

## CAPÍTULO 3. TECNOLOGÍAS DOCUMENTALES

### 3.1 Sistemas de información hipertextuales

#### 3.1.1 Antecedentes

En un principio la lectura era lineal, y esto cambió por completo entre el segundo y el cuarto siglo de la era cristiana, cuando una nueva forma de soporte se impuso a los lectores griegos y romanos. El códice suplantó el rollo de papiro. La invención del códice, de la paginación y de los índices instituía una relación inédita entre el lector y el texto, al mismo tiempo que permitía acciones imposibles con el rollo de papiro, por ejemplo, hojear o encontrar rápidamente un pasaje en particular. Esto representa la implementación de las asociaciones para encontrar dentro de un texto, lo que se buscaba de manera más sencilla, sin tener que recorrer el texto por completo. Un cambio similar ocurrió con la aparición del hipertexto y su estructura no lineal o multiseccional.

El desarrollo de este concepto comienza en 1937 con Vannevar Bush, profesor en el Massachusetts Institute of Technology (MIT), quien presenta el *Rapid Selector*, un dispositivo analógico basado en la tecnología del microfilm capaz de acelerar el archivo y búsqueda de información. Posteriormente Vannevar Bush deja el MIT para hacerse cargo del Carnegie Institute de Washington. En este período comienza a trabajar en el proyecto de una máquina analógica capaz de "extender la memoria".

Ya como consejero científico del presidente estadounidense Roosevelt, publica en 1945 el artículo "As we may think" en *The Atlantic Monthly*, en donde presenta un sistema analógico para almacenar e interconectar bases de datos, que denomina *Memex*, como una posible solución a los problemas originados por el tratamiento de grandes cantidades de información. Bush especifica algunas de las características que hoy comparten todos los sistemas hipertextuales, como son el acceso rápido a la información, la posibilidad de establecer uniones y recorridos entre los bloques de información, y la capacidad de contribución del usuario creando nuevas conexiones o contenidos.

Sin duda este artículo es el más citado en lo que a la historia del hipertexto se refiere, y esto se debe al argumento que en él sostenía. Mencionaba que hasta ese momento no existía ninguna herramienta para recuperar información de manera rápida y eficaz, solo se contaba con índices, y la cantidad de información que se generaba era demasiada para controlarla con ese sistema; por lo que planteaba el diseño de una máquina que funcionara similar a la asociación de ideas que se realiza en el cerebro humano al recuperar información.

En 1958 Bush vuelve al MIT y prepara el primer borrador de un artículo titulado "Memex II" (o "Memex Mark II") en el cual se propone actualizar el anterior Memex a las nuevas tecnologías. En realidad el Memex nunca se construyó, solo se hicieron dibujos de su posible apariencia.

La idea original de hipertexto se debe a Vannevar Bush, pero el primero en crear y publicar el término de hipertexto fue Ted Nelson.

Theodor Holme Nelson era estudiante de postgrado de la Universidad de Harvard y en 1960 comienza a trabajar en el proyecto Xanadu, un sistema informático basado en la organización no secuencial de los textos. Nelson planteó la idea de crear una red de documentos que pudiesen ser recuperados desde cualquier parte del mundo y que concentrara toda la literatura producida por la humanidad.

En 1962 Douglas C. Engelbart, trabajando en el Stanford Research Institute, publica su obra *A conceptual framework for the argumentation of man's intellect*, la cual describía un sistema de aumento en la capacidad intelectual humana. Inspirado en el Memex de Vannevar Bush, Engelbart creó el primer sistema que operaba bajo el principio de hipertexto y lo denominó oNLine System (NLS). Las investigaciones de Engelbart llevaron a la invención de instrumentos como el *mouse*, del sistema de ventanas (windows), del "help-on-line", del correo electrónico y del procesador de textos. Es hasta el año de 1965 cuando Ted Nelson utiliza por primera vez el concepto "hipertexto" para definir el sistema Xanadu.

Otro hombre importante en el desarrollo de sistemas hipertextuales como el WWW fue Licklider, quien en 1961 desarrolló investigaciones sobre las bibliotecas del futuro, donde mencionaba la necesidad de crear una red que conectara a los recursos que se relacionaran entre sí.<sup>66</sup>

A continuación se presentan los hechos más relevantes en el desarrollo de los hipertextos:

Año	Sistema	Autor	Hito
1945	MEMEX	Vannevar Bush	Dispositivo basado en microfichas
1965	XANADU	Ted Nelson	Primero en acuñar el término "hypertext"
1967	Hypertext Editing System	Andy Van Dam (Brown University)	Sistema hipertexto en funcionamiento
1968	oN Line System NLS	Doug Engelbart (Stanford University)	Sistema hipertexto con manipulación directa con utilización de ratón
1978	Aspen Movie Map	Andrew Lippman (MIT)	Primer sistema hipermedio en funcionamiento
1985	Intermedia	Yankelovich (Brown University)	Se utiliza el concepto de ancla y red

<sup>66</sup> Para más información consultar: SCOLARI, Carlos Alberto. "Definiciones". — *En Hyperpage* : <http://www.kweb.it/hyperpage/defini1.htm>. — Italia : Kaleidos, 1999. Consultado el 3 de septiembre de 2002

1986	GUIDE	OWL	Primer producto para autoría de hiperdocumentos
1987	HyperCard	Apple Computer, Inc.	Producto entregado con cada Macintosh
1987	Hypertext '87	University of North Carolina	Primera conferencia auspiciada por la ACM para el tratamiento de la tecnología hipertexto
1991	World Wide Web	Tim Berners-Lee, CERN	Proyecto para llevar la tecnología hipermedial en Internet
1993	Mosaic	NCSA	Navegador gráfico para el WWW

*Tabla I. Principales hitos en la historia de los sistemas hipertextuales.*

### 3.1.2 Definición y conceptos

Conocer la etimología siempre esclarecerá el camino para precisar un concepto. En el caso del Hipertexto, el prefijo hiper se deriva del griego “hyper” que significaba "más allá", "sobre", "encima" (Ej. hipérbole); en las palabras de aparición moderna representa una cantidad o grado superior al normal o excesivo, en ocasiones se utiliza de la misma manera el término súper (Ej. hipermercado). El sufijo texto, significa el conjunto de palabras que componen un documento o un tratado manuscrito o impreso. Del lat. textum o textus "trama", deriva de texere "tejer".

Analizando el origen de la palabra podemos entenderla como el conjunto de palabras que componen un documento de grado superior al normal y que va más allá. La característica que le da un grado superior al normal se debe a que puede integrar diversos textos e incluso imágenes, videos, etc., a través de ligas o referencias que los relacionan. El ir más allá se debe al salto que se tiene que dar para ir de un texto a otro y que permite tomar diversas rutas según el lector lo desee.

Recogiendo algunas definiciones de estudiosos en el tema podemos rescatar las siguientes:

Antonio De las Heras precisa *"el texto es la forma de organizar la información en dos dimensiones. Con sistemas de escritura muy variados el hombre ha conseguido distribuir linealmente la información en una superficie. Entonces, si un texto es la organización de la información en una superficie, en un espacio de tres dimensiones sería un hipertexto"*<sup>67</sup>.

Pierre Lévy menciona *"técnicamente un hipertexto es un conjunto de nudos ligados por conexiones. Los nudos pueden ser palabras, imágenes, gráficos o partes de gráficos, secuencias sonoras, documentos completos que a su vez pueden ser hipertextos. Los items de información no están conectados linealmente, como los nudos de una cuerda, sino en*

<sup>67</sup> HERAS, Antonio de las. *Navegar por la información*. — España : Fundesco, 1991. — p. 84

*modo que cada uno de ellos, o la mayor parte, extienden sus conexiones en forma de estrella, según un modelo reticular. Navegar en un hipertexto es diseñar un recorrido en una red que puede ser tan compleja como se quiera. Porque cada nudo puede contener a su vez toda una red*"<sup>68</sup>.

