la fecha de publicación y el periódico que la realizó, es decir, las páginas no cuentan con texto informativo relacionado con lo que se está publicando.
- Otro punto importante que puede manejarse como inconveniente es la fecha de publicación que aparece en las páginas HTML de cada periódico, ésta es una imagen y no texto, es decir, como se realizan copias de las páginas HTML de los periódicos, estas contienen un enlace a un archivo de imagen llamado “fecha.jpg”. Esta imagen, debe crearse cada día que se publiquen noticias y debe contener un texto con la fecha de publicación.

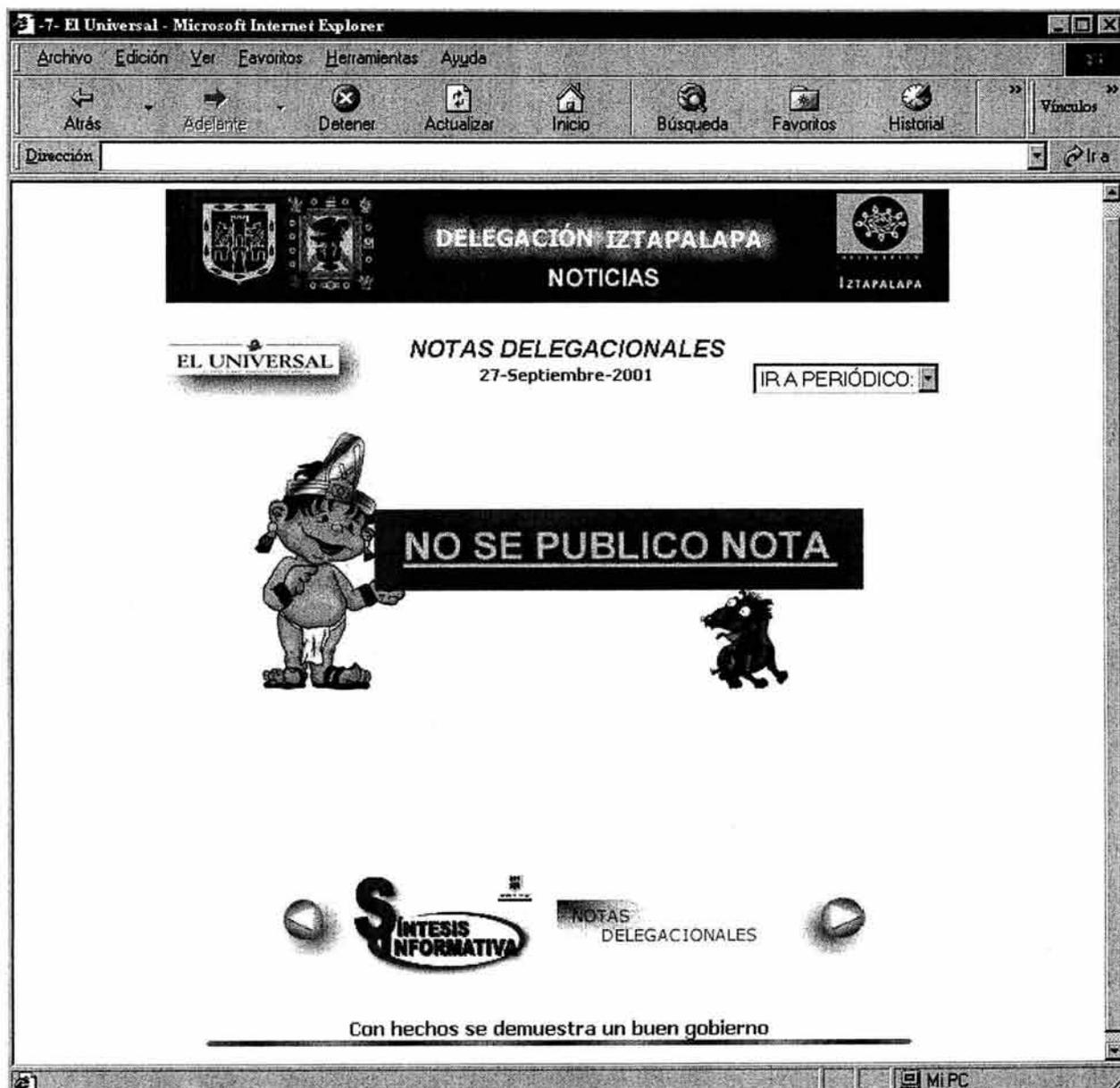
III.3.2. Limitaciones con el uso de HTML

Los problemas específicos que generan los inconvenientes señalados se concentran en los siguientes puntos:

- Almacenamiento de datos (imágenes y documentos HTML) que no son útiles
- Publicación de páginas sin contenido informativo.
- Manipulación adicional de imágenes.
- Desconocimiento de los datos que contiene cada publicación.

Estos problemas pueden pasar desapercibidos debido a que la información se genera en el servidor dentro de una red pequeña y no genera conflictos, pero si se traslada la información hacia una red más extensa, se llevaría mas tiempo de lo necesario el consultar el Site de la Síntesis, debido a que se tendrían que transmitir todas las imágenes y páginas a la máquina del cliente, aun cuando éstas no tengan utilidad. El punto principal es mostrar información con contenido informativo y que sea de utilidad.

El siguiente esquema muestra una página de un periódico (El Universal) que no publicó noticias y en su lugar la imagen que correspondería a la noticia número uno, es sustituida por la imagen de la mascota de la Delegación y las imágenes 2 a 6 restantes son sustituidas con imágenes en blanco.



esquema 3.6. La imagen número uno que correspondería a la noticia uno del periódico El Universal, se sustituye por la imagen de la mascota de la Delegación, a causa de que el periódico El Universal no publicó noticias relacionadas con la Delegación el día 27 de septiembre de 2001.

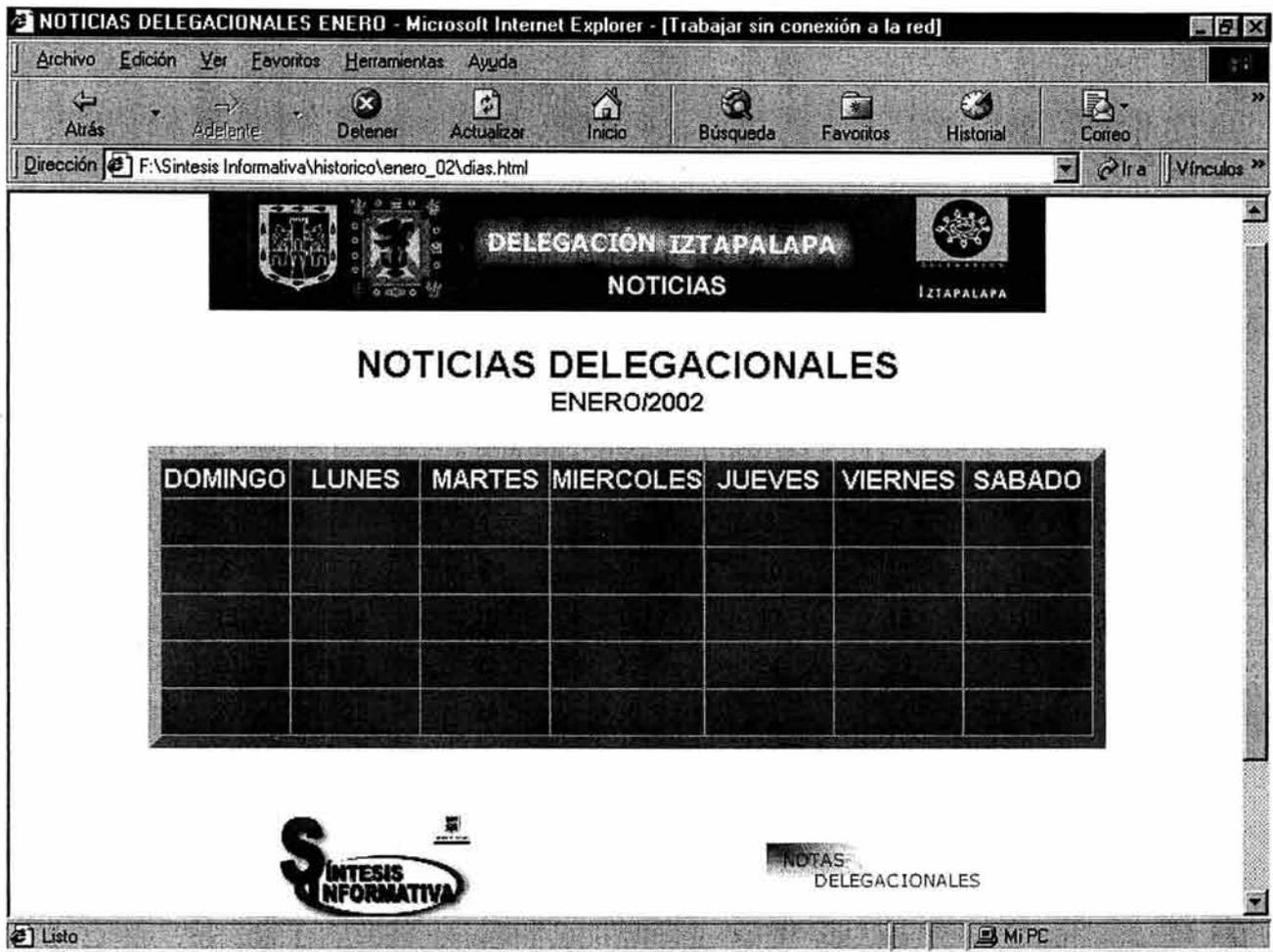
Las páginas de los periódicos que no publicaron noticias, se visualizarían como el esquema 3.6. Para el caso del periódico El Universal, que no publicó ninguna noticia, se puede notar un espacio en blanco en la parte inferior de la página como resultado de presentar cinco imágenes en blanco. Las cinco imágenes vacías, corresponden al espacio reservado para el caso de que el periódico publique más noticias acerca de la Delegación.

El esquema 3.7. muestra el espacio que ocupan las imágenes claramente.



esquema 3.7. Página del periódico El Universal sin el contenido de las imágenes de la carpeta del día de publicación de noticias, se muestra claramente las seis imágenes predeterminadas por periódico para la publicación. El lugar que ocupa la fecha de publicación también es una imagen.

El esquema 3.8. muestra la página HTML de acceso a las noticias por día. El acceso al contenido de las noticias de los periódicos se realiza por día. La actualización de esta página se tiene que realizar cada día de publicación de noticias, es decir, casi diariamente dependiendo de las publicaciones de los periódicos; sin embargo, no se realiza de forma automatizada, es decir, se requiere modificar el código en un editor de textos como indica su documentación o con ayuda de un programa de publicación de páginas como Frontpage o Netscape Composer, que no resulta fácil para el área encargada de actualizar la Síntesis Informativa.



esquema 3.8. Página de acceso a las noticias por día. La actualización de esta página cada día de publicación, no se realiza de forma automatizada.

Como se menciona en los puntos anteriores, los problemas planteados han ocasionado que la publicación de la Síntesis Informativa en Intranet se haya abandonado, además de los problemas con el manejo de los archivos de imagen y con las actualizaciones en las páginas HTML del Site.

Lo anterior hace pensar en la forma de publicación de la Síntesis Informativa como documentos estáticos, que no sufren cambios entre los días de publicación, es cierto que los elementos que forman la síntesis son los mismos día con día, pero el número de noticias es variable y el contenido siempre será diferente, comenzando por la fecha de publicación.

En el capítulo I se mencionaba acerca de las limitaciones de HTML en la generación de páginas Web, en el sentido de como paso la intención de HTML, de marcar la información de acuerdo con su significado, sin importar cómo se mostraba en la pantalla, a ser un lenguaje de marcas más dirigido al control de la presentación.

En este sentido, la Síntesis Informativa funciona para mostrar una o varias noticias de un periódico, mostrando la información como imágenes: la fecha y las noticias. Pero como se menciona en el mismo capítulo 1.5, el hipertexto es una herramienta documental que debe conceder valor añadido a la información.

Sin embargo, aun se lleva a cabo el proceso de recolección de recortes de noticias de periódicos en el compendio por día en la Delegación. Sin descartar la opción de volver a utilizar el servicio de Intranet para distribuir la Síntesis Informativa.

III.4. XML EN EL DISEÑO DE LA SÍNTESIS

III.4.1. Utilización de XML

Ante el problema planteado con la Síntesis Informativa en conservar una estructura estática, se contrapone la idea de la estructura que tiene XML para publicar documentos, una estructura tipo árbol, la cual funcionaría de forma adecuada para publicar noticias, sin importar la cantidad de periódicos con noticias, ni el número de notas que publique cada uno.

Es sabido que HTML no ofrece de forma sencilla dinamismo en la publicación, mientras que XML ofrece una solución a ese problema. Además de la ventaja que tiene con el manejo de las etiquetas, las cuales ayudan a reconocer fácilmente los datos que se estén publicando dentro de cada documento XML.

Existen en el mercado programas que ayudan a la creación de documentos XML en forma gratuita, pero no existe alguno que ayude con el manejo de imágenes sin que cueste dinero o sea fácil de manejar.

III.4.2. Planteamiento de la solución

La idea planteada es rediseñar la estructura del Site bajo XML y que con la ayuda de un programa (sencillo al usuario) genere documentos XML con ayuda visual en imágenes.

El primer punto es definir los elementos que componen los documentos de publicación de noticias. Dentro del compendio por día que se elabora con los recortes de las noticias se tienen los siguientes elementos:

- Nombre del periódico
- Fecha de publicación
- Sección donde se publicó dentro del periódico
- Página donde se publicó dentro del periódico
- Recorte de la noticia

Estos son los elementos básicos que componen una noticia de la síntesis y pueden ser representados fácilmente dentro de la estructura del nuevo documento XML.

Se encuentran también otros elementos implícitos que son:

- Noticia (recorte del periódico), que se convierte a un archivo de imagen JPG.
- Título de la noticia
- Contenido de la noticia (el tema o los temas que trata la noticia)

Para la publicación de la Síntesis con XML, se considera incluir a cada página de periódico con noticias, un encabezado y un pie de página, con el fin de dar presentación al documento y relacionarlo con el Site de la Síntesis Informativa en la Delegación Iztapalapa. El encabezado y pie de página hacen alusión a los logotipos, emblemas y lema de la Delegación, obteniendo dos elementos más:

- Encabezado de documento (emblemas de la Delegación)
- Pie de documento (Logotipo de la Síntesis Informativa y lema de la Delegación)

Finalmente se obtienen los siguientes elementos que constituyen los documentos de publicación de noticias:

1. Nombre del periódico
2. Fecha de publicación
3. Imagen o Imágenes JPG: Noticias escaneadas de recortes del periódico
4. Nombre de la sección donde se haya publicado cada noticia dentro del periódico
5. Página donde fue publicada cada noticia dentro del periódico
6. Título de cada noticia
7. Contenido de cada noticia (el tema o los temas que trata cada noticia)
8. Encabezado de documento (emblemas de la Delegación)
9. Pie de documento (Logotipo de la Síntesis Informativa y lema de la Delegación)

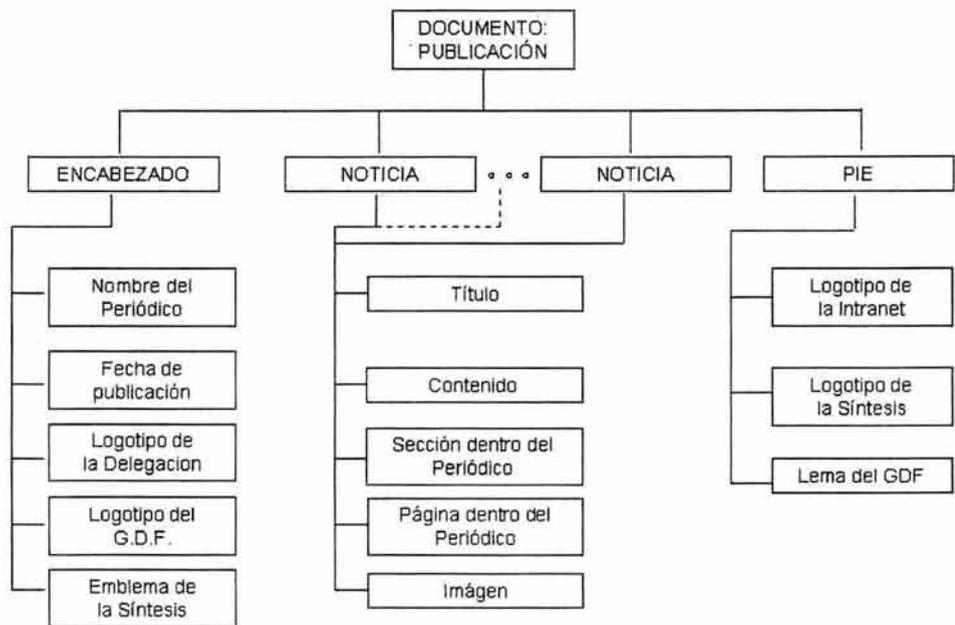
De los 9 puntos considerados, se puede notar que existen algunos elementos únicos por documento: El nombre de periódico, fecha de publicación, encabezado de documento y pie de documento. Existen elementos variables: la noticia, que puede ser sólo una o más. Partiendo de esta base, la publicación se compone de los siguientes elementos básicos:

- ENCABEZADO (elemento único)
- NOMBRE DE PERIÓDICO (elemento único)
- FECHA DE PUBLICACIÓN (elemento único)
- NOTICIA 1 (elemento variable)
- NOTICIA 2 (elemento variable)
- ...
- NOTICIA N (elemento variable)
- PIE (elemento único)

Cada elemento se compone de ciertos elementos:

- ENCABEZADO
 - Logotipo de la delegación
 - Emblema
 - Logotipo del Gobierno del D.F.
- NOMBRE DE PERIÓDICO (sin elementos)
- FECHA DE PUBLICACIÓN (sin elementos)
- NOTICIA
 - Título
 - Contenido
 - Sección dentro del periódico
 - Página dentro del periódico
 - Imagen
- PIE
 - Lema de la Gobierno del D.F. (Delegación)
 - Logotipo de la Síntesis Informativa
 - Logotipo de la Intranet

Se puede considerar el nombre del periódico y la fecha de publicación como parte del encabezado del documento, se distinguen entonces dos partes del encabezado, Logotipos e información de la página (fecha y periódico). Gráficamente, el resultado de esta estructura es el siguiente árbol:



esquema 3.9. Gráfica del árbol para la estructura de la publicación de documentos XML de noticias.

La estructura del documento XML para la publicación de noticias planteada quedaría de la siguiente forma, considerando el nombre del periódico y la fecha de publicación como parte del encabezado del documento:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<PUBLICACION>

  <ENCABEZADO>
    <LOGOS>
      <LOGO_GDF> </ LOGO_GDF >
      < EMBLEMA > </ EMBLEMA >
      < LOGO_DELEGACIÓN > </ LOGO_DELEGACIÓN >
    </LOGOS>
    <FECHA> </FECHA>
    <PERIODICO> </PERIODICO>
  </ENCABEZADO>

  <NOTICIA>
    <TITULO> </TITULO>
    <IMAGEN> </IMAGEN>
    <CONTENIDO> </CONTENIDO>
    <PAGINA> </PAGINA>
    <SECCION> </SECCION>
  </NOTICIA>

  ...