Bolter define *'hipertexto consiste de tópicos y sus conexiones; los tópicos pueden ser párrafos, oraciones o palabras simples. Un hipertexto es como un libro impreso en el cual el autor tiene disponible un par de tijeras para cortar y pegar pedazos de redacción de tamaño conveniente. La diferencia es que el hipertexto electrónico no se disuelve en una desordenada carpeta de anotaciones: el autor define su estructura definiendo conexiones entre esas anotaciones'*"<sup>69</sup>.

George Landow comenta *"el hipertexto denota un medio de información que conecta informaciones verbales y no verbales. El término hipermedia extiende simplemente la idea del texto en el hipertexto, incluyendo informaciones visuales, sonidos, animaciones y otras formas de información... Yo no hago ninguna distinción entre hipertexto e hipermedia"*<sup>70</sup>.

Carlo Rovelli expresa *"un hipertexto puede ser un conjunto de unidades informáticas, llamadas nudos, conectadas por links; un hipertexto, entonces, no es una sucesión de páginas, sino una red de nudos... Si un hipertexto es una red que conecta nudos informativos sobre la base de una libre asociación no puede limitarse al texto escrito, sino que debe ser necesariamente la síntesis de todas las formas modernas asumidas por la información: textos, músicas, voces, filmaciones..."*<sup>71</sup>.

Ted Nelson explica, *"con hipertexto entiendo una escritura no secuencial, un texto que se divide y permite al lector elegir; algo que se disfruta al máximo delante de una pantalla interactiva. El hipertexto incluye como un caso particular la escritura secuencial, y es por lo tanto la forma más general de escritura. No más limitados por la sola secuencia, con un hipertexto podemos crear nuevas formas de escritura que reflejen la estructura de aquello sobre lo que escribimos; y los lectores pueden elegir recorridos diversos según sus actitudes, o a partir de sus pensamientos, en un modo hasta ahora considerado imposible"*<sup>72</sup>.

*"Hipertexto es el término genérico; hay varios motivos para excluir otros conceptos posibles como 'texto con ramificaciones', 'texto con estructura gráfica', 'texto complejo' o 'texto arborescente'... La mejor definición de hipertexto, que cubre una gran variedad de tipos, es 'estructura que no se puede imprimir en modo apropiado'. Esta definición no es ni muy específica ni profunda, pero es aquella más apta..."*<sup>73</sup>.

---

<sup>68</sup> LÉVY, Pierre citado en SCOLARI, Carlos Alberto. Op. Cit.

<sup>69</sup> BOLTER J. citado en BIANCHINI, Adelaide. **"Conceptos y definiciones de hipertexto."** — *En*

<http://www ldc.usb.ve/~abianc/hipertexto.html>. — Venezuela : Universidad Simón Bolívar : Depto. de Computación y Tecnología de la Información, 2000. Consultado el 3 de septiembre de 2002.

<sup>70</sup> LANDOW, George, citado en SCOLARI, Carlos Alberto. Op. Cit.

<sup>71</sup> ROVELLI, Carlo, Ibidem.

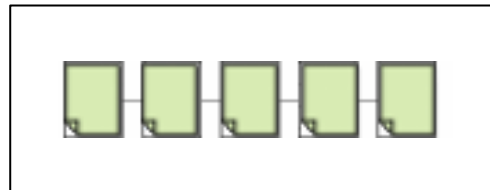
<sup>72</sup> NELSON, Theodor, Ibidem.

<sup>73</sup> Ibidem

### 3.1.3 Elementos básicos y estructura hipertextual

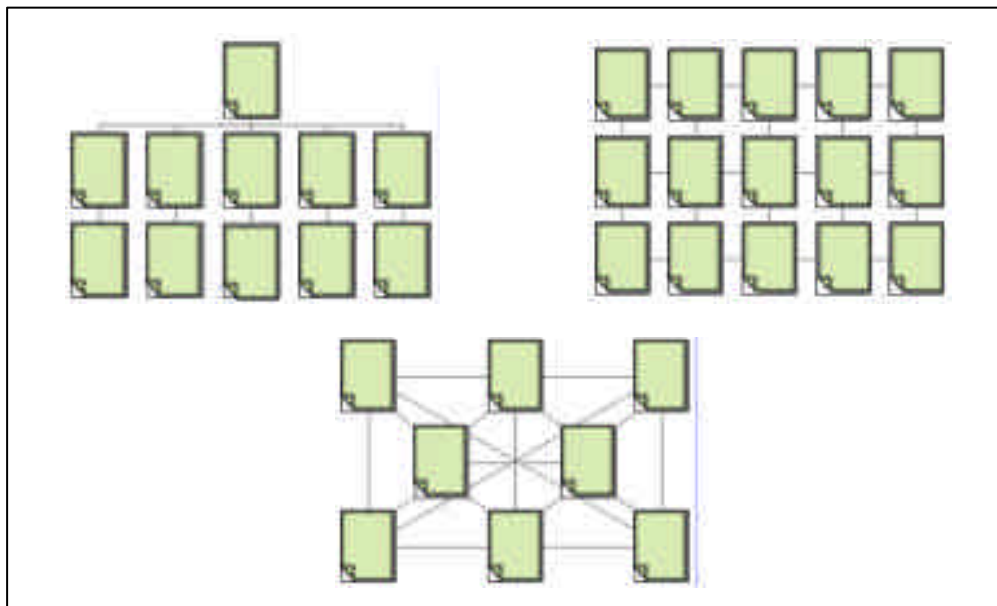
La estructura de un texto puede definirse de acuerdo a la forma que tiene de relacionarse cada uno de sus elementos. El hipertexto presenta dos formas, la lineal y no lineal.

Una estructura es lineal cuando cualquiera de sus elementos sólo puede estar enlazado con un solo elemento anterior y un único elemento posterior.



*Figura. 3.1.3-1 Organización lineal de la información.*

Una estructura es no lineal cuando cada uno de sus elementos puede tener más de un antecesor o más de un sucesor. Esto puede presentarse de varias formas, por ejemplo: que cada elemento tenga varios sucesores pero un solo antecesor y viceversa, o que cada elemento tenga varios antecesores y sucesores a la vez.



*Figura. 3.1.3-2 Organización no lineal de la información.*

La información lineal está ordenada de forma que para recuperarse no necesita referencias a sí misma o a otras informaciones externas; por el contrario la información no lineal presenta referencias o ligas que unen a los elementos con una idea asociativa lo que ocasiona una lectura multisequencial según las decisiones de “saltos” que desee realizar el lector.

La información se ha transmitido y organizado, tradicionalmente, de manera básicamente secuencial. Esto se debe, sobre todo, a que las tecnologías para escribir y transmitir información han impuesto esta forma. La implementación del hipertexto en este contexto proporciona ventajas que lo han colocado como la base para el desarrollo de los proyectos más ambiciosos en el campo de la computación y el desarrollo de sistemas de información mundiales.