  <NOTICIA>
    <TITULO> </TITULO>
    <IMAGEN> </IMAGEN>
    <CONTENIDO> </CONTENIDO>
    <PAGINA> </PAGINA>
    <SECCION> </SECCION>
  </NOTICIA>

  <PIE>
    < LOGO_INTRANET > </ LOGO_INTRANET >
    < LOGO_SISTESIS > </ LOGO_SISTESIS >
    <LEMA> </LEMA>
  </PIE >

</PUBLICACION>
```

Para poder utilizar funciones especiales como mostrar imágenes, o establecer vínculos con otras páginas, dentro de XML es necesario utilizar elementos del lenguaje HTML. Para ello, se requiere hacerle saber a la aplicación que manejara al documento XML que se hará uso de estos elementos, en este caso el Navegador.

Esto se consigue agregando la siguiente instrucción a la etiqueta raíz del documento:

```
xmlns:html="uri:html"
```

Por ejemplo, se tiene el siguiente par de noticias publicadas el día 9 de enero de 2002 por parte del periódico "El Sol de México", el documento que consigue mostrar la publicación de las noticias del periódico es el siguiente:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet href="articulo.css" type="text/css" ?>
<PUBLICACION xmlns:html="uri:html">
  <ENCABEZADO>
    <LOGOS>
      <LOGO_GDF>          <html:img SRC="logo_Delegación.jpg"></html:img> </LOGO_GDF>
      <EMBLEMA>          <html:img SRC="banner.gif " "></html:img>          </EMBLEMA>
      <LOGO_DELEGACION> <html:img SRC="logo_iztapalapa.jpg"></html:img> </LOGO_DELEGACION>
    </LOGOS>
    <FECHA>          2 de enero de 2002 </FECHA>
    <PERIODICO>      El Sol de México </PERIODICO>
  </ENCABEZADO>

  <NOTICIA>
    <TITULO>          Seguirá la PFP en el DF hasta fin de mes          </TITULO>
    <IMAGEN>          <html:img src="sol1-1-2.jpg"></html:img>          </IMAGEN>
    <CONTENIDO>      El GDF y la SSP acordaron que el operativo de vigilancia implementado por la Policía
                    Preventiva se extiende hasta fin de mes.          </CONTENIDO>
    <PAGINA>          Página: 13          </PAGINA>
    <SECCION>         Sección: Ciudad          </SECCION>
  </NOTICIA>

  <NOTICIA>
    <TITULO>          Defienden los delegados colores de sus partidos </TITULO>
    <IMAGEN>          <html:img src="sol2-1-2.jpg"></html:img>          </IMAGEN>
    <CONTENIDO>      Rechazan panistas que sus nuevos diseños tengan fines de proselitismo político, sino sólo
                    son usados para diferenciarse del gobierno local. </CONTENIDO>
    <PAGINA>          Página: 14          </PAGINA>
    <SECCION>         Sección: Ciudad          </SECCION>
  </NOTICIA>

  <PIE>
    <LOGO_INTRANET> <html:img src="../../imagenes/logo_Intranet.jpg"> </html:img> </LOGO_INTRANET>
    <LOGO_SINTESIS> <html:img src="../../imagenes/logo_sintesis.jpg"> </html:img> </LOGO_SINTESIS>
    <LEMA>           Con hechos se demuestra un buen gobierno          </LEMA>
  </PIE>
</PUBLICACION>

```

Para mostrar el documento XML de forma presentable en un programa Navegador, es necesario darle formato de salida para su presentación, para ello, se incluye la siguiente línea de código justo antes de la etiqueta raíz dentro del documento:

```

<?xml-stylesheet href="articulo.css" type="text/css" ?>

```

Que indica que el documento XML se apoyará en un archivo de hoja de estilos CSS llamada "articulo.css" para su presentación. El contenido de la hoja de estilo se explica más adelante.

Los documentos XML de cada periódico que publica noticias, se almacenarán en carpetas por día, conforme una estructura de archivos y directorios definida para el Site de la Síntesis Informativa.

III.4.3. Estructura del Site

El Site contiene la estructura básica de la síntesis actual. Publicar noticias por año, mes y día.

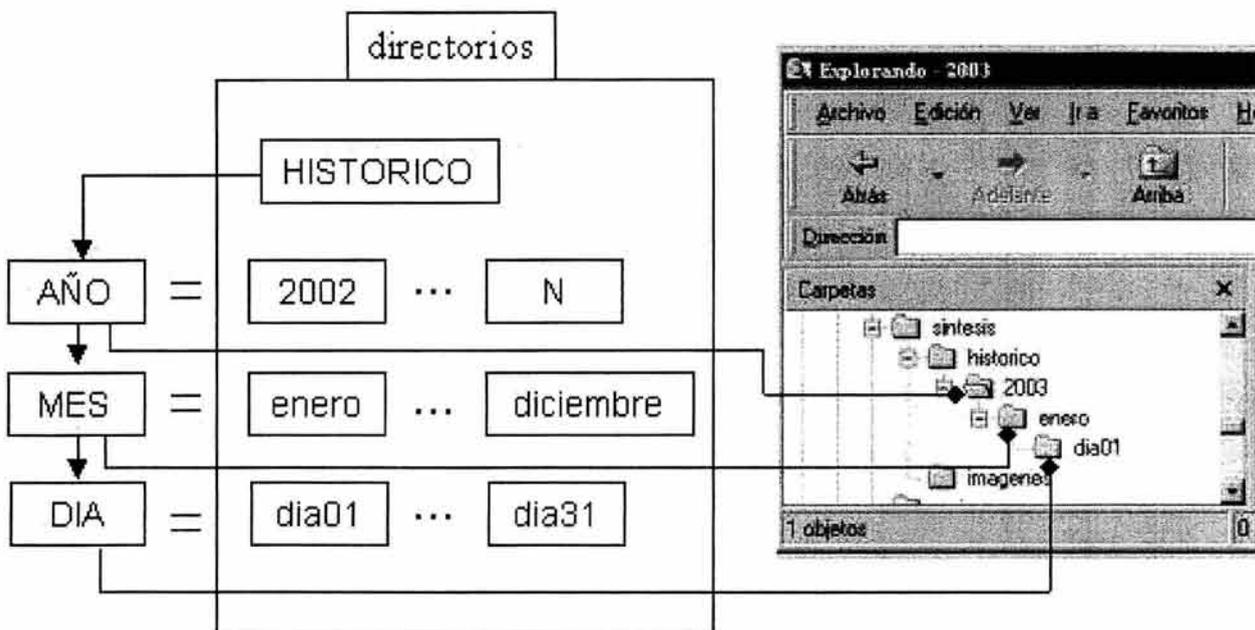
Primero, se debe tener un directorio designado específicamente para almacenar y generar los archivos que compondrán el Site de la Síntesis Informativa. En este caso el espacio reservado está dentro de la carpeta *sisntesis* del servidor.

Dentro de este directorio se debe encontrar el archivo *index.html*, que es la página de inicio o presentación de la Síntesis Informativa. El contenido de esta página, debe enlazar al calendario de consultas de noticias por año. El directorio *sisntesis*, deberá contener dos directorios para distribución de la información:

- 📁 *historico*, donde se almacenaran las noticias por carpetas de mes y de día;
- 📁 *imagenes*, que contendrá todos los logotipos utilizados dentro del Site.

Como ya se mencionó, la distribución de las noticias publicadas se realizará por año, por mes y por día. Por ejemplo para las noticias del año 2003 se debe tener una carpeta llamada *2003*, la cual contendrá una carpeta por cada mes de publicación. Por ejemplo, se puede tener el fólder de *julio* para que contenga las noticias del mes de julio de 2003, todo dentro de la carpeta *historico*.

La carpeta *julio*, contiene noticias de ese mes, organizadas en subcarpetas de cada día, el nombre de la subcarpeta se forma de la palabra *dia* junto con el número del día correspondiente, dando el formato *dia##*. Por ejemplo para las noticias de fecha 3 de julio, la carpeta se llama *dia03*. Escribiendo el dígito 0 (cero) si se trata de un día entre 1 y 9 antes del número correspondiente.



esquema 3.10. Estructura Lógica de los directorios dentro del Site de la Síntesis Informativa.

Es importante mantener la estructura descrita, porque las páginas del Site, contienen enlaces internos hacia elementos que se tienen ubicados en otros directorios, específicamente las imágenes y logotipos, archivos que deberán estar ubicados en la carpeta *Imágenes* de la raíz de la Síntesis, lo cual permite que por medio de dichos enlaces o links se evite hacer copias no necesarias de estos elementos en otras carpetas para su correcta visualización. También es necesario que los nombres de carpetas y de archivos que se generen, se escriban sin acentos ni símbolos de puntuación, esto con el fin de que las referencias a los enlaces de las páginas funcionen de manera adecuada.

Tomando la estructura descrita anteriormente, el contenido de cada directorio es el siguiente:

-  **sintesis:** contiene la carpeta *historico* y la carpeta imágenes. Contiene también el archivo *index.html* que es la página principal o de inicio de la Síntesis Informativa.
-  **imagenes:** logotipos de los periódicos tomados en cuenta para la generación de la Síntesis Informativa, logotipos de la Delegación Iztapalapa, logotipos del Gobierno del D.F., logotipos del Site de la Síntesis Informativa y logotipo de Intranet de la Delegación Iztapalapa.
-  **historico:** contiene los directorios de cada año que tienen noticias. Además, en este directorio se almacenan los archivos *articulo.css* (que es la hoja de estilo que utilizan los documentos XML para su presentación), *paramet.xml* (los parámetros de periódicos para la generación de documentos), *paralogo.xml* (parámetros de imágenes, y logotipos para el encabezado y pie de documentos) e *historico.html* (que es la página de interfase para acceder a los años de consulta). El contenido de estos archivos se explican en la siguiente sección de este capítulo III.
-  **número de año (2002, 2003, ...):** estas carpetas contendrán una carpeta por cada mes de publicación, además, un archivo llamado *meses.xml* (que se genera con la aplicación) y un archivo llamado *meses.html*, donde ambos archivos son en conjunto la página de interfase para acceder a los meses de consulta)
-  **mes (enero ... diciembre):** cada uno de estos folders tendrá una carpeta por cada día de publicación, además, un archivo llamado *dias.xml* (que se genera con la aplicación) y un archivo llamado *dias.html*, donde ambos archivos son en conjunto la página de interfase para acceder a las noticias o documentos XML de publicación de noticias de cada periódico.
-  **día (dia01 ... dia31):** cada una de estas carpetas, deberá contener todas las imágenes escaneadas de las noticias publicadas por los periódicos del día correspondiente. En esta carpeta es donde se generan también, los documentos XML que muestran las noticias correspondientes por día.

Finalmente, se ha considerado que puede ocurrir que alguno de los periódicos de circulación en el D.F. termine su publicación, se tengan nuevos periódicos o cambien de nombre algunos. Se debe contemplar que se puedan agregar nuevos diarios en circulación o eliminar periódicos.

De igual forma es sabido que los logotipos de las delegaciones y lemas de gobierno, son susceptibles a sufrir cambios junto con un cambio de administración o gobierno, por eso es importante mantener e identificar estos elementos de manera separada.

III.5 DESARROLLO DE LA SÍNTESIS

III.5.1. Generación de la estructura y actualización

El punto principal es publicar las noticias dentro del Site de la Síntesis Informativa. Las noticias se van generando día con día y con ello se tienen que crear las páginas que sirven de interfase para publicar las noticias de cada periódico. Además, como no es constante el número de periódicos que publican noticias, ni de la misma forma es constante el número de notas que publicará cada periódico, no se puede tener predeterminada la publicación diaria.

Se recurre entonces a crear las carpetas de año, de mes y de día conforme se va avanzando cada año. Poniendo un ejemplo amplio, considérese que el día de mañana es 1 de enero de 2003, entonces dentro de la carpeta *historico* se tendrán que crear 3 carpetas. Primero crear el directorio *2003*, dentro de este directorio se debe generar el folder *enero* y finalmente se crea la carpeta *dia01* en el interior de este último. Se debe de actualizar entonces el enlace dentro del archivo *historico.html* cuyo contenido es el siguiente:

```

<html> <head> <title>NOTICIAS DELEGACIONALES</title> </head>
<body>
<center>
<table WIDTH="480" BGCOLOR="#336666" bordercolor="#336666">
<tr> <th>
<img SRC="../imagenes/logo_gdf.jpg" height="67" width="124">
<img SRC="../imagenes/emblema.gif" height="55" width="252">
<img SRC="../imagenes/logo_izt.jpg" height="67" width="58">
</th></tr>
</table>
<P> <font size="4" face="Arial" color="#336666">CALENDARIO DE CONSULTAS POR AÑO</font> </P>
<table BORDER="8" CELLSPACING="0" CELLPADDING="10" COLS="3" BGCOLOR="#09989F">
<tr ALIGN="CENTER" VALIGN="CENTER" HEIGHT="30">
<td WIDTH="130"><a href="2001/meses.html"><font color="#000099" size="+1">2001</font></a></td>
<td WIDTH="130"><a href="2002/meses.html"><font color="#000099" size="+1">2002</font></a></td>
</tr>
</table>
<table cellpadding="20">
<tr>
<td><a href="../index.html">
<img SRC="../imagenes/image_001.jpg" ALT="PAGINA DE INICIO" BORDER="0"></a>
</td>
<td>
<a href="../boletines/boletines.html">
</a>
</td>
</tr>
</table>
<br>
<table cellpadding="20">
<tr> <td>
<font face="Tahoma" size="2" font color="#336666">
<b>Con hechos se demuestra un buen gobierno</b>
</font>
<BR>
</td> </tr>
</table>
<br>
</center>
</body>
</html>

```

Para actualizar la página hacia las consultas del nuevo año 2003, se consigue modificando el archivo de la siguiente forma:

1.- Se abre en un editor de textos planos (no procesador de palabras), como Notepad o Wordpad, el archivo historico.html.

2.- Se ubica y copia el renglón del último año actualizado, que para el ejemplo descrito debe ser el año 2002, siendo la siguiente línea:

```
<td WIDTH="130"><a href="2002/meses.html"><font color="#000099" size="+1">2002</font></a></td>
```

3.- En el renglón siguiente, se pega o escribe el mismo texto que la línea 2002, solo que modificando el año **2002** por el nuevo año, es decir, 2003.

4.- Se guarda el archivo y se verifica la actualización en el navegador.

Dentro del nuevo directorio llamado 2003 se debe generar el archivo "meses.html" que servirá como acceso a los meses que se deseen consultar dentro del año 2003. Que para el ejemplo planteado solamente corresponde al mes de enero. El contenido básico de este archivo es el siguiente:

```
<html> <head> <title>CALENDARIO ANUAL</title> </head>
<body>
<center>
<font color="#336660" size="5" face="Arial"> <b>CALENDARIO</b> </font>
<xml id="calendario" src="meses.xml"></xml>
<table datasrc="#calendario" border="8" cellpadding="0" cellspacing="0" bgcolor="#09989F">
<tbody>
<tr>
<th align="center" width="80"><font color="#000099">
<table datasrc="#calendario" datafld="uno">
<tr><td> <a href=<span datafld="enlace"></span><span datafld="mes"></span></a>
</td></tr>
</table> </font>
</th>
<th align="center" width="80"><font color="#000099">
<table datasrc="#calendario" datafld="dos">
<tr><td> <a href=<span datafld="enlace"></span><span datafld="mes"></span></a>
</td></tr>
</table> </font>
</th>
<th align="center" width="80"><font color="#000099">
<table datasrc="#calendario" datafld="tres">
<tr><td> <a href=<span datafld="enlace"></span><span datafld="mes"></span></a>
</td></tr>
</table> </font>
</th>
<th align="center" width="80"><font color="#000099">
<table datasrc="#calendario" datafld="cuatro">
<tr><td> <a href=<span datafld="enlace"></span><span datafld="mes"></span></a>
</td></tr>
</table> </font>
</th>
</tr>
</tbody>
</table>
</center>
</body>
</html>
```

Para tener acceso a las noticias de mes, se debe generar dentro del directorio llamado *enero* el archivo "dias.html", el cual contiene una tabla de enlaces por día en forma de calendario. Para el ejemplo planteado se tiene acceso solamente al día primero de enero. El contenido de este archivo es el siguiente:

```
<html> <head> <title>CALENDARIO</title> </head>
<body>
<center>
<font color="#336660" size="5" face="Arial"> <b>CALENDARIO</b> </font>
<xml id="calendario" src="dias.xml"></xml>

<table datasrc="#calendario" border="8" cellpadding="0" cellspacing="0" bgcolor="#09989F" cols="7">
<thead>
<tr>
<th align="center" width="80" height="25"><font color="#FFFFFF" face="Arial">
<b>DOMINGO</b></font></th>
<th align="center" width="80" height="25"><font color="#FFFFFF" face="Arial">
```

```

<b>LUNES</b></font></th>
<th align="center" width="80" height="25"><font color="#FFFFFF" face="Arial">
  <b>MARTES</b></font></th>
<th align="center" width="80" height="25"><font color="#FFFFFF" face="Arial">
  <b>MIERCOLES</b></font></th>
<th align="center" width="80" height="25"><font color="#FFFFFF" face="Arial">
  <b>JUEVES</b></font></th>
<th align="center" width="80" height="25"><font color="#FFFFFF" face="Arial">
  <b>VIERNES</b></font></th>
<th align="center" width="80" height="25"><font color="#FFFFFF" face="Arial">
  <b>SABADO</b></font></th>
</tr>
</thead>

<tbody>
<tr>
<th align="center" width="80"><font color="#000099">
  <table datasrc="#calendario" datafld="domingo">
    <tr><td> <a href=<span datafld="enlace"></span><span datafld="dia"></span></a>
    </td></tr></table> </font>
</th>
<th align="center" width="80"><font color="#000099">
  <table datasrc="#calendario" datafld="lunes">
    <tr><td> <a href=<span datafld="enlace"></span><span datafld="dia"></span></a>
    </td></tr></table> </font>
</th>
<th align="center" width="80"><font color="#000099">
  <table datasrc="#calendario" datafld="martes">
    <tr><td> <a href=<span datafld="enlace"></span><span datafld="dia"></span></a>
    </td></tr></table> </font>
</th>
<th align="center" width="80"><font color="#000099">
  <table datasrc="#calendario" datafld="miercoles">
    <tr><td> <a href=<span datafld="enlace"></span><span datafld="dia"></span></a>
    </td></tr></table> </font>
</th>
<th align="center" width="80"><font color="#000099">
  <table datasrc="#calendario" datafld="jueves">
    <tr><td> <a href=<span datafld="enlace"></span><span datafld="dia"></span></a>
    </td></tr></table> </font>
</th>

<th align="center" width="80"><font color="#000099">
  <table datasrc="#calendario" datafld="viernes">
    <tr><td> <a href=<span datafld="enlace"></span><span datafld="dia"></span></a>
    </td></tr></table> </font>
</th>
<th align="center" width="80"><font color="#000099">
  <table datasrc="#calendario" datafld="sabado">
    <tr><td> <a href=<span datafld="enlace"></span><span datafld="dia"></span></a>
    </td></tr></table> </font>
</th>
</tr>
</tbody>
</table>
</center>
</body>
</html>

```

Para el ejemplo descrito, el fin es publicar las noticias del día 1 de enero de 2003, por el momento el archivo días, no contiene ningún enlace debido a la suposición de que el día 1 de enero apenas va a comenzar y por tanto se desconoce si habrá al menos un periódico que publique noticias con relación a la Delegación Iztapalapa. De ser así, el contenido de la página días.html será quedara sin modificaciones.

Si para el día 1 de enero de 2003 sí existen noticias publicadas en los diarios del D.F. referentes a la Delegación, entonces el proceso llevado a cabo después de tener la estructura antes planteada es el siguiente:

1. Se genera el compendio de noticias de la Síntesis Informativa, con los recortes de las notas de los periódicos en los formatos establecidos.
2. Con ayuda del compendio, se escanean o digitalizan los recortes para obtener imágenes con formato JPG por cada nota publicada.
3. Los archivos de imagen obtenidos, se deberán almacenar dentro de la carpeta del día de publicación. Para el ejemplo planteado, deberán estar contenidos dentro de la ruta: *historico/2003/enero/dia01/*. Es en ese momento cuando se crea el folder *dia01*
4. El nombre de la imagen de cada noticia deberá llevar una palabra principal del nombre del periódico y un orden progresivo de numeración, en el caso de que un mismo periódico publique varias noticias y el número del día junto con el número del mes de publicación, separados por guiones. Para el ejemplo planteado, si el periódico "El universal" publicó dos noticias, los archivos de imagen de noticias se nombrarán: *universal1-1-1.jpg* y *universal2-1-1.jpg*, lo cual indica noticia 1 del primero de enero y noticia 2 del primero de enero.
5. Se crean los documentos XML que funcionan como interfase para la consulta de noticias con ayuda de la aplicación de la Síntesis Informativa. Con lo cual se debe actualizar el enlace del día correspondiente en el documento *dias.xml* después de capturar y generar los documentos XML por medio de la aplicación.
6. Finalmente, se debe visualizar el resultado en el navegador para verificar la actualización en el Site de la Síntesis.

La aplicación utilizada para generar los documentos XML está creada en Visual Basic 6.0. El programa funciona como interfase para la creación de la o las páginas que publican las noticias diariamente dentro del Site. El programa funciona únicamente para sistemas basados en Windows de Microsoft.

Se ha comentado anteriormente sobre la compatibilidad de XML con los navegadores para páginas Web. El Site de la Síntesis Informativa podrá visualizarse correctamente sólo con el navegador de Internet Explorer versión 5.0. o versiones posteriores a ésta.

El programa se puede ejecutar desde cualquier directorio o desde el servidor de red. Pero es necesario indicar al programa la ruta donde se localiza la síntesis informativa para que la publicación se realice correctamente. Además, para su correcto funcionamiento se debe contar con la estructura de directorios ya descrita y con los siguientes archivos dentro del directorio de *historico*:

- **paramet.xml.** Archivo XML de texto plano, de este archivo, se toma la cantidad y nombres de los periódicos contemplados para publicar noticias, el cual tiene una estructura de documento XML donde su etiqueta nombre indica el nombre de cada periódico para publicación de noticias.
- **paralogo.xml.** Archivo XML de texto plano, de este archivo, se toma los nombres de archivo de las imágenes emblemas y logotipos, además de contener el lema de la Delegación. Se toma como parámetros que pueden ser modificados o cambiados en cualquier momento.

Contenido del Archivo de parámetros (Paramet.xml)

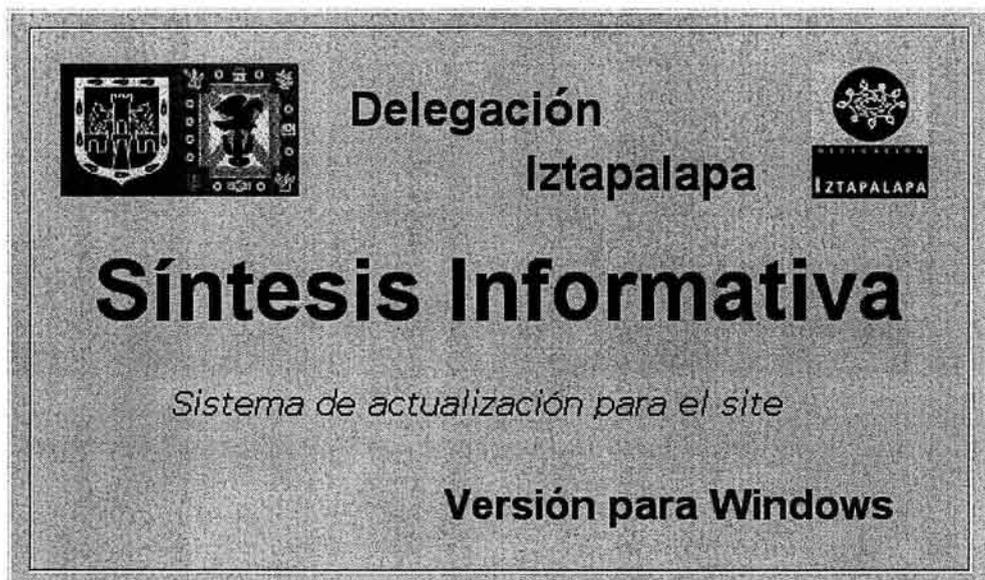
```
<?xml version="1.0" >
<PERIODIOCOS>
<PERIODIOCO>
<NOMBRE>Universal</NOMBRE>
<NOMBRE>Novedades</NOMBRE>
<NOMBRE>El sol de mexico</NOMBRE>
<NOMBRE>El diario de Mexico</NOMBRE>
</PERIODIOCO>
</PERIODIOCOS>
```

Contenido del Archivo de parámetros (Paralogo.xml)

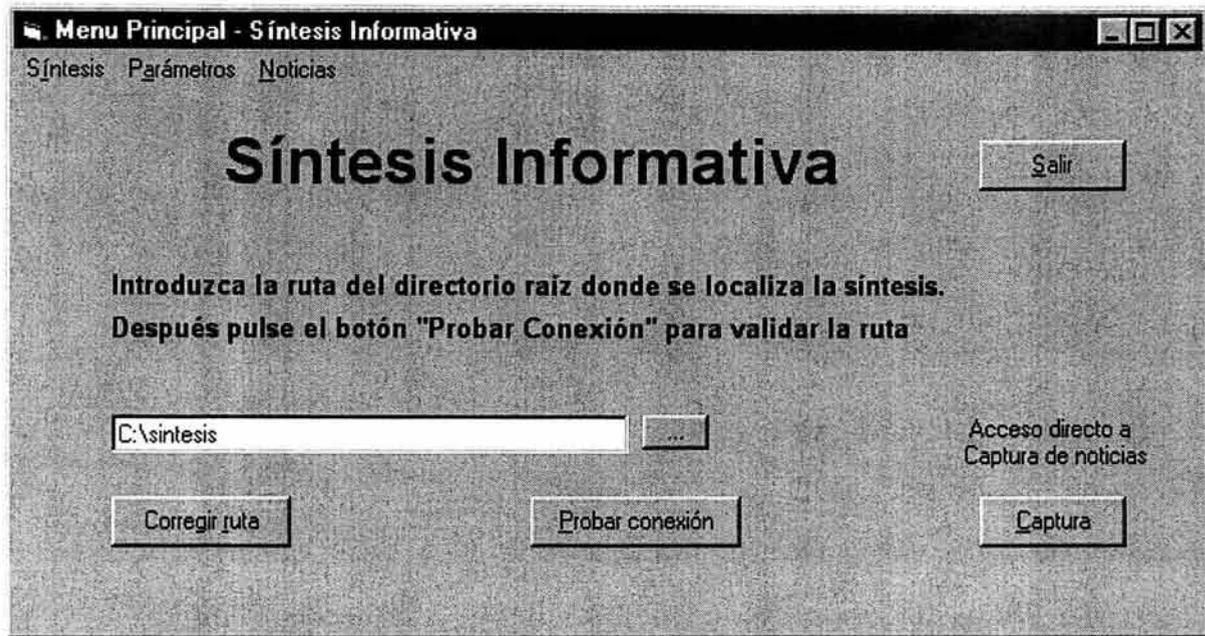
```
<?xml version="1.0" ?>
<IMAGENES>
<ENCABEZADO>
<LOGO_GDF> gdf_logo.jpg </LOGO_GDF>
<EMBLEMA> emblema.gif </EMBLEMA>
<LOGO_DELEGACION> logo_izt.jpg </LOGO_DELEGACION>
</ENCABEZADO>
<PIE>
<LOGO_INTRANET> Intranet.jpg </LOGO_INTRANET>
<LOGO_SISTESIS> síntesis.jpg </LOGO_SISTESIS>
<LEMA> Con hechos se demuestra un buen gobierno </LEMA>
</PIE>
</IMAGENES>
```

III.5.2. Uso del Sistema de Actualización para la página

Al iniciar el programa, se muestra la pantalla de inicio en forma de presentación y enseguida se abre la pantalla del menú principal, la cual contiene varias opciones para trabajar.



esquema 3.11. Pantalla de Presentación.



esquema 3.12. Menú principal de la aplicación.

El primer paso para poder trabajar con la aplicación, es realizar una prueba de conexión, con el fin de verificar la estructura de los directorios contenidos dentro de la Síntesis y señalar al programa el directorio sobre el cual se va a trabajar.

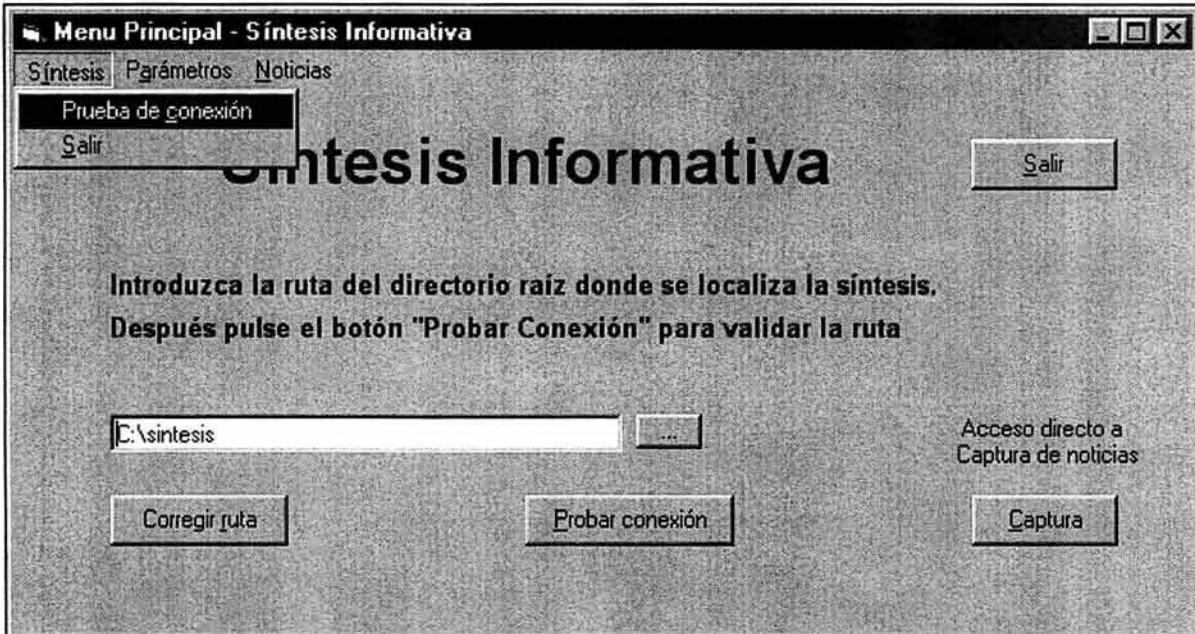
Para realizar la prueba de conexión, se debe indicar primero la ruta de trabajo donde se localiza la Síntesis Informativa para Intranet.

La ruta establecida por defecto es `c:\síntesis` dentro del directorio de la computadora donde se ejecuta el programa, si ésta no es la ubicación donde se localiza el Site de la síntesis, se debe modificar utilizando el botón **Corregir Ruta** e introducir la correcta en el espacio en blanco reservado. Por ejemplo, una ruta puede ser: `"a:\site\síntesis\"`.

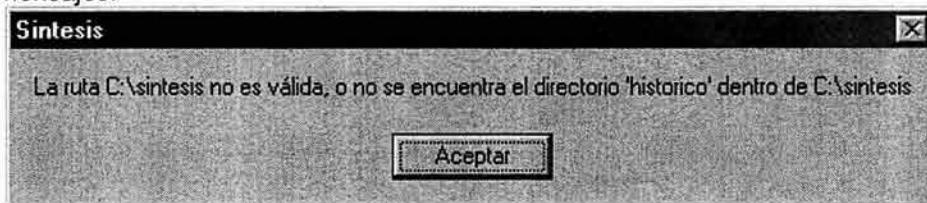
También se puede seleccionar la ruta correcta con ayuda del botón de directorio  situado junto al espacio en blanco de la ruta, el cual muestra el siguiente cuadro de diálogo, donde se puede seleccionar el directorio de forma visual:



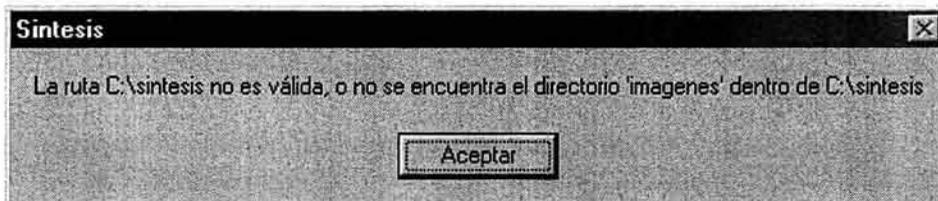
Una vez contenida la ruta en el espacio reservado, se debe validar con ayuda del botón **Probar Conexión** o por medio del menú principal seleccionando **Síntesis** y después **Prueba de conexión**.



Si al realizar la prueba de conexión el sistema determina que la ruta introducida no es válida aparece uno de los siguientes mensajes:

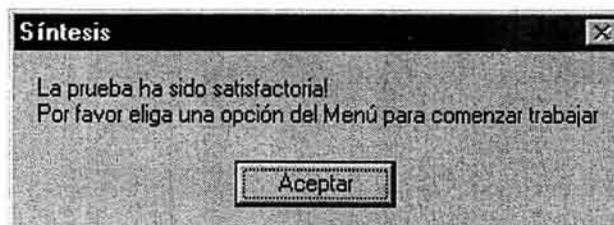


Este mensaje indica que no localizó la ruta "C:\síntesis" que se especificó en el espacio en blanco reservado o que no se encontró el directorio "historico" dentro del directorio "C:\síntesis". Entonces corresponde al usuario, verificar la ruta establecida y modificarla.

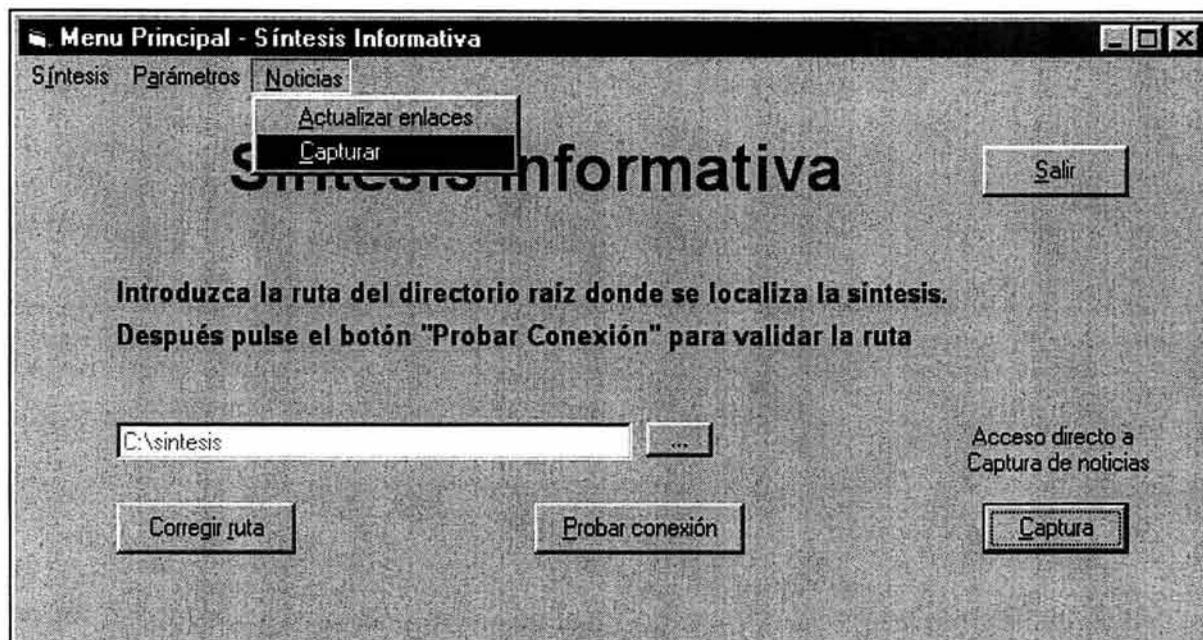


Como indica el mensaje, el sistema indica que no se localizó el directorio "imagenes" dentro del directorio "C:\síntesis". Corresponde al usuario en este caso, verificar la ruta establecida y modificarla.

Si la ruta introducida es correcta aparece el siguiente mensaje, lo cual indica que la estructura de la Síntesis es correcta:

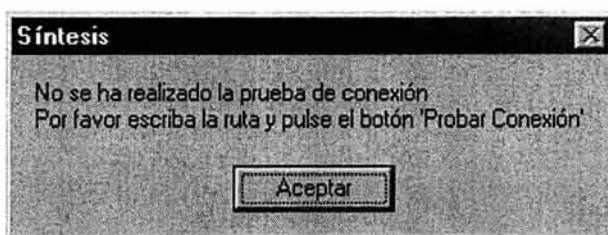


Con la validación de la ruta, se puede trabajar con el sistema de captura de noticias. Para ello se tienen dos opciones, una es a través del menú en la parte superior y la otra por medio del botón de Captura. Indicado por la leyenda Acceso directo a Captura de Noticias



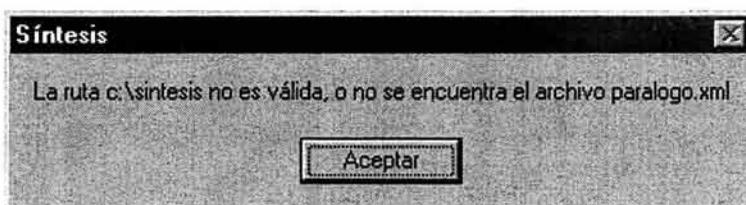
esquema 3.13. Acceso a la captura de Noticias.

Cabe señalar que si no se ha realizado y validado la prueba de conexión, no se pueden utilizar las demás opciones del menú y aparece el siguiente mensaje de error.

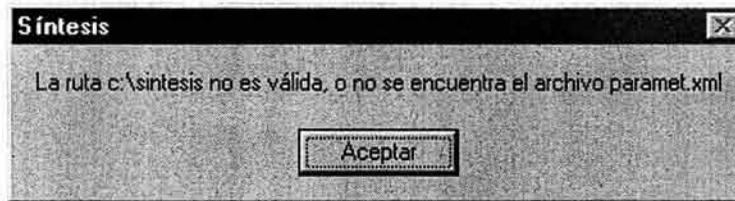


También es importante hacer mención, que para continuar con la captura de noticias, deben existir dentro del directorio de *historico*, los archivos *paramet.xml* y *paralogo.xml*. Para poder capturar noticias, debe haber periódicos registrados en el archivo *paramet.xml* y deben establecerse las imágenes y logotipos que se utilizarán en la generación de las páginas de noticias (encabezado y pie de los documentos).

Si al acceder a la captura de Noticias el sistema no localiza el archivo correspondiente manda alguno de los siguientes mensajes y regresa al menú principal



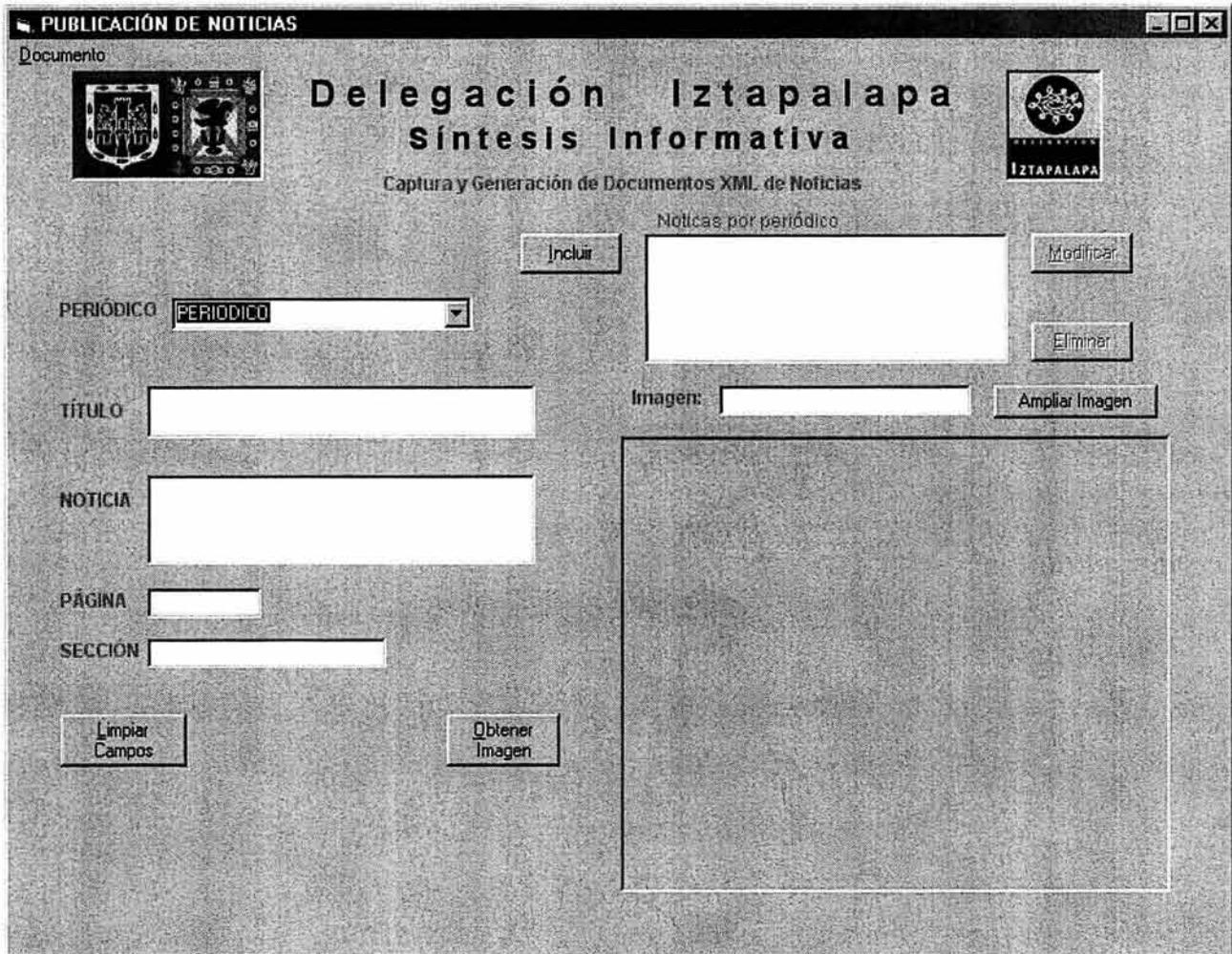
Lo cual indica que no localizó el archivo *paralogo.xml* dentro de la ruta *C:\síntesis*.



Este mensaje indica que no localizó el archivo paramet.xml dentro del directorio C:\síntesis.