Sin embargo, no todos los sistemas de hipertexto cubren las expectativas de sus usuarios o no se explota al máximo su capacidad. Esto se debe a que el sistema no cumple con las características mínimas recomendables para un sistema de esta naturaleza. De acuerdo con Conklin<sup>74</sup>, un sistema de hipertexto ideal debe cumplir con los siguientes mínimos:

- Esta tecnología debe proveer un medio adecuado para organizar y presentar información poco o nada estructurada, no ajustada a esquemas tradicionales y rígidos como es el caso de las bases de datos. Pueden utilizarse esquemas jerárquicos para la utilización de sistemas de documentación de texto tradicionales, muy organizados o simplemente creando estructuras de redes con poco o ningún atributo de precedencia.
- Tener asociada una interfaz de usuario muy intuitiva, pues se pretende imitar el funcionamiento de la mente humana, haciendo uso de modelos cognitivos, por lo que el usuario no debería realizar grandes esfuerzos para obtener la información requerida.
- La información se encuentra distribuida y puede ser accedida en forma concurrente por varios usuarios, por lo tanto es un ambiente compartido.
- Es un ambiente colaborativo: un usuario puede crear nuevas referencias entre dos documentos cualesquiera en forma inmediata e independiente de los tipos de contenido, haciendo crecer su hiperdocumento, sin generar cambios en el hiperdocumento referenciado. Estas referencias pueden estar embebidas en el documento, de modo que aunque éste se cambiara de instalación, el enlace seguiría proporcionando acceso a la información referenciada.
- Tiene asociados varios mecanismos de recuperación y búsqueda de información a través de las navegaciones, ya sean dirigidas o no dirigidas.

Para la mejor comprensión y aplicación de sistemas hipertextuales se han creado modelos que describen las características y funciones básicas de este tipo de aplicaciones. En un nivel general los sistemas de hipertexto se basan en un modelo básico que se divide en submodelo de datos y submodelo de procesos.

---

<sup>74</sup> CONKLIN, J. “**Hipertext: an introduction and survey.**” — p. 33. — *En IEEE Computer* : <http://www.ai.univie.ac.at/~paolo/lva/vuhtmm1998/html/conklin87/Conklin87.html>. — Vol. 20, (sep. 1987) Consultado el 3 de septiembre de 2002.

En el submodelo de datos, “los nodos se interrelacionan mediante enlaces dirigidos, para formar la estructura de un grafo<sup>75</sup>. Todas las operaciones de inserción, eliminación y actualización de nodos y enlaces están provistas de las operaciones de grafos. Los nodos, son las unidades primitivas para organizar la información en el espacio hipermedial: funcionan como colecciones de datos no estructurados los cuales son agrupados de forma tal de crear una entidad lógica. Esta entidad existe para proveer abstracciones coherentes de información dentro del espacio de información representado en la estructura hipermedial. Los enlaces son las otras unidades fundamentales del submodelo de datos: implementan la dirección de las conexiones entre nodos, están asociados como anclas a un nodo fuente y proveen al usuario la habilidad de activarlos para moverse hacia un nodo destino”<sup>76</sup>.

El submodelo de procesos se refiere, en el nivel básico, “a los mecanismos de acceso a la información representada en el hiperdocumento, se requiere de la descripción de la dinámica, es decir las navegaciones que constituye el primer medio que tiene el usuario para acceder a los contenidos y los demás procesos que intervienen en la interacción con el usuario. Por lo tanto, el submodelo de procesos debe contener, además del proceso propio de exploración vía enlaces, otros procesos que pueden influir sobre la forma en que los usuarios inspeccionan la estructura de los hiperdocumentos y que permiten al usuario obtener información de acuerdo a distintas necesidades estudiadas para caracterizar este paradigma”<sup>77</sup>.

De los distintos modelos planteados, el que ha recibido mayor aceptación es el modelo de Arquitectura a 3 niveles de Campbell y Goodman. Este modelo se estructura de la siguiente forma:

**Nivel de presentación o Interfaz de usuario.** “Este nivel comprende los instrumentos de acceso a la información (ratón, cursores, menús, etc.), el sistema de ayuda, los niveles de acceso (autor, lector, semi-autor), las herramientas de navegación (localizadores, mapas o “browsers”, posibilidades de retroceso o “backtrack”, etc.), y la metáfora global del sistema (presentación de la información homogenizada). Este nivel está estrechamente interrelacionado con el HAM, pues la organización de la información determina en gran medida las posibilidades de presentación”<sup>78</sup>.

**Nivel HAM (Hypertext Abstract Machine) o Elementos estructurales.** “La HAM está en el centro de la arquitectura general de un sistema hipertexto y determina la naturaleza básica de sus nodos y enlaces junto a las relaciones mantenidas entre ellos. La HAM tendría conocimiento de la forma de los nodos y enlaces y conocería qué atributos están relacionados con ellos”<sup>79</sup>.

---

<sup>75</sup> Un grafo es una lista no lineal, formada por un conjunto de nodos relacionados y conectados entre sí por un conjunto de líneas.

<sup>76</sup> BIANCHINI, Adelaide. Op. Cit.

<sup>77</sup> Ibidem

<sup>78</sup> CANTOS GÓMEZ, Pascual. *Hipertexto y documentación* / Francisco Javier Martínez Méndez y Gregorio Moya Martínez. — España : Secretariado de Publicaciones, Universidad, 1994. — p. 26

<sup>79</sup> Ibidem

**Nivel de Base de Datos.** “Este nivel se encuentra en el fondo de la arquitectura a tres niveles y hace referencia a todos los aspectos tradicionales del almacenamiento de información. Es transparente al usuario de un sistema hipertexto y se entiende como estructura de ficheros, códigos de identificación, herramientas de control, etc.”<sup>80</sup>.

Los tipos de enlaces en un sistema hipertexto según el destino de la relación son:

- **Externos e internos:** si el enlace se produce entre los nodos pertenecientes al mismo documento, o entre nodos de dos documentos distintos.
- **De conexión entre aplicaciones:** son aquellos que permiten activar una aplicación distinta a la del Software de hipertexto en la que está incluida.
- **De comando:** aquellas que conectan un nodo y una acción o una orden del programa.
- **De unirrelación o de multirrelación:** un nodo puede estar enlazado con otro nodo (1:1), pero también existen los llamados ‘fat links’, que relacionan con varios nodos (1:3).
- **Conectados a una ventana virtual:** los nodos que están entrelazados con un ‘pop-up menú’ son los nodos más seguros de usar, pues no suponen un desplazamiento del curso del documento origen.
- **De información multimedia:** Este tipo de enlaces tienen requerimientos especiales, dadas las características de reproducción de la imagen animada o del sonido digital.
- **De desplazamientos:** Son aquellos que se desplazan a otro lugar del mismo o de otro documento.

### 3.2 Documentos digitales

El desarrollo de los soportes de información ha dado un gran salto con la aparición de tecnologías digitales que permiten la creación de documentos en diversos formatos con características de almacenamiento, organización, edición, transferencia, acceso y capacidad, que soportes anteriores no presentaban.

El diseño y la producción de documentos digitales se inició con la digitalización de los ya impresos, dando prioridad a los de mayor valor histórico. Para esto se utilizó el scanner y los documentos se presentaban como colecciones de imágenes con la idea hipertextual para ir hojeando las imágenes simulando las hojas del libro. Después se continuó con la idea de escanear documentos y por medio de la tecnología OCR convertirlos a texto en formato digital.

---

<sup>80</sup> CANTOS GÓMEZ, Pascual. Op. Cit., p. 27



Posteriormente se impulsó la idea de crear documentos a partir de procesadores de texto, esto con la idea de producir documentos impresos, pero con su origen en una computadora. Finalmente se ha visto un enorme crecimiento en la aparición de diversos documentos con distintos formatos y cualidades digitales que nacen y se desarrollan únicamente en el ámbito digital.

Esto trae consigo un crecimiento acelerado en la producción de documentos que tienen que ser organizados, conservados y difundidos, de forma que la cantidad de ellos no sea obstáculo para su recuperación.

No solo la explosión de información y su control han sido tema de discusiones y posiciones divididas. La definición de términos propios a las nuevas formas de los soportes se muestra en la diversidad de definiciones utilizadas.

El término documento digital se refiere a *“un fenómeno que encierra diversas características relacionadas con las prácticas sociales que emplean la tecnología de la información y las redes de telecomunicación como medios para registrar, estructurar y difundir información”*<sup>81</sup>.

Como hemos mencionado anteriormente, los documentos son, esencialmente, instrumentos y soportes de comunicación que recogen formas objetivadas de una experiencia o de un conocimiento relacionado con un determinado sector de las actividades del espíritu humano.

Por otra parte lo digital, se refiere a la información que sólo puede variar de forma discreta. Es lo contrario que analógico, que es aquella información que puede variar de forma continua. Entendemos entonces a lo digital como la información en unos y ceros, en apagado o encendido, etc. De aquí que consideremos a los documentos digitales como todo aquel soporte electrónico (eléctrico, magnético u óptico) que contenga formas objetivadas del conocimiento y estas sólo se compongan o se codifiquen por dos valores.

### 3.2.1 Lenguajes de marcado

En la historia del origen de los documentos escritos se puede apreciar que los primeros carecieron de una estructura textual debido a la ausencia de normas fijas para su organización semántica.

Elsa M. Ramírez Leyva apunta: *“... la escritura alfabética no dio como resultado inmediato el texto; éste requirió de todo un método que lo fue perfeccionando conforme su uso se hizo más intensivo y diverso. Pero debieron pasar varios siglos para lograr el predominio del texto sobre el discurso oral, ... Más tarde se lograría –en el siglo XVII– la exactitud requerida para trasladar a la página el trabajo de observación experimental,*

---

<sup>81</sup> LAFUENTE LÓPEZ, Ramiro. *En el umbral del cambio : las tecnologías de la información y la comunicación.* — México : UNAM, Centro Universitario de Investigaciones Bibliotecológicas, 1997. — p. 34

*mediante la aplicación de normas para la elaboración del texto sin distorsión o pérdida de información, e incluso los elementos complementarios de las obras fueron estandarizados. En síntesis, la página se legitima en cuanto a su composición y uso en el texto escrito, al igual que sucedería más tarde con el impreso... ”<sup>82</sup>.*

Un fenómeno similar ocurre con el texto digital y es por ello que surgen los lenguajes de marcado para estructurar documentos digitales, cuya idea básica se sustentó en crear medios para estructurar información, independientemente del software que se utilice.

### 3.2.1.1 SGML

Standard Generalized Mark-Up Language o SGML es el Lenguaje Estándar Generalizado de Marcado<sup>83</sup> y actualmente es la base para la descripción de información digital. Este metalenguaje fue establecido por la Organización Internacional de Normalización (ISO) y se ideó para que a través de él se pudieran definir lenguajes (mark-up languages, un ejemplo es el HTML) que sirvieran para marcar aquellos elementos que identificaran documentos digitales, permitiendo el intercambio entre diferentes plataformas.

Daniel M. Germán y Alejandro López-Ortiz en su artículo “SGML: Lo que está escrito en los espacios en blanco” describen como ventajas o características principales las siguientes:

- El estándar de SGML no incluye información sobre como deben ser interpretadas las anotaciones. Su principal objetivo es describir las unidades lógicas de un documento, más no como deben ser tipografiadas.
- SGML formaliza la idea de asociar los documentos dependiendo el tipo, al crear la noción de Definición de Tipo de Documento (Document Type Definition - DTD). Una DTD describe una clase o familia de documentos con ciertas características comunes.
- SGML evita la dependencia de los conjuntos de caracteres, tales como ASCII y EBCDIC, específicos a fabricantes de equipo. SGML utiliza caracteres normalmente encontrados en casi todos los sistemas y define mecanismos para emplear caracteres especiales o poco comunes.
- SGML logra la independencia de la plataforma-procesador de texto que utiliza una organización. Normalmente, la transición de un procesador de textos de un fabricante a otro es una tarea que requiere que cada uno de los documentos sea convertido al nuevo formato si no se quiere caer en el riesgo de que el futuro el documento sea ilegible, que no es el caso con SGML.<sup>84</sup>

---

<sup>82</sup> RAMIREZ LEYVA, Elsa M. “El lector: del mundo manuscrito al cibernético.” — p. 101-102. — *En La información en el inicio de la era electrónica* / Centro Universitario de Investigaciones bibliotecológicas. — México : UNAM, El Centro, 1998.

<sup>83</sup> Mark-Up es un término de imprenta que significa el conjunto de instrucciones estilísticas detalladas escritas en un manuscrito que debe ser tipografiado.

<sup>84</sup> GERMÁN, Daniel M. *SGML: Lo que está escrito en los espacios en blanco* / Daniel M. Germán y Alejandro López-Ortiz. — Canadá : Universidad de Waterloo, 199?. p. 1

SGML fue aceptado como estándar en 1986. Desde entonces ha sido utilizado con éxito en diversas áreas.

### 3.2.1.2 HTML

Hyper Text Mark-Up Language o HTML es el Lenguaje de Marcado para Hipertexto y se conoce como el lenguaje de programación de documentos hipertextuales o páginas para el World Wide Web. Este lenguaje fue creado como un DTD (Definición de Tipo de Documento) de SGML. *“El HTML es un lenguaje cuya finalidad es organizar la estructura de un documento digital en forma de hipertexto, con objeto de que éste pueda ser fácilmente procesado por un programa visualizador independientemente de la máquina o el sistema operativo que se emplee”*<sup>85</sup>.

La relevancia que adquiere el HTML por su aceptación como un lenguaje estandarizado para la clasificación de los elementos característicos de documentos digitales hipertextuales en el WWW y la utilidad como herramienta para diseñar, visualizar y recuperar documentos, conlleva a la imperiosa necesidad por parte de los bibliotecarios de comprender en su totalidad este lenguaje.

El estudio de este lenguaje requiere sobre todo de práctica y podría abarcar todo un libro la explicación del lenguaje. A continuación se describirá el lenguaje de manera general a modo de introducción en el estudio del mismo, no se abordará exhaustivamente la programación de documentos en HTML, se explicarán los conocimientos básicos para el desarrollo de habilidades en el manejo de estos documentos.

#### Generalidades

Para iniciar en el desarrollo de documentos en HTML se emplean las siguientes aplicaciones:

- Se necesita de un editor de texto plano para poder escribir el programa, se recomienda el “Block de notas” o “Wordpad” que incluye el sistema *Windows* en su menú de accesorios.
- La segunda aplicación utilizada es un *browser* para poder ver el resultado de la programación (Ej. Internet Explorer, Netscape Navigator, Mosaic, etc.).

Todo programa de HTML está formado de “tags” que vienen siendo como los comandos del lenguaje. Estos *tags* van a ir encerrados entre corchetes angulares (“<” y “>”). Entonces, el programa en HTML debe comenzar así:

---

<sup>85</sup> LAFUENTE LÓPEZ, Ramiro. Op. Cit., p. 105

```
<HTML>
<!-- El código de la página -->
</HTML>
```

La primera línea de código indica que se trata de un documento en HTML. La segunda línea es un comentario, el cual no va a ser tomado en cuenta por el *browser* y que sirve para realizar anotaciones personales o avisos dentro del programa que ayuden a la identificación de las partes del documento. Dicho de otra forma, todo lo que esté entre “<! -- y -->” no va a aparecer en el *browser*.