Para el caso de que el sistema envíe alguno de estos mensajes, se debe utilizar la opción del menú Parámetros para establecer Los Periódicos, el encabezado de documentos y el Pie de documentos. Estas opciones se explican más adelante en esta sección,

Si se realizó la prueba de conexión correctamente y no se tienen problemas con los archivos de parámetro, se abre entonces la pantalla de captura de noticias siguiente:



esquema 3.14. Pantalla de captura de noticias por periódico.

Los artículos se capturan mediante un formulario que contiene el nombre del periódico de la noticia, el título de la noticia, contenido básico, la sección y página donde fue publicada en el periódico, así como el archivo de imagen que contiene la noticia (previamente escaneada del recorte de periódico y almacenada en el directorio del día correspondiente).

Primero se debe seleccionar el periódico que publicó noticias en el día, para posteriormente capturar la o las noticias que publicó con sus respectivas imágenes. Esto se logra, seleccionando por medio del listado de Periódicos el correspondiente nombre del diario que publica la noticia.

PERIÓDICO

TÍTULO

NOTICIA

Después de seleccionar el periódico, se realiza la captura de los datos de la noticia en los campos: Título, Noticia, Página y Sección. La selección del archivo de imagen que contiene la noticia, puede hacerse antes, durante o después de la captura de los campos mencionados. El poder visualizar la noticia antes de capturar, puede ayudar al usuario a relacionar el contenido a capturar con la noticia (imagen).

PUBLICACIÓN DE NOTICIAS

Documento

Delegación Iztapalapa
Síntesis Informativa
 Captura y Generación de Documentos XML de Noticias

Noticias por periódico

PERIÓDICO

TÍTULO

NOTICIA

PÁGINA

SECCIÓN

Imagen:

Minimizan manipulación

POR ARTURO PÁRAMO

LA CONTROVERSI SURGIDA A RAÍZ DEL uso de los colores del PAN y del PRD en fachadas, papelería y mantas en algunas Delegaciones del DF fue minimizada por los dirigentes de esos partidos en la capital, quienes señalaron que lo importante es la eficiencia de las administraciones.

José Luis Luege y Carlos Imaz, presidentes del PAN y del PRD del DF, respectivamente, señalaron que no existe una legislación que obligue a homogeneizar los colores y logotipos que se utilizan para identificar a las administraciones.

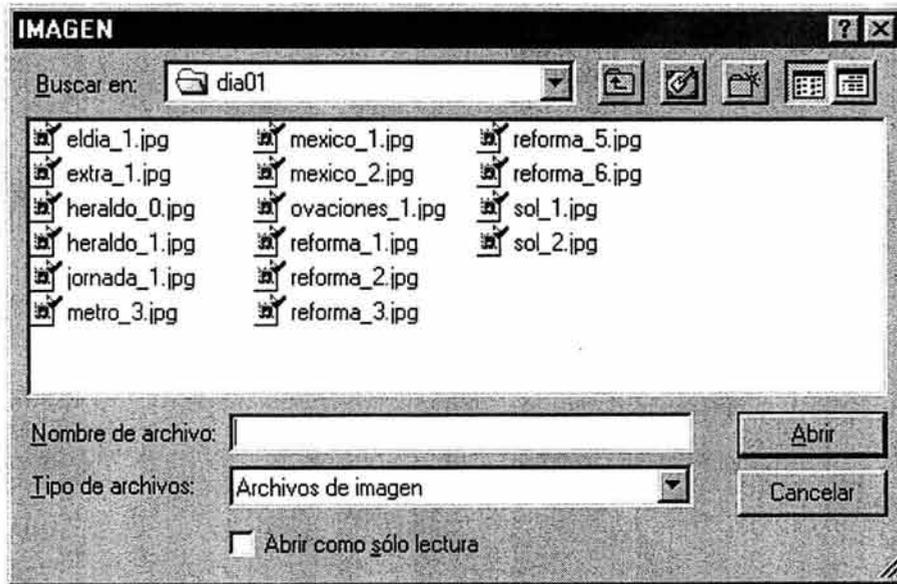
"Para mí no tiene mayor importancia. Como partido no damos indicación alguna al respecto. Cada delegación tiene una imagen propia, como imagen corporativa, como logotipo. También el Gobierno del DF ha tomado una imagen propia."

Un puente peatonal en la Gustavo A. Madero

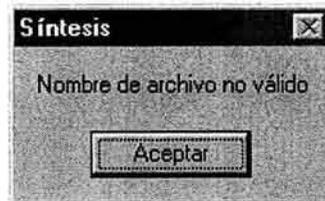
esquema 3.15. Llenado del contenido de una noticia de periódico.

El botón ubicado en la parte inferior izquierda llamado "Limpiar campos", realiza la función de borrar el contenido de los campos durante la captura incluyendo la imagen. Se puede utilizar en cualquier momento de la captura y sirve en el caso de que se hayan introducido datos erróneos en los campos de captura.

Para seleccionar el archivo de imagen que contiene la noticia, se debe utilizar el botón "Obtener Imagen", o el método abreviado de teclas <Alt> <M>. Con lo cual se despliega un cuadro de diálogo para seleccionar primeramente el directorio del día correspondiente a la publicación y después se debe escribir o seleccionar el nombre del archivo de imagen que contiene la noticia a publicar.



El sistema verifica que el archivo establecido sea válido, si se escribe un nombre de archivo no existente y se selecciona el botón Abrir, se genera el siguiente mensaje de error, cerrando el cuadro de diálogo y mostrando nuevamente la pantalla de captura. Se puede volver a seleccionar el archivo de imagen con ayuda del el botón "Obtener Imagen" como se explicó anteriormente.



Después de seleccionar el archivo de imagen de la noticia, en la pantalla del formulario se muestra la noticia (en forma parcial al tamaño del cuadro) y se observa también el nombre del archivo en la parte superior. Si se hubiese elegido una imagen no deseada, se puede repetir el proceso de "Obtener Imagen" y seleccionar otro archivo de imagen.



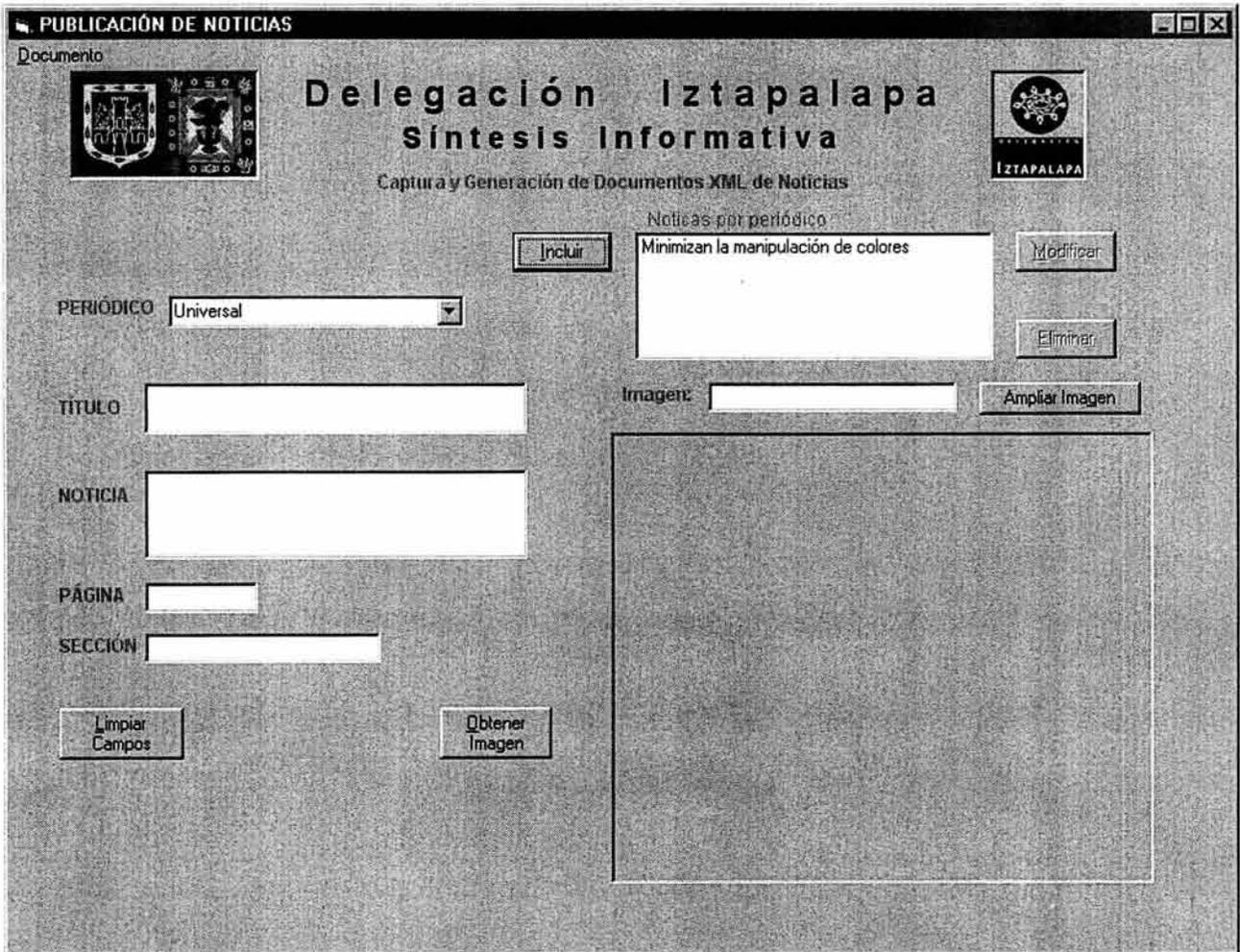
Se puede conseguir visualizar de forma completa la imagen, en el caso de que se requiera, con ayuda del botón Ampliar Imagen o el método abreviado <Alt><G>. Con lo cual se despliega la siguiente ventana con el nombre de la imagen previamente seleccionada como referencia:



Una vez visualizada la imagen, se debe cerrar la ventana, ya sea utilizando el botón "Cerrar" o el botón de cruz de la barra superior de la ventana.

Cuando se ha llenado el formulario y seleccionado la imagen, se debe incluir la noticia en el listado de noticias por periódico. Esto se consigue con ayuda del botón "Incluir" o el método abreviado <Alt><I>. Con lo cual aparecerá en el cuadro "Noticias por periódico" de la derecha, un renglón con el título de la noticia capturada y se pondrán los campos del formulario en blanco, listos para registrar otra noticia.

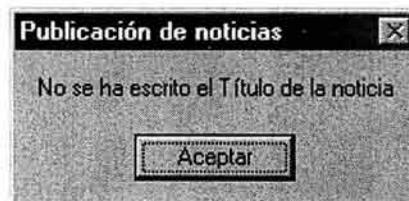
La pantalla de captura se visualizaría de la siguiente forma:



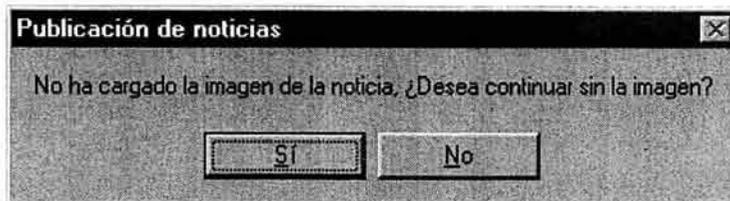
esquema 3.16. Noticia incluida dentro de un periódico.

En el caso de que se quiera incluir una noticia sin haber concluido la captura, el sistema enviará un mensaje de error dependiendo del dato dentro del formulario que fue omitido.

Por ejemplo, el siguiente mensaje indica que se hizo acción del botón incluir noticia sin haber capturado el Título en el formulario.

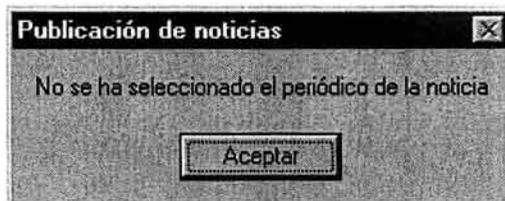


De la misma forma, si no se ha seleccionado una imagen para la noticia, se envía el siguiente mensaje de error:

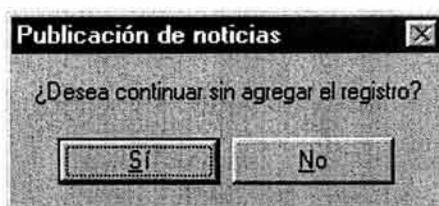


Al seleccionar la opción "SI", el sistema procederá a incluir la noticia al periódico sin relacionarla con una imagen. Si se requiere que se publique alguna noticia, aún cuando no se tenga escaneada su imagen, entonces el sistema brinda la opción para hacerlo. Por otro lado, se pudo haber omitido la imagen por error, si este es el caso, se debe seleccionar la opción "NO", con lo cual, el sistema no incluye la noticia al periódico y continua la captura para poder seleccionar una imagen con ayuda del botón Obtener Imagen.

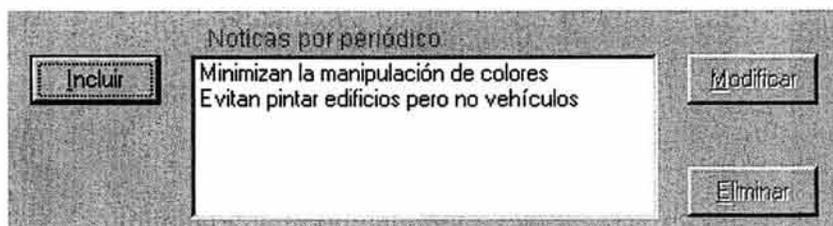
En el momento de la captura, se puede modificar por error el contenido del nombre del periódico seleccionado y si después se procede a incluir la noticia al periódico, el sistema enviará un mensaje de error indicando que no se ha seleccionado el periódico de la noticia debido a la causa explicada. También puede ocurrir que al cargar la pantalla de captura se proceda directamente a escribir el contenido de los campos sin antes haber seleccionado el periódico correspondiente, a lo cual el sistema enviara el mismo mensaje.



Es importante hacer la aclaración, de que si no se agrega la noticia al listado de noticias por periódico y se selecciona otro periódico, se perderá la información capturada. Cuando sea éste el caso, el sistema manda el siguiente mensaje, en el cual pregunta si realmente se desea cambiar de periódico sin antes haber agregado la noticia al listado, dando la opción de continuar sin almacenar lo capturado o de no cambiar de periódico para poder agregarle la noticia capturada.



Se puede hacer una revisión de las noticias capturadas. Para ello, se debe primeramente seleccionar un periódico en el que ya se haya registrado alguna noticia, si el periódico contiene noticias capturadas, se mostraran los títulos en el listado Noticias por Periódico



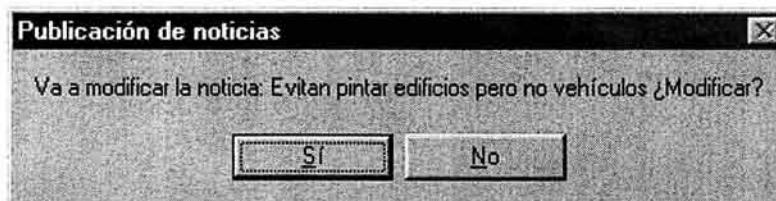
Después, se debe seleccionar la noticia que se quiere mostrar seleccionando el renglón correspondiente del listado de noticias por periódico. Al realizar esta acción, el resultado es visualizar los datos en los campos del formulario y la imagen de la noticia.

Al mostrar una noticia como consulta se debe notar que el espacio donde se muestra el nombre de la imagen queda en blanco sin mostrar el nombre y solo mostrando la imagen, esto indica que la noticia mostrada es de consulta y no de captura.

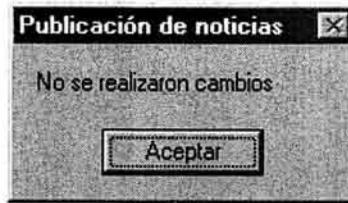


esquema 3.17. Mostrando una noticia de periódico ya capturada.

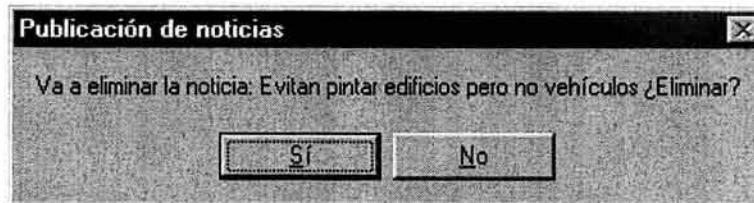
Si existiera alguna noticia ya registrada en la que se requiera modificar su contenido, se puede realizar la acción seleccionando primero el periódico correspondiente que contiene la noticia, después se debe seleccionar la noticia a modificar dentro del listado Noticias por periódico y finalmente se procede a corregir la información, que puede ser la imagen o los datos del formulario; realizada la corrección, se registra la modificación con el botón "Modificar" o el método abreviado <Alt><M>. El sistema confirmará que se va a modificar la noticia y el programa enviará el siguiente mensaje.



Si se selecciona la opción "SI", se registran los cambios y se limpian los campos para continuar con otra captura. De lo contrario si se selecciona la opción "NO", se envía el siguiente mensaje e igualmente se limpian los campos para continuar capturando.



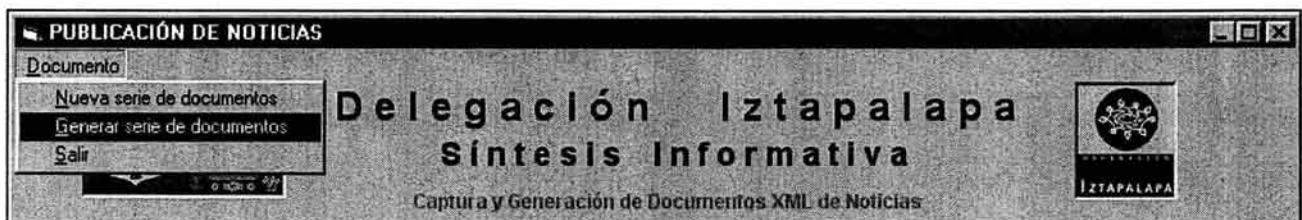
Si existiera alguna noticia ya registrada que se quiera eliminar del periódico, se debe seleccionar el periódico y enseguida una noticia de la manera descrita anteriormente y después se elimina con ayuda del botón "Eliminar" o el método abreviado <Alt><E>. El sistema confirmará que se quiere eliminar la noticia y enviará un mensaje de confirmación.



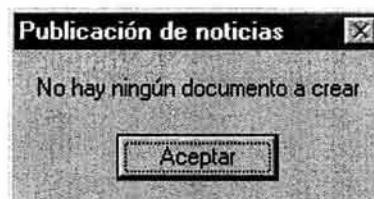
Una vez finalizada la captura, se deben generar los documentos XML, los cuales contendrán el nombre de cada periódico y la fecha establecida de publicación con el siguiente formato:
periodico #dedía _ #demes.xml.

Los documentos se crearán en la misma carpeta de donde se obtuvieron las imágenes, es decir deberá ser el directorio del día de publicación de las noticias (dia##) dentro del folder del mes correspondiente. El documento que inicia la secuencia de noticias llevará el nombre del primer periódico del listado de periódicos con noticias capturadas.

Para generar los documentos se debe utilizar el menú Documento ubicado en la parte superior derecha de la ventana y seleccionar la opción Generar serie de documentos.