La tercera línea indica el final del código. Este *tag* tiene un “slash” o “/” después del corchete angular. Cuando un *tag* tiene el símbolo de *slash*, quiere decir que se está cerrando algún *tag*, en este caso el de <HTML>. Cabe mencionar que el *browser* no es sensible a las mayúsculas y minúsculas, por lo tanto se puede escribir <HTML> o <html> o <HtMl>, de igual forma no toma en cuenta si el código está escrito en una sola línea, pero se recomienda separarlo en bloques de acuerdo a la programación para ir separando instrucciones y no perderse en el contenido.

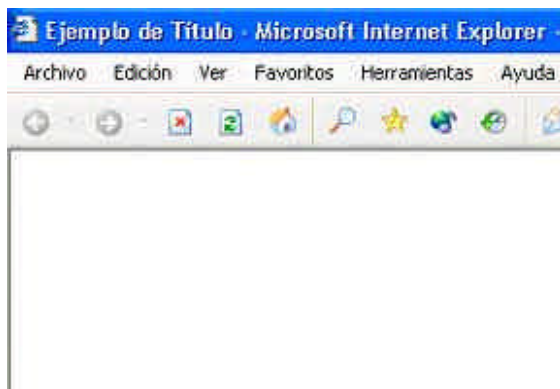
Ahora, todo programa de HTML tiene una cabeza y un cuerpo:

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<!-- El código de la página -->
</BODY>
</HTML>
```

Dentro de la cabeza pueden ir comentarios (de hecho, estos pueden ir donde sea) y el *tag* <TITLE>, este *tag* va a poner un título hasta arriba del *browser*. Por ejemplo :

```
<HTML>
<HEAD><TITLE>Ejemplo de Título</TITLE>
</HEAD>
<BODY>
<!-- El código de la página -->
</BODY>
</HTML>
```

Este código se escribe en el procesador de texto plano y se guarda con la extensión .htm o .html (Ej. ejemplo.htm o ejemplo.html). Posteriormente se abre el archivo en el *browser* (Ej. Internet explorer, Netscape, etc.). Este código visto en el *browser* muestra lo siguiente:



Como se observa no aparece el comentario y la ventana lleva el título en la parte superior. Dentro de <BODY> se encuentra el contenido de la página y dentro del tag <BODY> (que es diferente a decir dentro de <BODY>) pueden ir parámetros para cambiar el color del fondo, de las ligas y del texto.

Si en el código de arriba se cambia la cuarta línea por:

```
<BODY TEXT=999999 LINK=990000 ALINK=009900 VLINK=000099
BGCOLOR=000000>
```

Se observa en el *browser* que la parte donde iría la página cambió a negra<sup>86</sup>. Esto es por el atributo BGCOLOR que tomó el valor del color negro. El parámetro TEXT sirve para indicar el color del texto que aparecerá en la página, LINK, ALINK y VLINK sirven para las ligas: el primero da el color de cómo se verá la liga antes de entrar en ella; el segundo indica el color de cómo se verá la liga mientras se oprime y el último indica como se verá la liga cuando ya halla sido visitada anteriormente.

Si se observa, cada parámetro va seguido de 6 dígitos, estos representan los colores. Los dos primeros indican la cantidad de rojo que se quiere, los 2 siguientes indican la cantidad de verde y los 2 últimos indican el azul. Esto es porque se está trabajando en RGB (Red, Green, Blue). Para obtener los diferentes colores hay que mezclar diferentes cantidades en los tres componentes. Por ejemplo:

El blanco se hace con 100% de Red, 100% de Green y 100% de Blue.

El morado se hace con 100% de Red, 0% de Green y 100% de Blue.

El negro se hace con 0% de Red, 0% de Green y 0% de Blue.

El amarillo con 100% de Red, 100% de Green y 0% de Blue.

Se sabe que al mezclar amarillo y azul el color que se obtiene es verde, y no que la mezcla de rojo y verde da amarillo, esto se debe a que son considerados colores primarios, pero esto sucede en el sistema CMYK, el cual es diferente a RGB y en este último no se considera primario el amarillo.

---

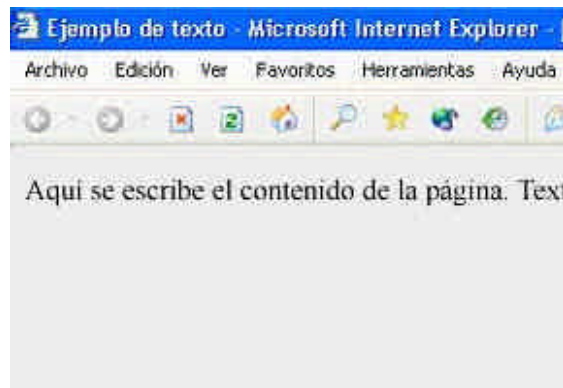
<sup>86</sup> Para ver el resultado primeramente se guardan los cambios en el archivo con el nombre "ejemplo.htm" después se oprime el botón "actualizar" del *browser*; esto para que llame nuevamente al documento y lea los cambios realizados.

Ahora, los atributos de BODY (TEXT, LINK, etc.) van a tomar valores “hexadecimales”, o sea que en vez de ir de 0 a 9 como en base 10, van a ir de 0 a 15. Por lo tanto, los dígitos hexadecimales son: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, donde A vale 10, B vale 11, C vale 12, etc. Así se logra tener un total de 16777216 colores. Entonces un rojo al 100% sería: FF0000.

## Texto

Como se mencionó anteriormente dentro de <BODY> se encuentra el contenido de la página.

```
<HTML>
<HEAD><TITLE>Ejemplo de texto</TITLE>
</HEAD>
<BODY TEXT=000000 LINK=990000 ALINK=009900 VLINK=000099
BGCOLOR=c0c0c0>
<!-- El código de la página -->
Aquí se escribe el contenido de la página.
Texto...
</BODY>
</HTML>
```



Se puede ver como las dos líneas de texto aparecen juntas, es decir, el *browser* no respetó el “enter”. Esto se soluciona poniendo el *tag* <BR> en donde se quiera que exista un *enter*. Este *tag* no necesita cerrarse, o sea, no requiere de </BR> el cual no existe aunque si se pone lo puede o no tomar como <BR> (depende de que *browser* se esté usando para visualizar el documento).

Otros *tags* que están involucrados con texto son:

```
<CENTER>texto</CENTER> Sirve para centrar todo aquello que esté entre ambos tags.
<H1>texto</H1> Coloca el texto en letras muy grandes.
<H2>texto</H2> Coloca el texto en letras grandes.
<H3>texto</H3> Coloca el texto en letras un poco grandes.
```

<H4>texto</H4> Coloca el texto en letras un poco pequeñas.  
<H5>texto</H5> Coloca el texto en letras pequeñas.  
<H6>texto</H6> Coloca el texto en letras muy pequeñas.  
<FONT SIZE=5 COLOR=FF0000>texto</FONT>

En SIZE se pone el tamaño de letra en píxeles y en COLOR el color que se quiera poner a lo que vaya entre ambos *tags*.

Ejemplo:

```
<HTML>
<HEAD><TITLE>Ejemplo</TITLE>
</HEAD>
<BODY TEXT=000000 BGCOLOR=FFFFFF>
<CENTER><H1>Bienvenidos</H1></CENTER>
<BR>
<H2>Menú</H2>
<BR>
Opción A <BR>
Opción B <BR>
Opción C <BR>
Opción D <BR>
Opción E <BR>
Opción F <BR>
<BR>
<H6><FONT COLOR=FF0000>Advertencia:</FONT>Fin del documento</H6>
</BODY>
</HTML>
```



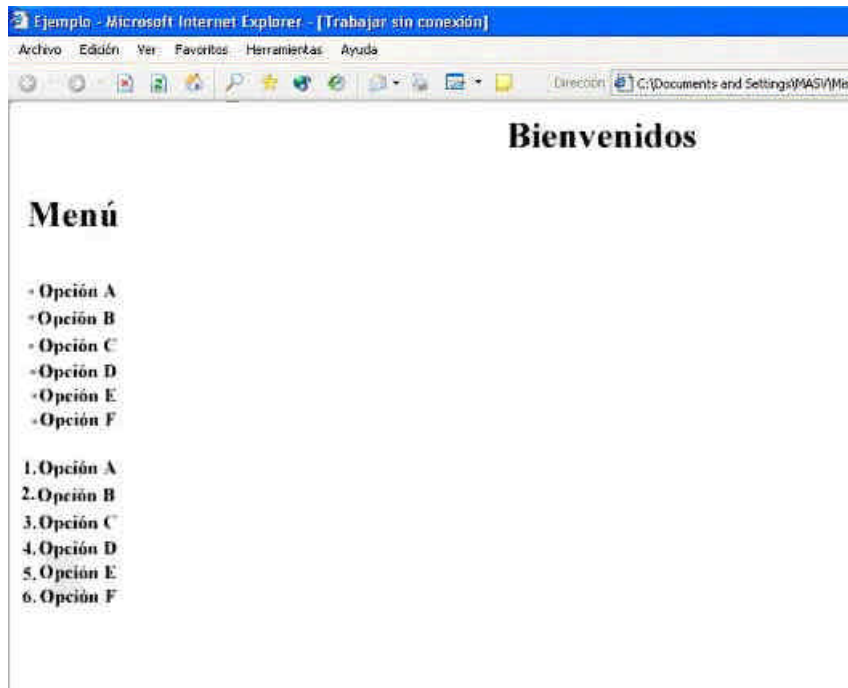
Para tener las opciones numeradas existen los siguientes *tags* :

<MENU><MENU> Que sólo recorre la lista de opciones a la derecha. <OL></OL> funciona igual que <MENU>, <LI> le coloca una bolita a cada opción si se está usando <MENU> o le pone números si se usa <OL>.

Ejemplo:

```
<HTML>
<HEAD><TITLE>Ejemplo</TITLE>
</HEAD>
<BODY TEXT=000000 BGCOLOR=FFFFFF>
<CENTER><H1>Bienvenidos</H1></CENTER>
<BR>
<H2>Menú</H2>
<BR>
<MENU>
<LI>Opción A <BR>
<LI>Opción B <BR>
<LI>Opción C <BR>
<LI>Opción D <BR>
<LI>Opción E <BR>
<LI>Opción F <BR>
</MENU>
<BR>
<OL>
<LI>Opción A <BR>
<LI>Opción B <BR>
<LI>Opción C <BR>
<LI>Opción D <BR>
<LI>Opción E <BR>
<LI>Opción F <BR>
</OL>
<BR>
<H6><FONT COLOR=FF0000>Advertencia:</FONT>Fin de documento</H6>
</BODY>
</HTML>
```





## Tablas

La información que se puede mostrar en un documento hipertexto puede ser muy extensa y en ocasiones será necesario tenerla ordenada de forma estructurada. Para esto se pueden crear las tablas.

```
<HTML>
<HEAD><TITLE>Ejemplo de tablas</TITLE>
</HEAD>
<BODY TEXT=000000 BGCOLOR=FFFFFF>

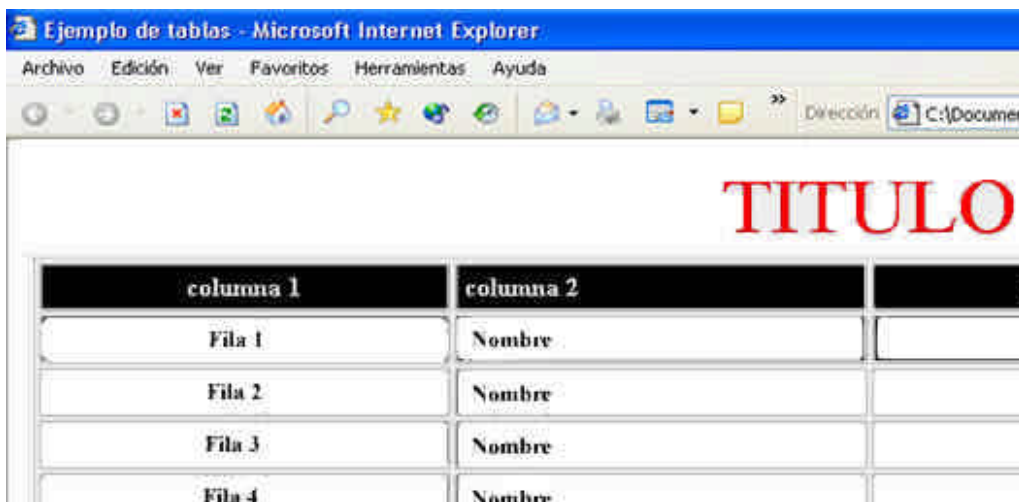
<CENTER><FONT SIZE=10 COLOR=FF0000>TITULO</FONT></CENTER>

<TABLE BORDER=4 CELLPADDING=4 CELLSPACING=4 WIDTH=100%>
<TR>
<TH WIDTH=25% BGCOLOR=000000>columna 1</TH>
<TH WIDTH=25% ALIGN=LEFT>columna 2</TH>
<TH WIDTH=25% ALIGN=CENTER>columna 3</TH>
<TH WIDTH=25% ALIGN=RIGHT>columna 4</TH>
</TR>
<TR>
<TD WIDTH=25% BGCOLOR=303030 ALIGN=CENTER>Fila 1</TD>
<TD WIDTH=25% BGCOLOR=303030 ALIGN=LEFT>Nombre</TD>
<TD WIDTH=25% BGCOLOR=303030 ALIGN=CENTER>Nombre</TD>
<TD WIDTH=25% BGCOLOR=303030 ALIGN=RIGHT>Nombre</TD>
</TR>
```

```

<TR>
<TD WIDTH=25% ALIGN=CENTER>Fila 2</TD>
<TD WIDTH=25% ALIGN=LEFT>Nombre</TD>
<TD WIDTH=25% ALIGN=CENTER>Nombre</TD>
<TD WIDTH=25% ALIGN=RIGHT>Nombre</TD>
</TR>
<TR>
<TD WIDTH=25% ALIGN=CENTER>Fila 3</TD>
<TD WIDTH=25% ALIGN=LEFT>Nombre</TD>
<TD WIDTH=25% ALIGN=CENTER>Nombre</TD>
<TD WIDTH=25% ALIGN=RIGHT>Nombre</TD>
</TR>
<TR>
<TD WIDTH=25% ALIGN=CENTER>Fila 4</TD>
<TD WIDTH=25% ALIGN=LEFT>Nombre</TD>
<TD WIDTH=25% ALIGN=CENTER>Nombre</TD>
<TD WIDTH=25% ALIGN=RIGHT>Nombre</TD>
</TR>
<TR>
<TD WIDTH=25% ALIGN=CENTER>Fila 5</TD>
<TD WIDTH=25% ALIGN=LEFT>Nombre</TD>
<TD WIDTH=25% ALIGN=CENTER>Nombre</TD>
<TD WIDTH=25% ALIGN=RIGHT>Nombre</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```



Una tabla tiene renglones y columnas. En HTML los renglones se definen entre <TR> y </TR>. A cada renglón hay que ponerle sus columnas, las cuales van entre <TD> y </TD> para texto normal o entre <TH> y </TH> para texto en negrillas o *bold*. Dentro de <TH> o

<TD> existen atributos como ALIGN que alinea el texto a la izquierda, lo pone en el centro o lo pega al lado derecho; otro atributo comando es WIDTH con el que se especifica qué porcentaje de la tabla va a ocupar esa columna.