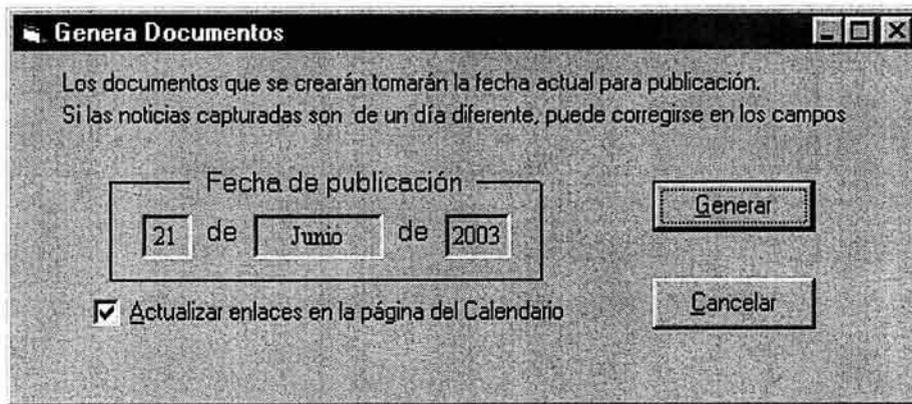


Si ocurriera el caso de que no existieran noticias registradas en ningún periódico, el programa enviaría un mensaje de error indicando que no se puede generar ningún documento de noticias precisamente porque no existen noticias capturadas. Se mostraría el siguiente mensaje:

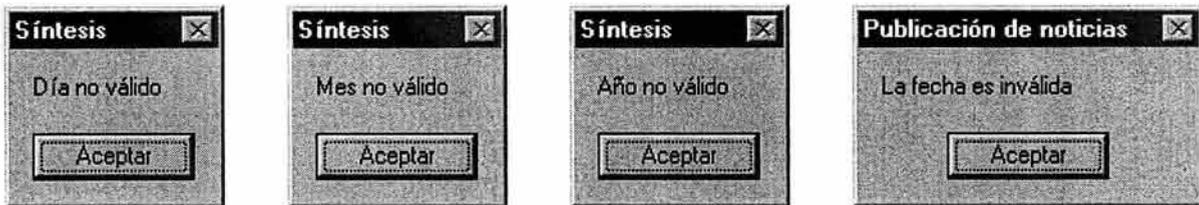


En el caso de que se hayan capturado y registrado las noticias a publicar de un día específico, después de seleccionar la opción para Generar la serie de documentos, se muestra un cuadro de diálogo, donde se debe indicar la fecha de publicación a la cual pertenecen las noticias capturadas, además, se debe señalar si se desea actualizar el calendario de acceso a las noticias, con el fin de que los documentos generados se

puedan acceder a través del Site en Intranet. El cuadro de diálogo para generar los documentos es el siguiente:



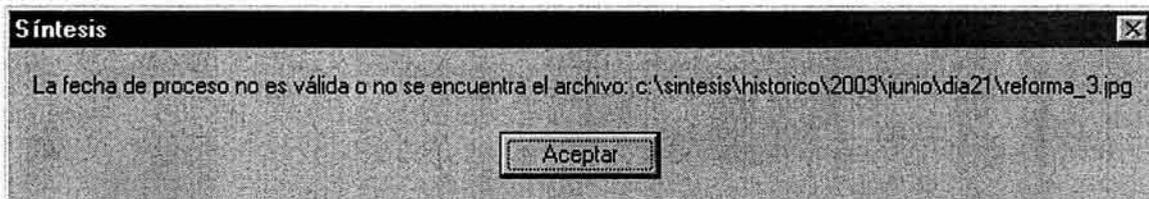
Por defecto, se establece la fecha de generación igual a la fecha actual de sistema de la computadora donde se ejecuta la aplicación. Si la fecha de las noticias capturadas es una distinta a la del sistema, se debe corregir en los campos ubicados dentro del recuadro Fecha de publicación. Si se introduce un valor no válido dentro de los campos del día, mes o año, el sistema envía alguno de los siguientes mensajes y espera a que el usuario introduzca un valor válido para el día, mes o año.



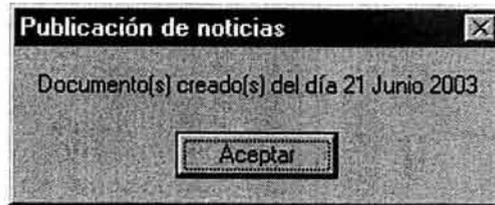
Al activar la casilla de verificación ubicada en la parte inferior del cuadro de diálogo para generar los documentos, se establece que se desea que el Site de la Síntesis Informativa se enlace a las noticias generadas del día correspondiente, con el fin de que se puedan acceder a través del calendario de noticias por mes, es decir la página dias.html.

Una vez establecida la fecha correcta, se deben generar los documentos con ayuda del botón Generar.

El sistema verifica que las imágenes de noticias seleccionadas se localicen dentro del directorio especificado con la fecha de publicación establecida. Si existiera alguna imagen que no se localice en el directorio del día correspondiente, se envía un mensaje de advertencia como el siguiente, señalando el nombre de la imagen no encontrada:



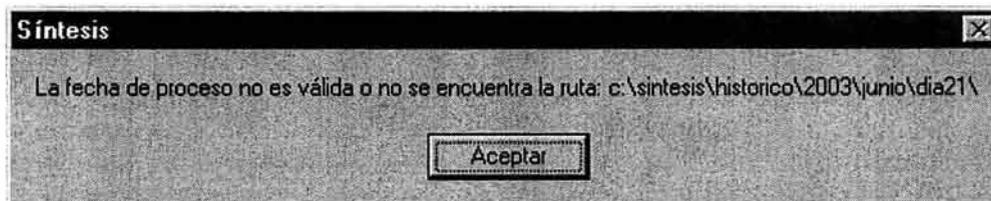
Si no existen problemas con los archivos de imagen, el sistema envía un mensaje de confirmación, indicando que no hubo error en la captura y que además, se generaron correctamente los documentos en la carpeta del día correspondiente conforme el día especificado.



Si se activó la casilla de verificación para actualizar en enlace del día, se genera el siguiente mensaje de confirmación, indicando que no se encontró ningún error y se realizó correctamente la actualización de enlaces.

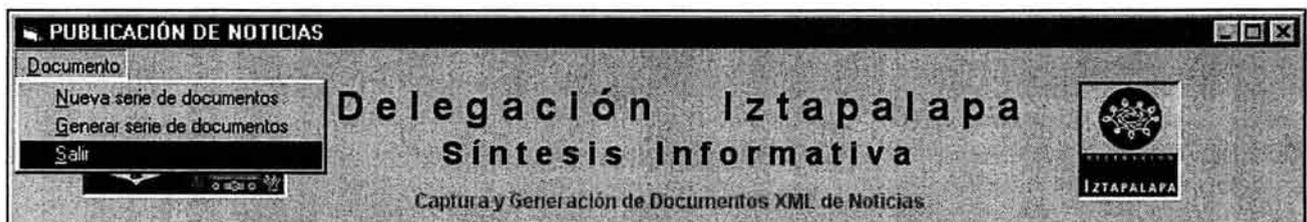


El sistema verifica que la fecha introducida al momento de generar los documentos, corresponda con la estructura de carpetas dentro del directorio de la Síntesis. Si dentro del directorio historico no se localiza el directorio del año correspondiente, el mes o el día de la fecha introducida, se genera el siguiente mensaje de error, donde se especifica cual es la ruta que se esperaba encontrar y no fue localizada.



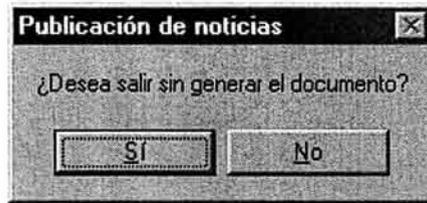
Es recomendable una vez que ya se generaron los documentos, visualizar la salida en el Site de la Síntesis Informativa con ayuda del navegador, esto con el fin de verificar el número de documentos que se generaron por periódico y la cantidad de noticias que se publicaron en cada uno. Si existiera algún error en alguno de los periódicos o en alguna de las noticias, se pueden realizar correcciones sobre los datos capturados siempre que no se haya cerrado la pantalla de Captura de Noticias por Periódico y no se haya generado una nueva serie de documentos. Para realizar alguna corrección, se debe proceder conforme lo descrito anteriormente.

Si los documentos generados del día correspondiente son correctos, entonces se puede cerrar la pantalla de captura para finalizar con la actualización del día y posteriormente cerrar la aplicación por completo. Existen dos formas para cerrar la ventana de Publicación de Noticias: a través del menú Documento y la opción Salir, o con el botón de cruz de la barra superior de la ventana.



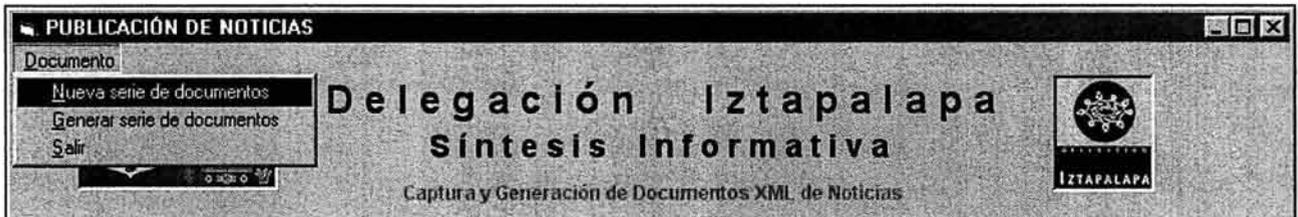
En el caso de ejecutar la acción antes mencionada para cerrar la pantalla de Publicación de noticias y se hayan capturado previamente datos sin haber generado la serie de documentos correspondiente, el

programa muestra un mensaje para confirmar al usuario si en realidad se desea cerrar la pantalla de captura y no tomar en cuenta los datos registrados.

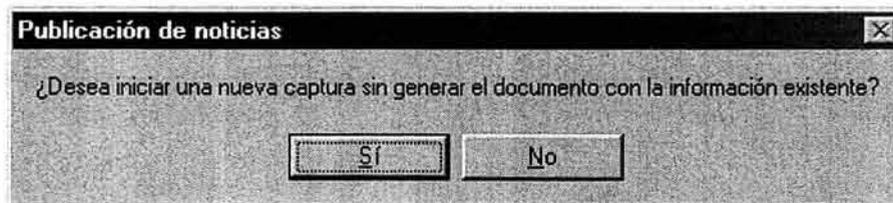


Si se capturaron noticias y por error se ejecutó la acción de salir de la ventana, se debe seleccionar la opción del botón "No", para poder así continuar con la captura o generar la serie de documentos de la forma descrita anteriormente. Si por el contrario, los datos capturados son erróneos o se prefiera abandonar la captura sin tomar en consideración los datos registrados, se debe seleccionar la opción del botón "Sí" y con esa acción se cerrará la ventana de Publicación de noticias y se mostrará el menú principal sin generar ningún documento ni ninguna actualización.

En cualquier momento de la captura de noticias, se puede reiniciar el procedimiento desde cero, es decir, borrar todo el contenido de noticias de los periódicos, de forma tal que los datos capturados se eliminen, que no se genere ningún documento ni se realice ninguna actualización. Esto se logra con la opción Nueva serie de documentos del Menú Documento.



Esta opción es recomendable para el caso de que se deseen capturar dos o más días de publicación de noticias al mismo tiempo. Siendo este el caso, se procede primero a capturar las noticias de un día específico, después se genera la serie de documentos del día capturado, se verifica la salida en el Site con ayuda del navegador y si todo es correcto, se deben entonces reiniciar los datos desde cero con la opción Nueva serie de documentos del Menú Documento. El sistema verifica que se haya generado la serie de documentos del día ya capturado. En el caso de que no se hayan generado los documentos, se muestra el siguiente mensaje de confirmación.

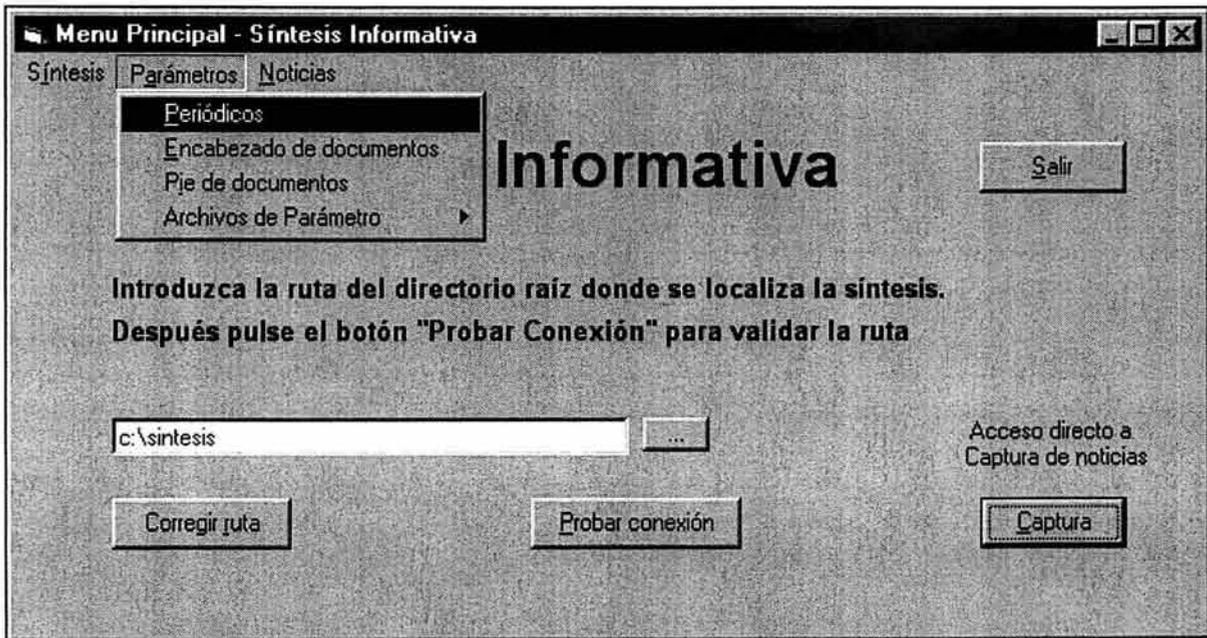


Si se capturaron noticias y por error se olvidó generar los documentos del día capturado, se debe seleccionar la opción del botón "No", para poder así continuar con la captura y generar la serie de documentos. Si por el contrario, los datos capturados son erróneos o se prefiera iniciar una nueva captura sin tomar en consideración los datos registrados, se debe seleccionar la opción del botón "Sí" con lo cual se reiniciarán los datos desde cero, es decir, sin ninguna información.

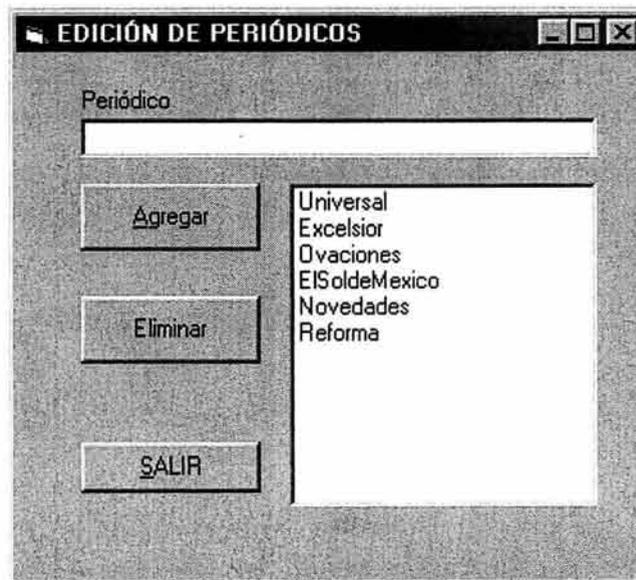
Se sabe que el total de diarios de circulación en la ciudad de México es una variable, la cual puede cambiar en algún momento. Pueden crearse nuevos diarios, o por el contrario, algunos pueden quedar fuera de circulación o cambiar de nombre. Igualmente, pueden cambiar los logotipos de la Delegación, o las leyendas que esta requiera. Todo lo anterior se describe como parámetros del sistema. La aplicación tiene igualmente la opción de agregar y eliminar periódicos para los cuales se requiere publicación dentro de la

síntesis y también la opción de cambiar el encabezado y pie de los documentos XML a generar. (Imágenes y leyendas) por medio del Menú principal.

Para establecer los periódicos que se van a considerar para publicar noticias, se debe utilizar el Menú Parámetros y seleccionar la opción Periódicos.



Al realizar la acción anterior, se abre el cuadro de diálogo de Edición de Periódicos siguiente:



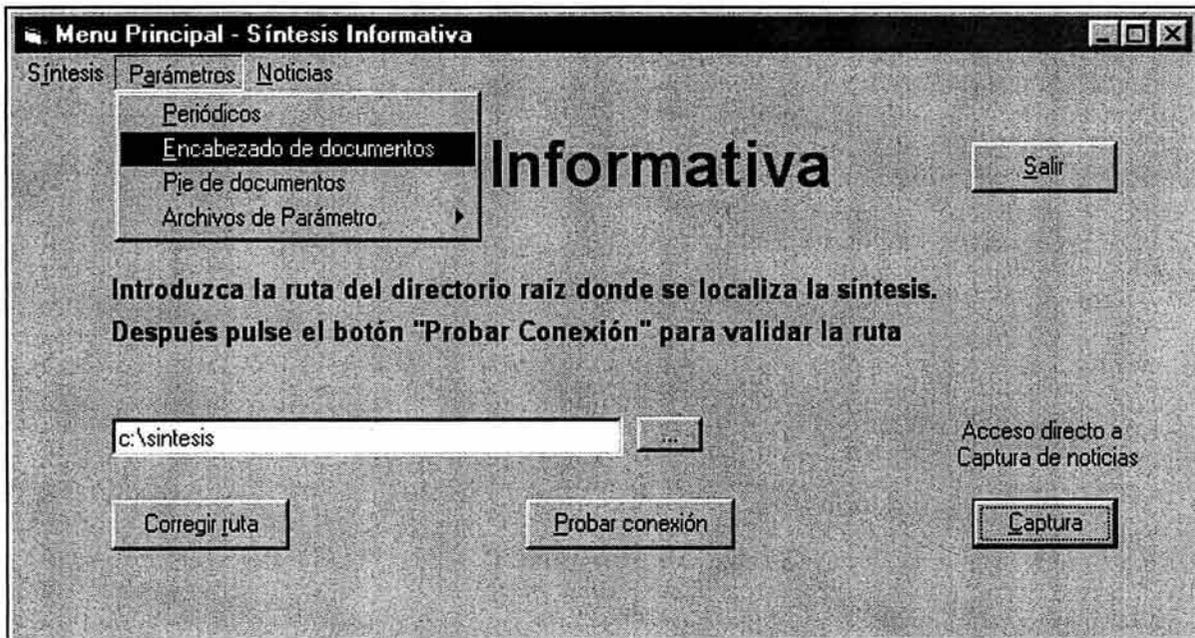
El cuadro de diálogo de Edición de Periódicos está relacionado con el archivo "paramet.xml", con el cual se pueden agregar periódicos al archivo de parámetros o eliminar algún diario que quede fuera de circulación.

Para agregar nuevos periódicos, se debe escribir el nombre del Periódico en el espacio en blanco ubicado justo debajo de la palabra Periódico, después se debe utilizar el botón Agregar o el método abreviado <Alt><A>, con lo cual se verá agregado en el cuadro de la derecha el nombre del periódico escrito.

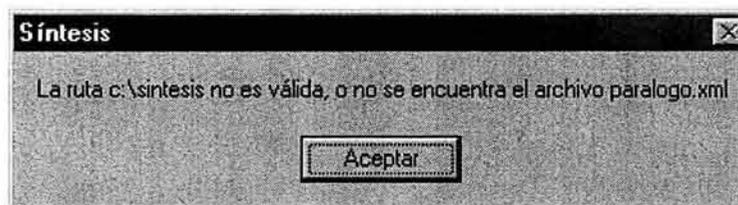
El cuadro de la derecha muestra cuales son los nombres de periódico, que actualmente están registrados. Si se desea excluir a alguno de ellos, se puede lograr por medio del botón Eliminar o el método abreviado <Alt><E>, acción con la cual se verá como desaparece el nombre del periódico de la lista.

Si no se desean realizar modificaciones a los datos que aparecen en la lista, se puede cerrar la ventana de Edición de periódicos con el botón Cerrar, o con el botón de cruz de la barra superior. Si el archivo "paramet.xml" no estaba creado, al cerrar la ventana se crea por primera vez. Si el archivo ya existía, simplemente se modifica su contenido.

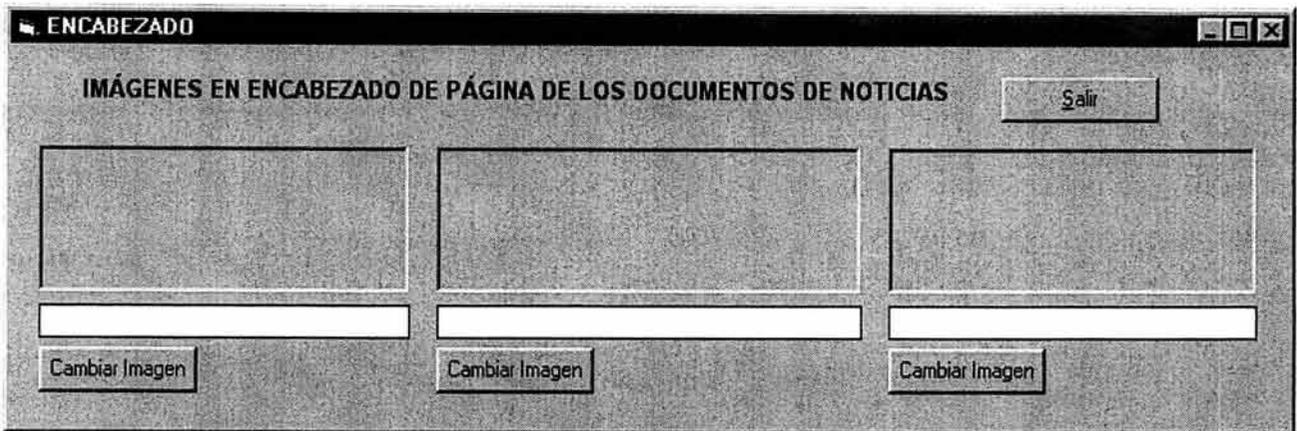
Para establecer las imágenes que se van a considerar para publicar noticias, específicamente en el encabezado de los documentos, se debe utilizar el Menú Parámetros y seleccionar la opción Encabezado de documentos.



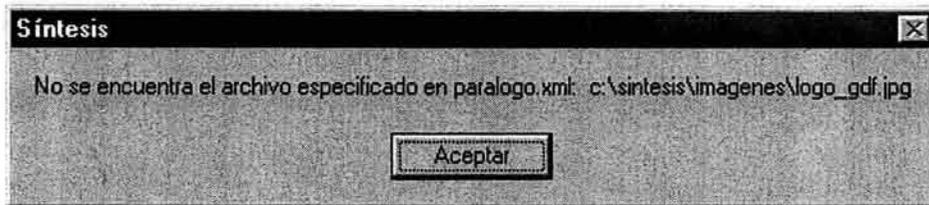
Si el archivo "paralogo.xml" no se ha creado previamente o simplemente no existe, al abrir el cuadro de diálogo de Encabezado se mostrará el siguiente mensaje de advertencia, señalando que no se localizó el archivo de parámetros mencionado y por lo tanto no hay archivos de imagen especificados para los documentos



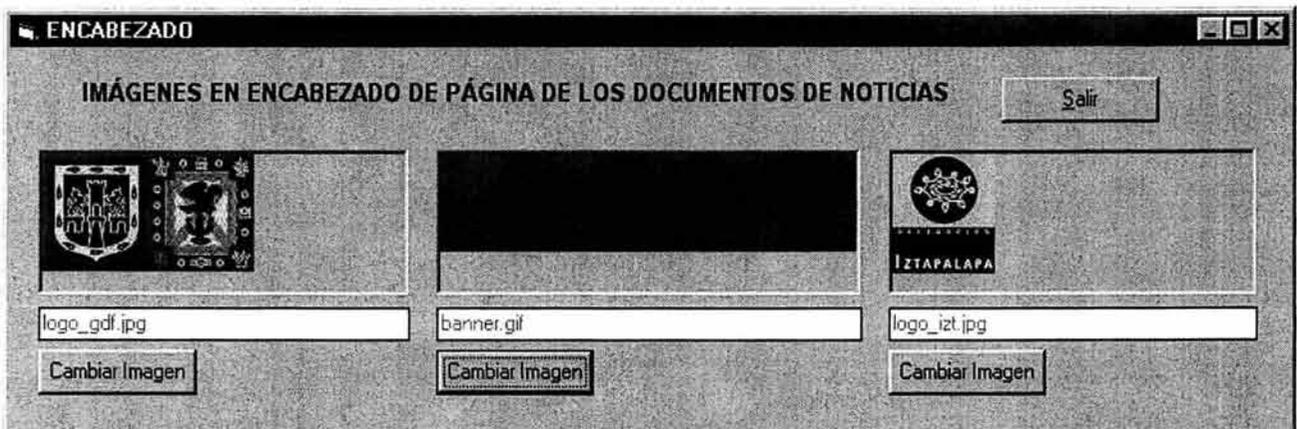
Si no se han establecido cuáles son las imágenes y logotipos de encabezado, se muestran vacíos los espacios de las imágenes. Esto puede ser, debido a que el archivo "paralogo.xml" no se localice dentro del directorio *historico* de la Síntesis o que los nombres de archivo de imagen no estén declarados dentro del documento "paralogo.xml".



De igual forma al abrir el cuadro de diálogo de encabezado, se buscan los archivos de imagen declarados dentro del archivo "paralogo.xml", para mostrarlos en la pantalla. Si alguno de los archivos especificados no se localiza dentro del directorio de imágenes de la Síntesis, se envía un mensaje de advertencia indicando cual fue el archivo que no pudo localizar.

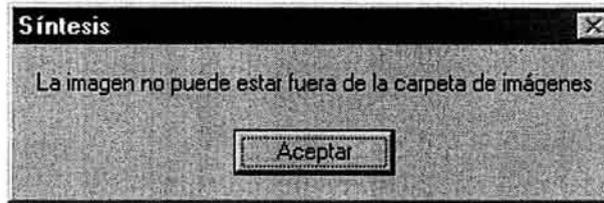


De cualquier forma, si no existe el archivo de parámetros "paralogo.xml" o no están especificados los nombres de archivo que se utilizarán para los documentos a crear, se abre el cuadro de diálogo de Encabezado. El cuadro de diálogo de Encabezado está relacionado con el archivo "paralogo.xml", en el cual se establecen cuáles son las imágenes o logotipos que se van a considerar como parte del encabezado de los documentos a generar. Se tienen considerados tres archivos de imagen distintos, el primero localizado del lado izquierdo, se prevé para el logotipo del Gobierno del Distrito Federal, la imagen del centro, se considera para el emblema o banner de la Síntesis y la imagen de la derecha, para el logotipo de la Delegación Iztapalapa.



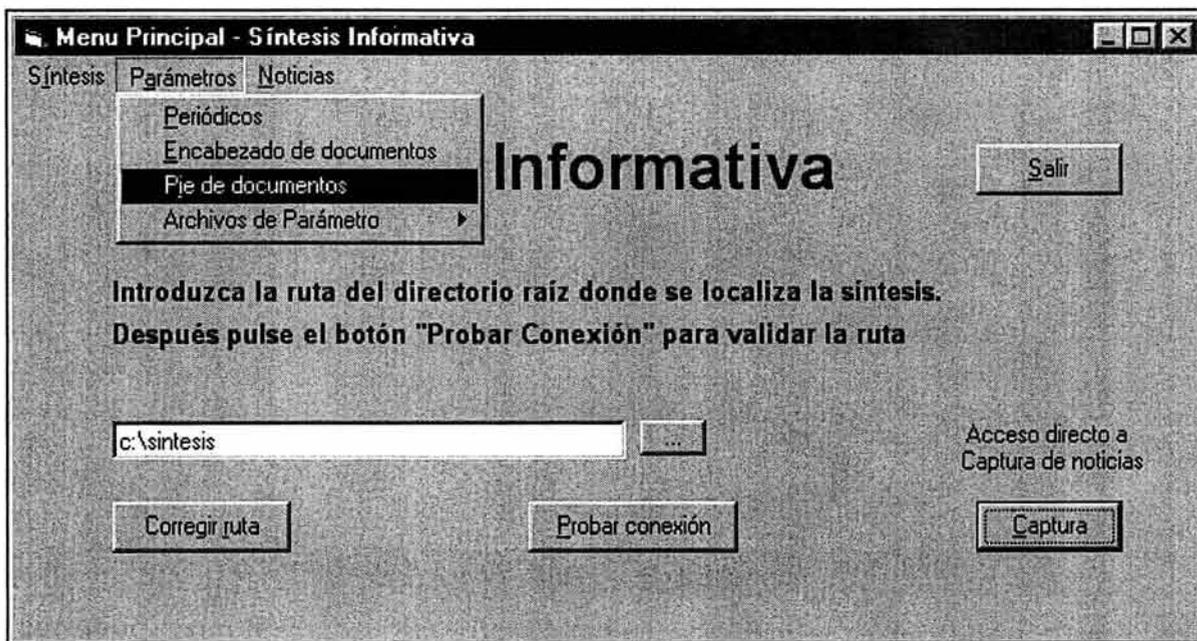
Para modificar el archivo de imagen que se muestra tanto en imagen dentro del cuadro grande como en texto, dentro del cuadro blanco debajo de la imagen, se debe utilizar el botón "Cambiar Imagen" correspondiente ubicado debajo de la imagen a modificar. Es importante señalar que cualquiera que sea la

imagen a modificar, se debe ubicar dentro del directorio de imágenes dentro de la síntesis. Si se selecciona una imagen localizada en una carpeta diferente a la de *imagenes*, el sistema mostrará un mensaje de error.



Si no se desean realizar modificaciones a las imágenes que aparecen en pantalla, se puede cerrar la ventana de Encabezado con el botón Salir, o con el botón de cruz de la barra superior. Si el archivo "paralogo.xml" no estaba creado, al cerrar la ventana se crea por primera vez. Si el archivo ya existía, simplemente se modifica su contenido.

Para establecer las imágenes que se van a considerar en el pie de los documentos, se debe utilizar el Menú Parámetros y seleccionar la opción Pie de documentos.



De la misma forma que la opción Encabezado de documentos, Si el archivo "paralogo.xml" no se ha creado previamente o simplemente no existe, al abrir el cuadro de diálogo de Pie de documento, se mostrará un mensaje de advertencia, señalando que no se localizó el archivo de parámetros mencionado y por lo tanto no hay archivos de imagen especificados para los documentos. El cuadro de diálogo de Pie de documento está relacionado con el archivo "paralogo.xml", en el cual se establecen cuáles son las imágenes o logotipos que se van a considerar como parte del pie de los documentos a generar.

Se tienen considerados dos archivos de imagen distintos, el primero localizado del lado izquierdo, se prevé para el logotipo del Site de la síntesis y la imagen de la derecha, para el logotipo de la Síntesis Informativa. Se considera también un espacio para introducir la Leyenda o Slogan del Gobierno del Distrito Federal en la parte inferior de la ventana.

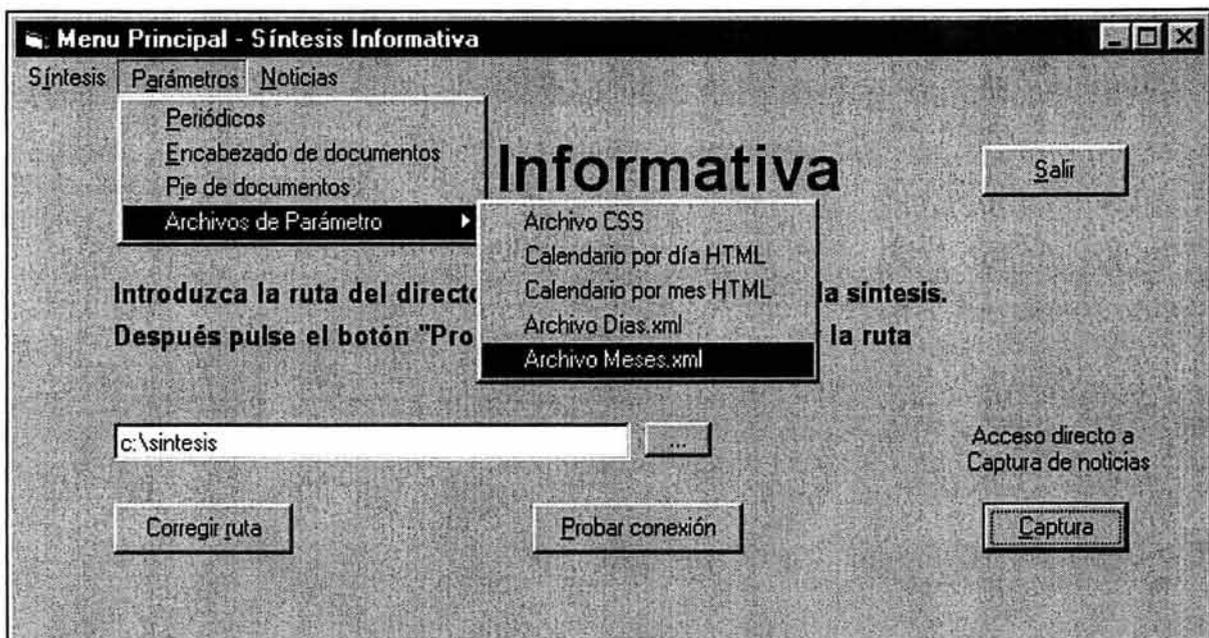


Para modificar el archivo de imagen que se muestra tanto en imagen dentro del cuadro grande como en texto, dentro del cuadro blanco debajo de la imagen, se debe utilizar el botón "Cambiar Imagen" correspondiente ubicado justo a un lado de la imagen a modificar.

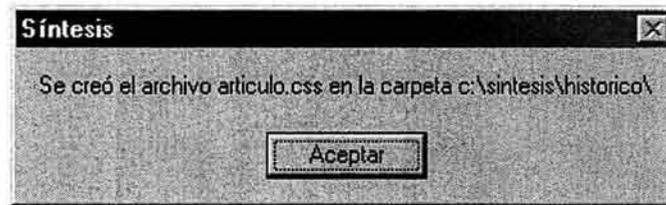
De igual forma que con el Encabezado, cualquiera que sea la imagen a modificar, se debe ubicar dentro del directorio de imágenes dentro de la síntesis. Si se selecciona una imagen localizada en una carpeta diferente a la de *imagenes*, el sistema mostrará un mensaje de error.

Si se realizaron las modificaciones necesarias a las imágenes que aparecen en pantalla, se puede cerrar la ventana de Pie de documento con el botón Salir, o con el botón de cruz de la barra superior. Si el archivo "paralogo.xml" no estaba creado, al cerrar la ventana se crea por primera vez. Si el archivo ya existía, simplemente se modifica su contenido.

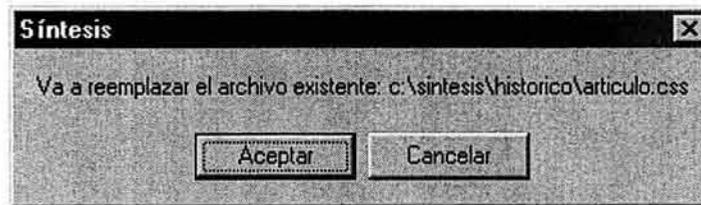
Para que el Site de la síntesis pueda ser mostrado correctamente, se necesitan las páginas que sirven de interfase específicamente el calendario de noticias, así como el documento de Hoja de estilo en cascada. Si estos archivos no existen dentro de la Síntesis, se pueden crear en cualquier momento por medio del menú principal en la dentro del menú Parámetros y la opción Archivos de Parámetro, mostrando un submenú en el cual se debe seleccionar el archivo de parámetro que se quiere crear.



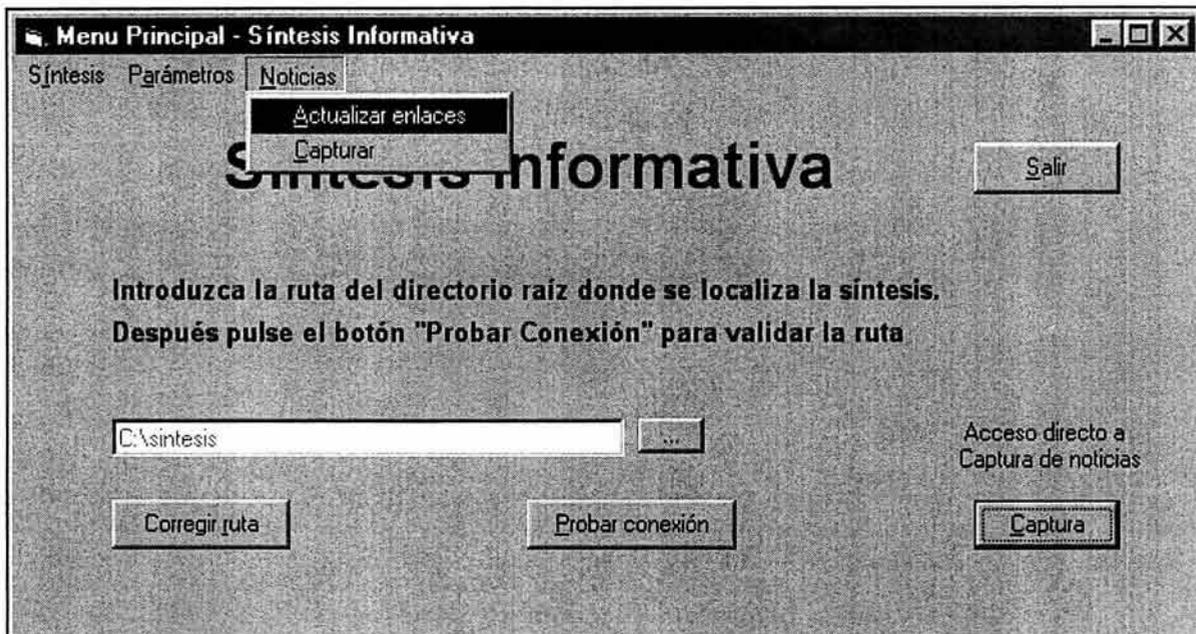
Al seleccionar alguno de los archivos de parámetro del submenú, automáticamente el sistema procede a la creación del archivo especificado, generando el documento dentro de la carpeta *historico* de la síntesis, confirmado la acción realizada con el siguiente mensaje.



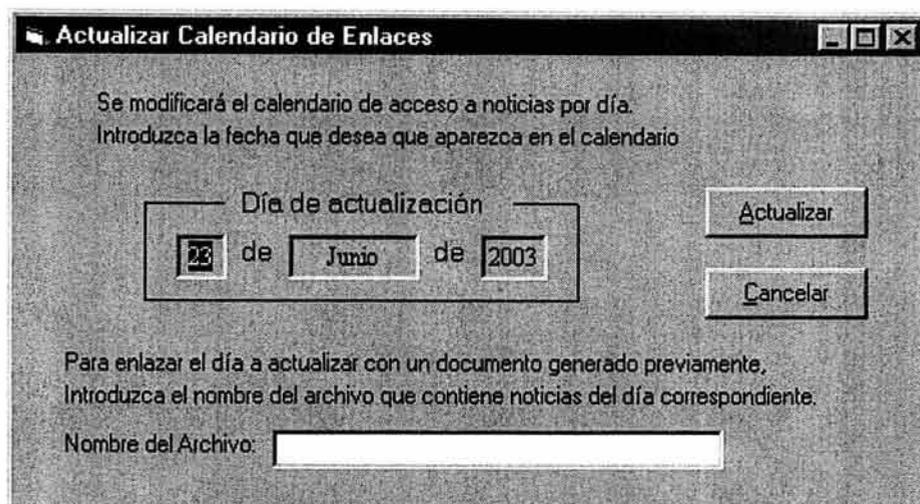
Si se diera el caso de que el archivo a generar ya exista dentro del directorio *historico* de la síntesis, se confirma la creación del archivo que reemplazará al ya existente, mostrando el siguiente mensaje de confirmación. Donde se deberá señalar si en realidad se quiere regenerar el archivo o cancelar la acción.



El programa ofrece la posibilidad de actualizar el calendario de enlace hacia las noticias por día, sin que sea necesario realizar alguna captura. Esta opción sirve para actualizar la página del calendario de noticias dentro del Site de la síntesis conforme van pasando los días del calendario. Para realizar esta actualización, se utiliza el menú Noticias y la opción Actualizar enlaces.

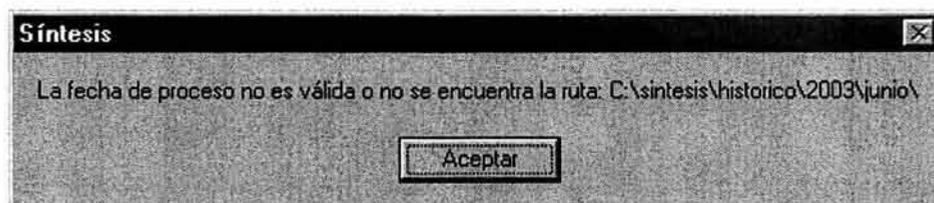


Al realizar la acción, se muestra la ventana Actualizar Calendario de enlaces. En la cual se debe especificar dentro de los campos del recuadro "Día de actualización", la fecha que se desea actualizar y en el campo en blanco "Nombre del Archivo" se debe escribir el nombre del documento XML de noticias al que se va a enlazar el día de la fecha especificado.

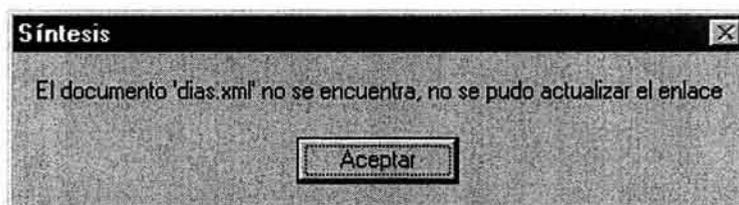


Si no existen noticias que publicar en el día de la fecha a actualizar, se debe dejar el campo "Nombre del Archivo" en blanco, para que solamente se actualice el día dentro del calendario, pero sin establecer ningún enlace.