Todo lo anterior debe ir dentro de los *tags* <TABLE y </TABLE>. Dentro de este *tag* también van los siguientes atributos: CELLPADDING es el espacio alrededor del elemento de la tabla. CELLSPACING es el ancho de las líneas de separación entre renglones y columnas. BORDER es el ancho del marco. WIDTH y HEIGHT indican la cantidad de espacio que va a ocupar toda la tabla, el cual puede anotarse en porcentaje (como en este caso, se le está diciendo que ponga la tabla de manera que ocupe el 100% de la longitud del *browser*) o en píxeles. Dentro de las columnas el segundo renglón del ejemplo está el atributo BGCOLOR=303030 con el que se le indica al *browser* que el fondo de ese cuadro lo cambie por un gris oscuro.

### Imágenes

Algo importante que hay que explicar es cómo poner imágenes. El *tag* <IMG> sirve para esto y se usa así:

```
<HTML>
<HEAD><TITLE>Este es un ejemplo de imagen</TITLE>
</HEAD>
<BODY TEXT=000000 BGCOLOR=FFFFFF>
<CENTER>
<FONT SIZE=5>
Este es un ejemplo de cómo poner una imagen
</FONT>
<BR><BR>
<IMG SRC="imagen.jpg">
</CENTER>
</BODY>
</HTML>
```



La línea `<IMG SRC="imagen.jpg">` indica al *browser* que coloque una imagen, la cual se llama "imagen.jpg" y está en el mismo directorio donde está el archivo de HTML. Aquí es importante remarcar algunos aspectos:

1. Las imágenes que se pueden mostrar deben ser del tipo "JPG" o "GIF".
2. Dentro del *tag* de `<IMG>` pueden estar los atributos `ALIGN` y `BORDER`. El primero puede llevar los valores `LEFT` o `RIGHT`, y el segundo toma valores de 0 al número que se quiera. Ejemplo:

```
<HTML>
<HEAD><TITLE>Esta es una imagen de prueba</TITLE>
</HEAD>
<BODY TEXT=000000 BGCOLOR=FFFFFF>
```

```
<IMG SRC="imagen.gif" WIDTH=145 HEIGHT=145>
texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto,
texto, texto,
texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto,
texto, texto, texto,...
```

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

```
<IMG SRC="iimagen.gif" ALIGN="LEFT" WIDTH=145 HEIGHT=145>
texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto,
texto, texto,
texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto, texto,
texto, texto, texto,...
```

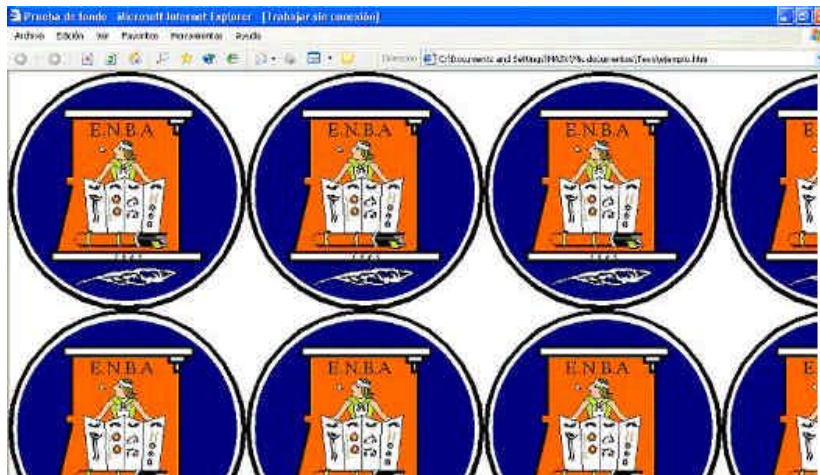
```
<BR><BR>
<BR><BR>
```



Las imágenes que presentan movimiento son un formato especial de GIF. Para crear estas animaciones es necesario hacer todos los cuadros de animación de la imagen y luego con un paquete especial se juntan y se les da valores de tiempo. Otra cosa que también ofrece el formato GIF es la transparencia. Si se observa, el fondo de la página se puede ver en aquellas partes del dibujo que no tienen nada.

Otro de los atributos que va dentro del tag <BODY> es BACKGROUND y sirve para poner una imagen como fondo en la página:

```
<HTML>
<HEAD><TITLE>Prueba de fondo</TITLE>
</HEAD>
<BODY TEXT=FFFFFF BGCOLOR=000000 BACKGROUND="fondo.jpg">
</BODY>
</HTML>
```

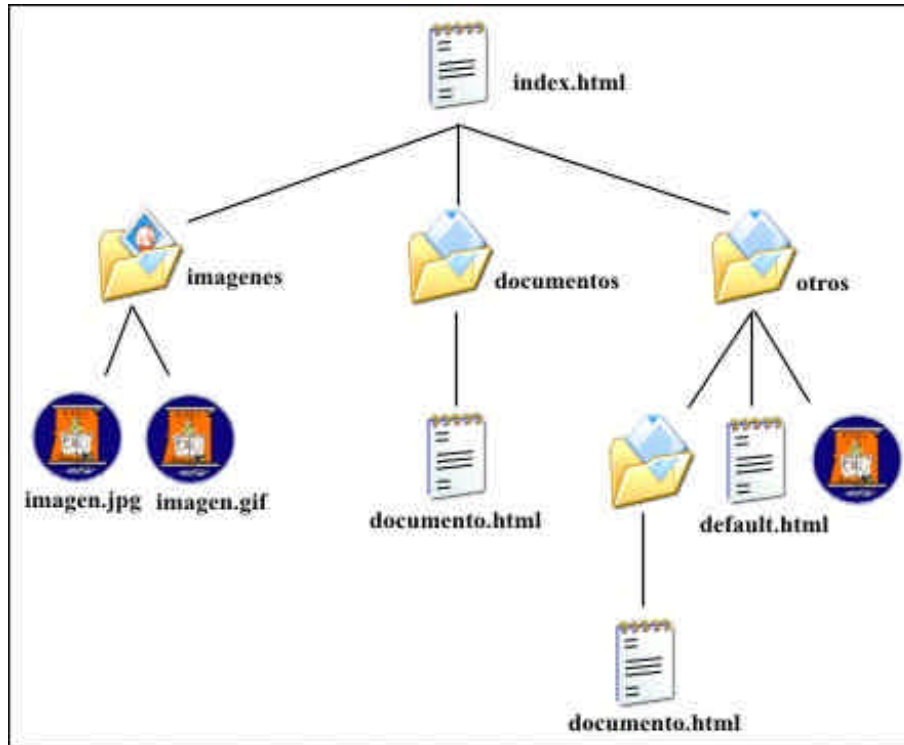


Se deja el atributo BGCOLOR para que el fondo quede negro en lo que el *browser* carga la imagen del fondo, la cual la va a repetir varias veces para que cubra todo el espacio que ocupa la página.

Un aspecto importante en el diseño de documentos en HTML son las ligas, las cuales pueden ir dirigidas a otras partes de la misma página, a páginas que son parte de la página principal, o archivos en otros lugares de la red.

Una liga se establece así: <A HREF="path"> El nombre de la liga </A>. En la parte de " El nombre de la liga " se puede poner texto o imágenes y cuando se hace *click* en esa parte, el *browser* desplegará el contenido de *path*. Cuando hay texto dentro del *tag* <A> y </A>, éste aparece subrayado. Cuando es una imagen aparece con un marco del color que se hayan puesto las ligas (en LINK de <BODY>). El grosor de este marco se controla con BORDER; si se le pone 0, entonces no aparecerá ningún marco alrededor de la imagen.