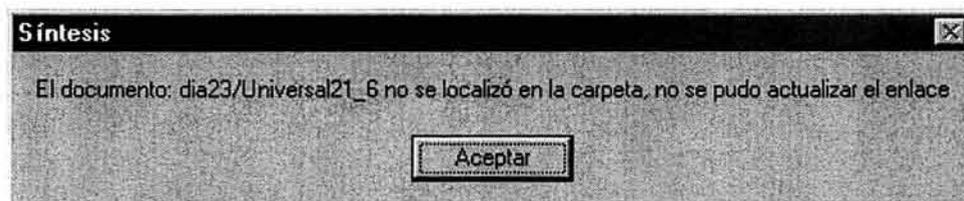
El sistema verifica que se haya creado la carpeta correspondiente del mes conforme la fecha establecida y que dentro de dicha carpeta, se localice el archivo "días.xml" que es el que se va a actualizar. Si no se localiza la carpeta, se muestra el error siguiente:



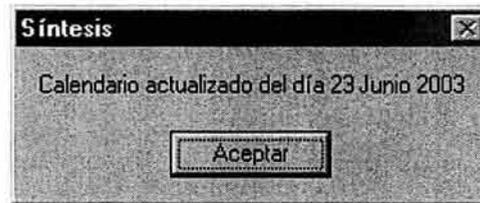
Si la carpeta del mes existe pero no se localiza el archivo "días.xml" se muestra el mensaje siguiente:



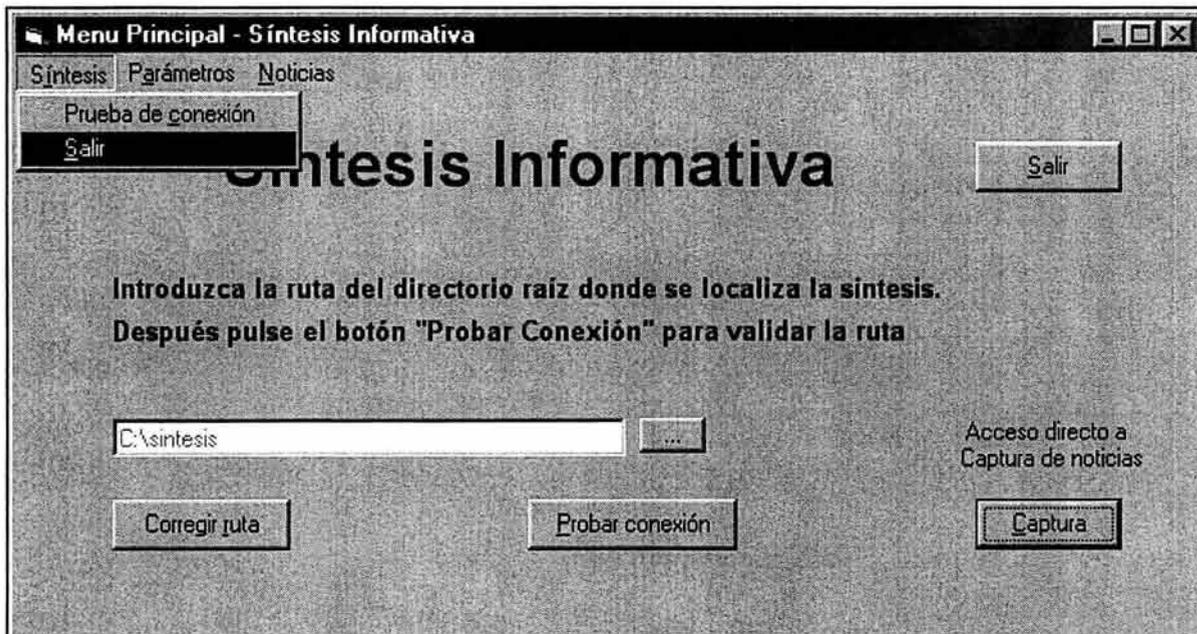
Al realizar la actualización con enlace hacia un documento de noticias XML, el sistema verifica que el archivo especificado se localice en el directorio del día correspondiente, si el archivo de noticias no se encuentra dentro de la carpeta del día, o la carpeta correspondiente no existe, se muestra un mensaje de error como el siguiente, cerrando la ventana de actualización y regresando el sistema a la pantalla inicial:



De igual forma, si no existen problemas con el enlace del documento o la estructura de los directorios del mes y día especificados, se confirma la acción de actualización del calendario, con el siguiente mensaje, mostrando la fecha que se actualizó:



Finalmente, para abandonar el sistema y finalizar la sesión, se puede cerrar la aplicación de tres formas, por medio del menú Síntesis y la opción Salir, con ayuda del botón Salir o el método abreviado <Alt> <S> o con el botón de cruz ubicado del lado derecho en la barra superior de la ventana.



III.5.3. Descripción del resultado

La aplicación tiene la ventaja, de que evita trabajar directamente con código de Lenguaje de Marcado; además, los documentos generados, cumplen con la norma XML de tener Buena Formación. Al mismo tiempo, no es indispensable tener una DTD para el documento, debido a que la aplicación genera solamente un tipo de documento con una estructura única, elementos ya definidos y en orden específico.

El contenido del documento XML de noticias generado por la aplicación (Texto plano con extensión .XML) es el siguiente:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet href="../../../articulo.css" type="text/css" ?>
<PUBLICACION xmlns:html="uri:html">
  <ENCABEZADO>
    <LOGOS>
      <LOGO_GDF><html:img SRC="../../../imagenes/logo_gdf.jpg" height="67" width="124"> </html:img>
    </LOGO_GDF>
      <EMBLEMA><html:img SRC="../../../imagenes/emblema.gif" height="55" width="252"> </html:img>
    </EMBLEMA>
      <LOGO_DELEGACION><html:img SRC="../../../imagenes/logo_izt.jpg" height="67" width="58"> </html:img>
    </LOGO_DELEGACION>
    <LOGOS>
      <SIGUIENTE><html:a href="Heraldo.xml"> SIGUIENTE </html:a> </SIGUIENTE>
      <ANTERIOR><html:a href="javascript:history.back()"> ANTERIOR </html:a> </ANTERIOR>
      <FECHA> 5 Junio 2003 </FECHA>
      <PERIODICO> Reforma</PERIODICO>
    </ENCABEZADO>

    <NOTICIA>
      <TITULO>Minimizan la manipulacion de colores</TITULO>
      <IMAGEN><html:img src="reforma_3.jpg"></html:img></IMAGEN>
      <CONTENIDO> PAN Y PRD Minimizan la manipulacion de colores </CONTENIDO>
      <PAGINA>Página: 15</PAGINA>
      <SECCION>Seccion: Ciudad</SECCION>
    </NOTICIA>

    <NOTICIA>
      <TITULO> ... </TITULO>
      <IMAGEN> ... </IMAGEN>
      <CONTENIDO> </CONTENIDO>
      <PAGINA> ... </PAGINA>
      <SECCION> ... </SECCION>
    </NOTICIA>

    <NOTICIA>
      <TITULO> ... </TITULO>
      <IMAGEN> ... </IMAGEN>
      <CONTENIDO> </CONTENIDO>
      <PAGINA> ... </PAGINA>
      <SECCION> ... </SECCION>
    </NOTICIA>

    <PIE>
      <LOGO_INTRANET> <html:a href="../../../dias.html"><html:img src="../../../imagenes/intranet.gif" alt="Calendario"
border="0"></html:img></html:a> </LOGO_INTRANET>
      <LOGO_SISTESIS> <html:a href="../../../index.html"><html:img src="../../../imagenes/sintesis.jpg" width="125"
height="71" alt="Menú Sintesis Informativa" border="0"></html:img></html:a> </LOGO_SISTESIS>
      <LEMA> Con hechos se demuestra un buen gobierno <html:img src="../../../imagenes/linea.jpg" width="480"
height="4" align="Baseline"></html:img> </LEMA>
    </PIE>
  </PUBLICACION>
```

Tomando como referencia el esquema 3.9. se tiene el cuerpo base del documento XML siguiente:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet href="../../../articulo.css" type="text/css" ?>
<PUBLICACION xmlns:html="uri:html">
  <ENCABEZADO>
  <NOTICIA>
  <NOTICIA>
  <PIE>
</PUBLICACION>
```

Los primeros dos renglones, son el encabezado del documento XML como tal.

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

Estas dos líneas son requeridas e indican, el tipo de documento junto con la versión de XML que se está utilizando y la hoja de estilos CSS a la que hace referencia para mostrar al documento con formato en un navegador.

Después, se encuentra la etiqueta raíz del contenido del documento, <PUBLICACION>, a la cual se requiere agregar el atributo: *xmlns:html="uri:html"*, con el fin de indicar que el documento XML hará referencia a etiquetas predeterminadas de HTML.

```
<PUBLICACION xmlns:html="uri:html">
```

Enseguida, se ubican los elementos <ENCABEZADO>, <NOTICIA> y <PIE> que como se muestra en el esquema 3.9. son elementos que conforman el contenido del documento y que se refieren al encabezado de noticias, las noticias publicadas y el pie del documento del periódico respectivamente. A continuación se explican estos 3 elementos dentro del documento XML generado.

1.- La etiqueta o sección **ENCABEZADO** contiene lo que es la parte superior que se visualiza del documento XML como salida en un navegador y que contiene información general respecto a los datos de la o las noticias que se estén publicando.

```
<ENCABEZADO>
  <LOGOS>
    <LOGO_GDF><html:img SRC="../../../imagenes/logo_new.jpg" height="67" width="124"> </html:img>
  </LOGO_GDF>
    <EMBLEMA><html:img SRC="../../../imagenes/banner_1.gif" height="55" width="252"> </html:img>
  </EMBLEMA>
    <LOGO_DELEGACION><html:img SRC="../../../imagenes/logo_izt.jpg" height="67" width="58">
  </html:img> </LOGO_DELEGACION>
  </LOGOS>
  <SIGUIENTE><html:a href="../dias.html"> SIGUIENTE </html:a> </SIGUIENTE>
  <ANTERIOR><html:a href="javascript:history.back()"> ANTERIOR </html:a> </ANTERIOR>
  <FECHA> 5 Junio 2003 </FECHA>
  <PERIODICO> Elsoldemexico</PERIODICO>
</ENCABEZADO>
```

La sección entre las etiquetas <logos> y </logos>, contiene tres elementos, que son los logotipos del gobierno de la ciudad de México, el emblema o banner de la síntesis y el logotipo de la Delegación Iztapalapa en forma de enlaces en código HTML.

La sección entre las etiquetas <siguiente> y </siguiente> contiene el enlace al documento consecutivo que publica noticias.

La sección entre las etiquetas <anterior> y </anterior> contiene el enlace al documento anterior que hizo el enlace al documento actual.

La sección entre las etiquetas <fecha> y </fecha> contiene información de la fecha de publicación de noticia.

La sección entre las etiquetas <periodico> y </periodico> contiene el nombre del periódico que publica las noticias en el documento.

2.- La etiqueta o sección **NOTICIA** contiene básicamente los datos registrados por la aplicación con el formulario: TITULO, IMAGEN, CONTENIDO, PAGINA y SECCION por noticia. La cantidad de estas secciones se crearán conforme sea el número de noticias registradas por cada periódico (en cada documento).

```
<NOTICIA>
  <TITULO>Seguirá la PFP en el D.F. hasta fin de mes</TITULO>
  <IMAGEN><html:img src="Heraldo_1.jpg"></html:img></IMAGEN>
  <CONTENIDO> PAN Y PRD Minimizan la manipulación de colores </CONTENIDO>
  <PAGINA>Página: 8</PAGINA>
  <SECCION>Sección: Ciudad</SECCION>
</NOTICIA>
...
<NOTICIA>
  <TITULO>...</TITULO>
  <IMAGEN>...</IMAGEN>
  <PAGINA>...</PAGINA>
  <SECCION>...</SECCION>
</NOTICIA>
```

La sección entre las etiquetas <titulo> y </titulo> muestra el título de la noticia registrada.

La sección entre las etiquetas <imagen> y </imagen> contiene el enlace en código HTML hacia la imagen, que es la noticia escaneada del periódico.

La sección entre las etiquetas <contenido> y </contenido> registra el contenido básico de la noticia capturada.

La sección entre las etiquetas <pagina> y </pagina> señala la página donde se publicó la noticia dentro del periódico.

El texto ubicado entre las etiquetas <seccion> y </seccion> indican en que sección dentro del periódico se publicó la noticia.

3.- La etiqueta o sección llamada **PIE** contiene lo que es la parte inferior que se visualiza del documento XML como salida en un navegador; contiene enlaces al menú principal de la síntesis y al calendario principal para acceder a consultas de otros días.

```
<PIE>
  <LOGO_INTRANET> <html:a href=" ../dias.html"><html:img src=" ../ ../ ../imagenes/intranet.gif"
alt="Calendario" border="0"></html:img></html:a> </LOGO_INTRANET>
  <LOGO_SISTESIS> <html:a href=" ../ ../ ../index.html"><html:img src=" ../ ../ ../imagenes/sintesis.jpg"
width="125" height="71" alt="Menú Sintesis Informativa" border="0"></html:img></html:a>
</LOGO_SISTESIS>
  <LEMA> Con hechos se demuestra un buen gobierno <html:img src=" ../ ../ ../imagenes/linea.jpg"
width="480" height="4" align="Baseline"></html:img> </LEMA>
</PIE>
```

La sección entre las etiquetas <logo_intranet> y </logo_intranet> contiene el enlace al calendario por día junto con el enlace hacia la imagen del logotipo de la Intranet, ambos en código HTML.

La sección entre las etiquetas <logo_sintesis> y </logo_sintesis> contiene el enlace hacia el menú principal de la Síntesis junto con la imagen del logotipo de la síntesis, ambos también en código HTML.

El texto entre las etiquetas <lema> y </lema> presentan el lema del gobierno del Distrito Federal.

La vista en el navegador (Internet Explorer 5.0) del documento generado es la siguiente



esquema 3.18. Vista de la generación de una página por medio de la aplicación.

Como se puede ver en el esquema de la página generada con la aplicación, se muestran los 3 elementos básicos: Encabezado, Noticia y Pie.

Puntualizando los elementos dentro de la página y tomando como referencia la siguiente lista, se localizan fácilmente los elementos del encabezado y pie:

Encabezado

1. logos
 - 1-1. logo_gdf
 - 1-2. logo_emblema
 - 1-3. logo_delegacion
2. anterior
3. siguiente
4. fecha
5. periódico.

Encabezado

6. logos_pie
 - 6-1 logo_síntesis
 - 6-2 logo_intranet
7. lema

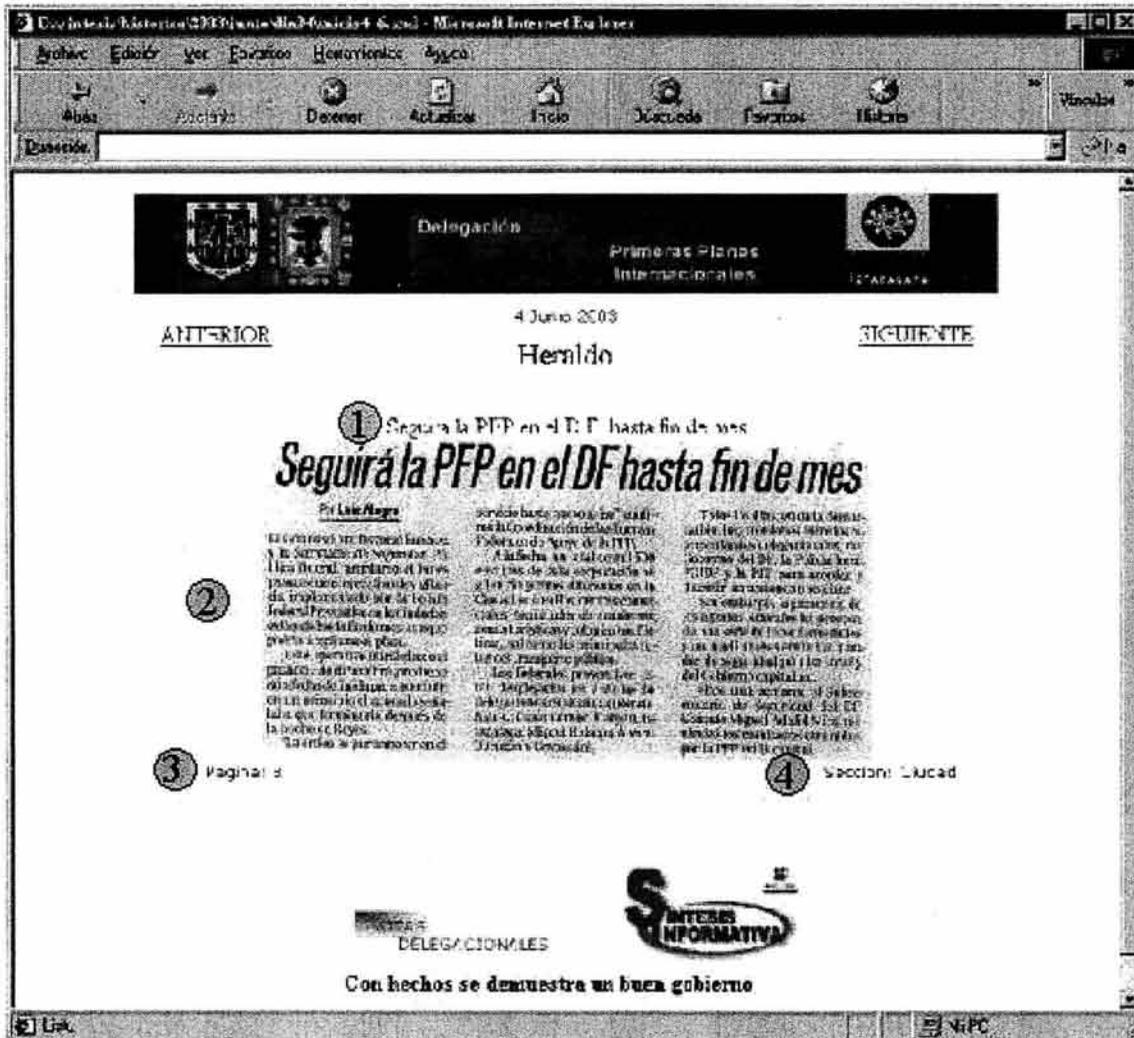


esquema 3.19. Identificación de elementos dentro de la página resultante (Encabezado y Pie).

En la parte central se localiza el contenido de cada noticia, los elementos que la componen son:

Noticia

1. título
2. imagen
3. página
4. sección
5. contenido



esquema 3.20. Identificación de elementos dentro de la página resultante (Noticia).

Pero como se exhibe en el esquema, la sección de *contenido* no se localiza. Dentro del formulario del programa para generar la página, se prevé capturar cierto contenido básico referente al tema o temas que trata la noticia, o en su defecto la noticia completa. Esta sección es realmente representada por el texto de la noticia como tal, es decir, la imagen escaneada de la noticia, por lo cual no se muestra dentro de la página.

Sin embargo es importante capturar el contenido de la noticia, sólo con el fin de tener un texto dentro del documento XML que indique sobre que trata la noticia. Poniendo dos puntos específicos, si se abre el documento XML en un editor de texto, sólo se muestra la imagen a la que hace referencia el documento ya que no es visible la imagen escaneada en un editor. Esto mismo ocurre para un motor de búsqueda. Si por ejemplo se desea implementar un buscador que localice noticias, buscar por su contenido regresaría un resultado nulo, debido a que no es posible obtener contenido textual de una imagen.

Como se tienen referenciados cada una de las partes que componen el documento, gracias a la marcación de las etiquetas XML, se puede manejar el contenido de los documentos generados. De esta forma, dentro de un buscador, se facilitan las búsquedas y se mejoran los resultados de búsqueda.

Con la ayuda de la hoja de estilo, se consigue ocultar la sección de *contenido* que no es visible dentro de la página.

De la misma forma la hoja de estilo "articulo.css", hace referencia a los elementos del documento XML:

- encabezado
 - logos
 - logo_gdf
 - emblema
 - logo_delegacion
 - siguiente
 - anterior
 - fecha
 - periodico
- noticia
 - imagen
 - titulo
 - pagina
 - seccion
 - contenido
- pie
 - logo_pie
 - logo_intranet
 - logo_sintesis
 - lema

El contenido del archivo **articulo.css**, es el siguiente:

```
encabezado {display:block; width:100%; position:relative; text-align:center;}
logos {display:inline; width:80%; clear:all; background-color:"#006600"; }
logo_gdf {align:left; width:30%;}
emblema {align:center; width:45%; }
logo_delegacion {align:right; width:25%; }

siguiente {display:inline; margin-top:20px; float:right; width:35%;}
anterior {display:inline; margin-top:20px; float:left; width:35%;}
fecha {display:block; font-size:8pt; color:green; font-family:Verdana; margin-top:10px; }
periodico {display:block; font-size:16pt; color:black; font-family:times; margin-top:10px;}

noticia {display:block; width:100%; position:relative; text-align:center; margin-top:20px; }
imagen {display:block; margin-bottom:1px; clear:all;}
titulo {display:block; font-size:12pt; color:black; font-family:times; margin-top:10px;}
pagina {width:40%; float:left; font-size:8pt; color:green; font-family:Verdana; }
seccion {width:40%; float:right; font-size:8pt; color:green; font-family:Verdana; }
contenido {display:none}

pie {display:block; width:100%; position:relative; text-align:center; margin-top:30px;}
logo_intranet {align:left; width:50%;}
logo_sintesis {align:right; width:50%; }
lema {display:inline; align:center; width:100%; color:"#006600", font-family:arial; font-weight: bold}
```

Como se comentó en el tema “Herramientas de apoyo” del capítulo II, dentro de las funcionalidades de las hojas de estilo, se pueden establecer propiedades para dar formato de salida a los documentos XML. Así mismo, el archivo de hoja de estilo **articulo.css**, establece la propiedad de no visible o no desplegar, para la etiqueta *contenido*.

```
contenido {display:none}
```

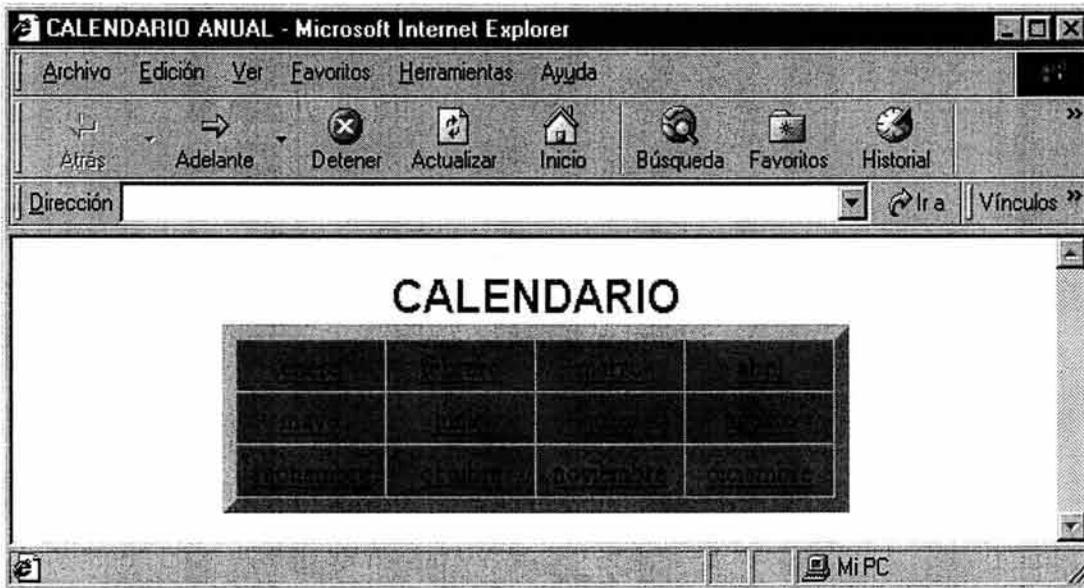
Por otra parte, como se mencionó dentro del funcionamiento de la aplicación para actualización de Noticias, se puede igualmente actualizar de forma automática los enlaces de las páginas de consulta por mes y por día, esto se logra por medio de los archivos *meses.xml* y *dias.xml*, que son archivos que trabajan conjuntamente con las páginas *meses.html* y *dias.html*, comentados en la primera parte del desarrollo de la Síntesis dentro de capítulo.

EL contenido básico del archivo *meses.xml* es el siguiente.

```
<?xml version="1.0" ?>
<meses>
  <cuatrimestre>
    <uno>
      <enlace>enero/dias.html</enlace>
      <mes>enero</mes>
    </uno>
    <dos>
      <enlace>febrero/dias.html</enlace>
      <mes>febrero</mes>
    </dos>
    <tres>
      <enlace>marzo/dias.html</enlace>
      <mes>marzo</mes>
    </tres>
    <cuatro>
      <enlace>abril/dias.html</enlace>
      <mes>abril</mes>
    </cuatro>
  </cuatrimestre>
  ...
</cuatrimestre>
<cuatrimestre>
  <uno>
    <enlace>septiembre/dias.html</enlace>
    <mes>septiembre</mes>
  </uno>
  <dos>
    <enlace>octubre/dias.html</enlace>
    <mes>octubre</mes>
  </dos>
  <tres>
    <enlace>noviembre/dias.html</enlace>
    <mes>noviembre</mes>
  </tres>
  <cuatro>
    <enlace>diciembre/dias.html</enlace>
    <mes>diciembre</mes>
  </cuatro>
</cuatrimestre>
</meses>
```

El archivo *meses.xml* contiene el nombre del mes que se desea consultar y el enlace al que debe hacer referencia dentro de la página *meses.html*. La estructura del archivo se basa en seccionar el documento por cuatrimestre en el cual dentro de cada cuatrimestre puede haber uno o ningún mes que mostrar relacionados de uno a cuatro, donde cada uno señala el nombre del mes y el enlace al que hace referencia.

La vista en el navegador (Internet Explorer 5.0) de la página de consultas por mes generada es la siguiente

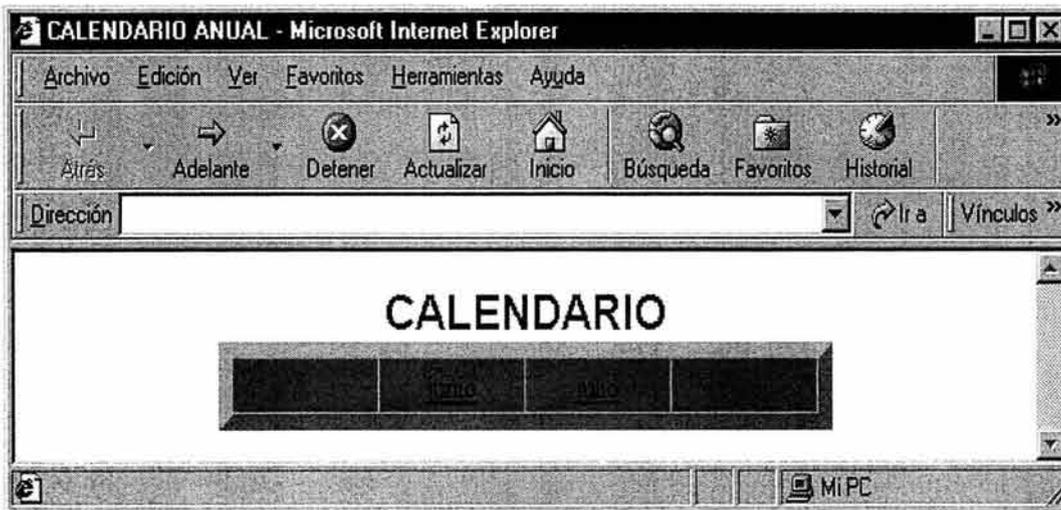


esquema 3.21. Vista de la página de consultas por mes.

Para entender el funcionamiento del documento meses.xml, supóngase que su contenido fuera el siguiente:

```
<?xml version="1.0" ?>
<meses>
  <cuatrimestre>
    <dos>
      <enlace>junio/dias.html</enlace>
      <mes>junio</mes>
    </dos>
    <tres>
      <enlace>julio/dias.html</enlace>
      <mes>julio</mes>
    </tres>
  </cuatrimestre>
</meses>
```

La vista de la página sería de la siguiente forma:



esquema 3.22. Vista de la página de consultas modificada.
EL contenido básico del archivo dias.xml es el siguiente.

```
<?xml version="1.0" ?>
<mes>
  <semana>
    <domingo>
      <enlace>dia01/periodico1.xml</enlace>
      <dia>1</dia>
    </domingo>
    <lunes>
      <enlace>dia02/periodico1.xml</enlace>
      <dia>2</dia>
    </lunes>
    <martes>
      <enlace>dia03/periodico1.xml</enlace>
      <dia>3</dia>
    </martes>
    <miercoles>
      <enlace>dia04/periodico1.xml</enlace>
      <dia>4</dia>
    </miercoles>
    <jueves>
      <enlace>dia05/periodico1.xml</enlace>
      <dia>5</dia>
    </jueves>
    <viernes>
      <enlace>dia06/periodico1.xml</enlace>
      <dia>6</dia>
    </viernes>
    <sabado>
      <enlace>dia07/periodico1.xml</enlace>
      <dia>7</dia>
    </sabado>
  </semana>
  ...
</semana>
  ...
</semana>
</mes>
```

De forma similar al archivo meses.xml, el archivo dias.xml contiene el número de día al que se desea consultar y el enlace al que debe hacer referencia dentro de la página dias.html. La estructura del archivo se basa en seccionar el documento por semana donde dentro de cada semana puede haber uno o ningún día que mostrar relacionados de domingo a sábado y en cada día, se señala el número de día y el enlace al que hace referencia.

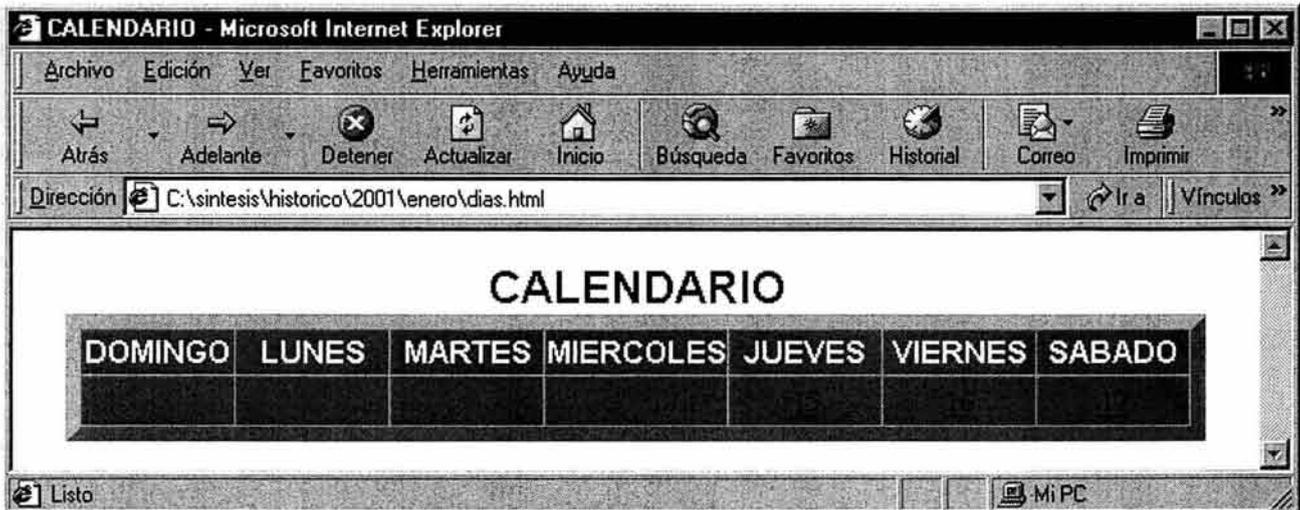
La vista en el navegador (Internet Explorer 5.0) de la página de consultas por día generada es la siguiente



esquema 3.23. Vista de la página de consultas por día.
Si el contenido del documento dias.xml fuera el siguiente:

```
<?xml version="1.0" ?>
<mes>
  <semana>
    <jueves>
      <enlace>dia15/periodico1.xml</enlace>
      <dia>15</dia>
    </jueves>
    <viernes>
      <enlace>dia16/periodico1.xml</enlace>
      <dia>16</dia>
    </viernes>
    <sabado>
      <enlace>dia17/periodico1.xml</enlace>
      <dia>17</dia>
    </sabado>
  </semana>
</mes>
```

La vista de la página sería de la siguiente forma:



esquema 3.24. Vista de la página de consultas por día modificada.

Conviene tomar de referencia el contenido de los archivos meses.html y dias.html mostrados en el capítulo III.5.1., con el fin de comprender la actualización que se lleva a cabo.

Dentro del archivo meses.html, se localiza el siguiente código:

```

...
<xml id="calendario" src="meses.xml"></xml>
<table datasrc="#calendario" >
<tbody>
<tr>
<th>
<table datasrc="#calendario" datafld="uno">
<tr><td> <a href=<span datafld="enlace"></span><span datafld="mes"></span></a>
</td></tr>
</table>
</th>
</tr>
</tbody>
</table>
...

```

La línea: `<xml id="calendario" src="meses.xml"></xml>` indica al navegador, que se tome de referencia el archivo `meses.xml` y que se identifique con la palabra `calendario` para trabajar con su contenido dentro de la página `meses.html`.

En la línea `<table datasrc="#calendario" datafld="uno">`, se indica que se van a acceder a datos contenidos en el archivo `meses.xml` a través del identificador `calendario` y que se hará uso de los valores contenidos entre las etiquetas `<uno>` y `</uno>` del archivo de referido `meses.xml`.

Dentro del código `<a href=`, se pueden identificar dos partes de datos: `` y ``. Con ayuda de la instrucción `<a href= ... `, se consigue realizar el vínculo deseado por el documento `meses.xml`, donde ``, toma como referencia el contenido entre las etiquetas `<enlace>` y `</enlace>` de la sección entre las etiquetas `<uno>` y `</uno>`, así mismo, ``, muestra el contenido entre las etiquetas `<mes>` y `</mes>` de la misma sección. Donde finalmente, "enlace" indica al navegador el enlace que debe vincular y "mes" indica al usuario el mes al que dicho enlace hace referencia.

De igual forma, se trabaja con el contenido de las secciones `<dos> </dos>`, `<tres> <tres>` y `<cuatro> <cuatro>`, simplemente cambia el texto "uno", por el texto correspondiente dentro del código ya explicado `<table datasrc="#calendario" datafld="uno">`. El resultado es siempre, mostrar una tabla de cuatro columnas, diseñado así, para facilitar la consulta al usuario.

De manera similar trabaja el archivo `dias.html`. Dentro de dicho archivo, se localiza el siguiente código:

```

<xml id="calendario" src="dias.xml"></xml>
<table datasrc="#calendario" cols="7">
...
<table datasrc="#calendario" datafld="domingo">
<tr><td> <a href=<span datafld="enlace"></span><span datafld="dia"></span></a>
</td></tr></table>
...

```

La línea: `<xml id="calendario" src="dias.xml"></xml>` indica al navegador, que se tome de referencia el archivo `dias.xml` y que se identifique con la palabra `calendario`, igual que en el código de `meses.html`

En la línea `<table datasrc="#calendario" datafld="domingo">`, se indica que se va a acceder a datos del archivo `dias.xml` y que se hará uso de los valores contenidos entre las etiquetas `<domingo>` y `</domingo>` conforme el archivo de referido con el identificador `calendario`.

Dentro del código `<a href=`, se pueden identificar igualmente dos partes de datos: `` y ``; donde ``, toma como referencia el contenido entre las etiquetas `<enlace>` y `</enlace>` dentro de la sección `<domingo> ... </domingo>`, así mismo, ``, muestra el contenido entre las etiquetas `<dia>` y `</dia>` de la misma sección. Donde finalmente, "enlace" indica al navegador el enlace que debe vincular y "día" indica al usuario el día al que dicho enlace hace referencia.

Las secciones `<lunes> </lunes>`, `<martes> </martes>`, `<miercoles> </miercoles>`, `<jueves> </jueves>`, `<viernes> </viernes>` y `<sabado> </sabado>`, trabajan de igual forma, simplemente cambia el texto "domingo", por el texto correspondiente dentro del código `<table datasrc="#calendario" datafld="domingo">`.

El resultado es obtener una tabla de 7 columnas, tal como si se estuviera trabajando con un calendario, donde la primera columna corresponde siempre al día domingo y así sucesivamente hasta la columna siete, que corresponde al día sábado. Cada sección `<semana></semana>`, indica un renglón dentro de la tabla.

El planteamiento es principalmente, que en los nuevos proyectos que se desarrollen para la Web o la Intranet de la Delegación, se contemple la utilización de XML para su desarrollo.

Las ventajas obtenidas con el uso de XML son:

- XML es un lenguaje libre y no requiere ningún tipo de licencia para su utilización.
- Tiene manejo de etiquetas propias que ayudan a reconocer la lógica del documento.
- Debido al marcado libre, es posible que se pueda crear un motor de búsqueda para noticias en el Site (con ayuda de programas tales como CGI, ASP, PHP, etc.)

Algunas desventajas que se tienen:

- XML no tiene compatibilidad con todos los Navegadores para Internet

Por último, cabe citar lo que según el consorcio de la Web (W3C), establece como los cuatro tipos de aplicaciones que impulsarán el desarrollo del XML:

- Aplicaciones que exijan que el cliente Web interactúe entre dos o más bases de datos
- Aplicaciones que intentan transferir una parte significativa de la carga del proceso del servidor al cliente Web.
- Aplicaciones que precisen que el cliente Web presente diferentes versiones de los mismos datos a diferentes usuarios.
- Aplicaciones en las que agentes Web inteligentes intenten adaptar la búsqueda de información a las necesidades de los usuarios individuales

CONCLUSIONES

El Lenguaje de Marcado eXtensible (XML) implementa en la realidad, la idea arraigada de que los datos deben ser intercambiados en la forma de documentos como tal: catálogos, recibos, ordenes de compra, contratos, entre otros; donde se especifican claramente los datos requeridos y brindados. De esa manera es más fácil y seguro entender la información y al mismo tiempo agilizar el procesamiento de datos a diferentes áreas.

Por la misma razón, XML es ideal para negocios que están familiarizados con el intercambio de documentos. Los usuarios finales pueden entender el concepto de documentos más sencillo de lo que pueden entender estructuras de datos abstractas. Las compañías y los individuos pueden coincidir en formatos comunes de documentos, mucho más fácilmente que en la selección de sistemas.

Con XML, un desarrollador consume, crea o modifica datos propios a partir de una estructura de datos lógica que es independiente de todo sistema. Fuentes de datos de múltiples orígenes pueden alimentar con información a un único tipo de estructura XML, permitiendo una elevada integración entre diferentes sistemas y aplicaciones. Desde un particular punto de vista, esa es la razón de peso por la cual se debe enfocar la atención para utilizar y conocer XML, debido a que XML es texto simple, estructurado y cuenta con herramientas de apoyo para su funcionamiento. Además, ésta integración, puede tomar lugar a través de la Web por medio de Internet y complementarse de manera eficiente con las transacciones que se realizan comercialmente, como el Intercambio Electrónico de Datos (EDI) y el Business to Business (B2B).

La experiencia obtenida con la Síntesis Informativa en la Delegación Iztapalapa, basando la actualización de la información con XML, es que resulta sencillo actualizar datos dentro de un documento con elementos específicos, gracias a que son identificables tanto por un programa de computación, como por los usuarios.

Por otra parte, la creación de documentos se refleja inmediatamente en el sitio Web de la Intranet, evitando, trabajar directamente con código de Lenguaje de Marcado y cumpliendo así, con el objetivo de la distribución de las noticias, evitando gastar recursos innecesarios.

La carrera de Matemáticas Aplicadas y Computación, brinda las herramientas para comprender elementos con los cuales XML está directamente relacionado, tales como: estructuras y almacenamiento de datos, bases de datos, sistemas de información y procesamiento de datos. XML es una tecnología que ofrece nuevas posibilidades para el trabajo con dichos temas, mismos que han ido creciendo aún cuando no es una tecnología que sea aceptada en el mercado de la informática con gran apertura, pero la distancia para alcanzar ese objetivo ha ido reduciéndose y con ello se esperaba que el respaldo sea total, tal como lo vivió HTML.

BIBLIOGRAFÍA

Charles F. Goldfarb, Paul Prescod. "Manual de XML". Ed. Prentice Hall. 1ª. edición, Madrid 1999.

Maruyama, Tamura, Uramoto. "Creación de Sitios Web con XML y Java". Ed. Pearson Educación, S.A. 1ª. edición, Madrid 2000.

Abraham Gutiérrez, Raúl Martínez. «XML a través de ejemplos.». Ed. Alfaomega Ra-Ma. 1ª. edición, Madrid 2001

Joachim Schwarte. "El gran libro de HTML: cómo publicar en Internet". Ed. Marcombo. 1ª. edición, Barcelona 1996.

Nicholas Chase. "Active Server Pages 3.0". Ed Prentice Hall, 1ª. edición, Madrid 2000.

DIRECCIONES DE INTERNET

<http://www.w3.org/>
World Wide Web Consortium – Consorcio de la World Wide Web.

<http://www.microsoft.com/xml>
XML developer center – Centro de desarrollo de XML.

<http://www.ibiblio.org>
The public's library and digital archive – La biblioteca y archivo digital de lo público.

<http://xml.com>
XML from the inside out (XML resources) – XML a fondo (recursos XML).

<http://xml.org>
XML and Web services standards in industry – XML y estándares de servicios Web en la industria.