Un Path puede ser el nombre de una imagen, página o archivo que puede estar precedido o no de nombres de directorios y también puede ser una dirección de Internet.



Ejemplo :

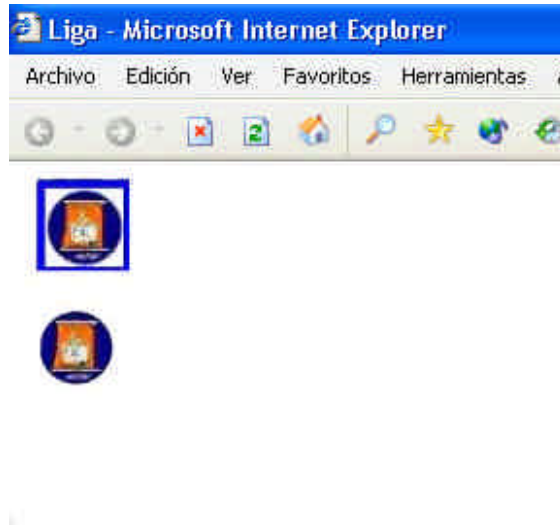
"imagen.jpg" Indica al *browser* que la imagen se encuentra en el mismo directorio en donde está el documento que la manda llamar. "imagenes/fondo.jpg" Indica al *browser* que la imagen está un directorio atrás de donde el documento. "/libros/imagenes/portada.jpg" Con la / se le indica al *browser* que busque la imagen en ese directorio, sin importar donde está el documento. Es como si se pusiera <http://www.enba.sep.gob.mx> ya que el *browser* irá a esa dirección sin importar dónde se encuentre el documento.

Ejemplo:

```

<HTML>
<HEAD>
<TITLE>Liga</TITLE>
</HEAD>
<BODY TEXT=000000 BGCOLOR=FFFFFF>
<A HREF="http://www.enba.sep.gob.mx">
<IMG SRC="icono.jpg" BORDER=5><BR><BR>
<IMG SRC="icono.jpg" BORDER=0><BR><BR>
ENBA
</A>
  
```

```
</BODY>  
</HTML>
```



Si se coloca:

```
<A HREF="http://www.enba.sep.gob.mx" onmouseover="window.status='Escuela  
Nacional de Biblioteconomía y Archivonomía'; return true">ENBA </A>
```

, pondrá un mensaje hasta abajo del *browser*.



## Mapas

En las ligas normales, si se hace *click* en cualquier parte de la imagen, siempre entrará a la misma liga, pero en los mapas, si se hace *click* en diferentes partes de la misma imagen, lleva a lugares distintos.

La sintaxis para estos es la siguiente:

```
<MAP NAME="nombreDelMapa">  
<AREA SHAPE="rect" COORDS="0,0,80,80" href="liga1.html">  
<AREA SHAPE="circle" COORDS="150,90,10" href="liga2.html">  
<AREA SHPAE="poly" COORDS="0,292,30,292,15,280" href="liga3.html">  
<AREA SHPAE="default" nohref></MAP>
```



Este código se puede poner en cualquier parte de la página, pero de preferencia junto a la imagen que se va a usar.

En la estructura del mapa hay un campo que dice NAME=nombreDelMapa. Este nombre es el que va a identificar al mapa y sirve para ligarlo a la imagen. Ahora, la sintaxis de la imagen que va a utilizar el mapa es:

```
<IMG SRC="dado.jpg" WIDTH=330 HEIGHT=255 USEMAP="#nombreDelMapa" BORDER=0>
```

USEMAP es el atributo que liga al mapa con la imagen, usando el nombre del mapa precedido por un #.

Es necesario mencionar que la imagen debe ser trabajada en pixeles y que el origen de una imagen está siempre localizado en la esquina superior izquierda.

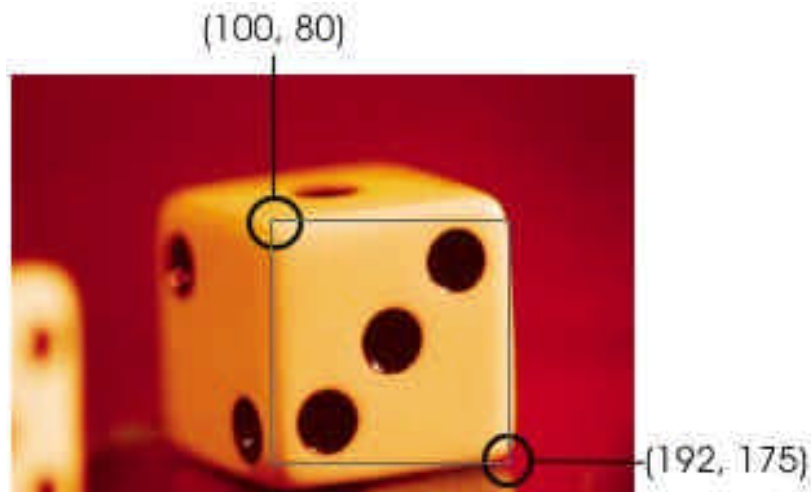


Cada parte de la imagen debe ser definida dentro del *tag* `<AREA SHAPE="FORMA" COORDS="COORDENADAS" HREF="DIRECCION">`

Existen 4 tipos de formas (SHAPES), que pueden ser manejadas en los mapas: los rectángulos, los círculos, los polígonos y lo que sobra.

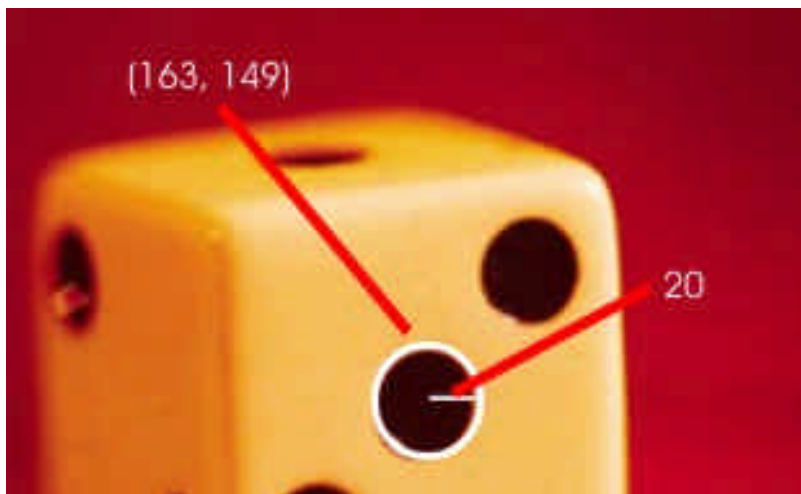
Si se quiere especificar un rectángulo, el atributo SHAPE debe tomar el valor de *rect* y el atributo COORD debe tener 4 valores entre comillas, que son las posiciones de las esquinas opuestas del rectángulo. Por ejemplo, si se quiere que una cara del dado sea una liga a la página "dado.com", entonces el *tag* completo AREA quedaría:

```
<AREA SHAPE=rect COORDS="100,80,192,175" HREF="http://www.dado.com">
```



Los dos primeros números en el atributo COORDS (x,y) representan la esquina superior izquierda, que está localizada a una altura de “y” píxeles y a una distancia horizontal del origen de “x” píxeles. Los dos siguientes números son las coordenadas de la esquina inferior derecha que delimitan al cuadrado.

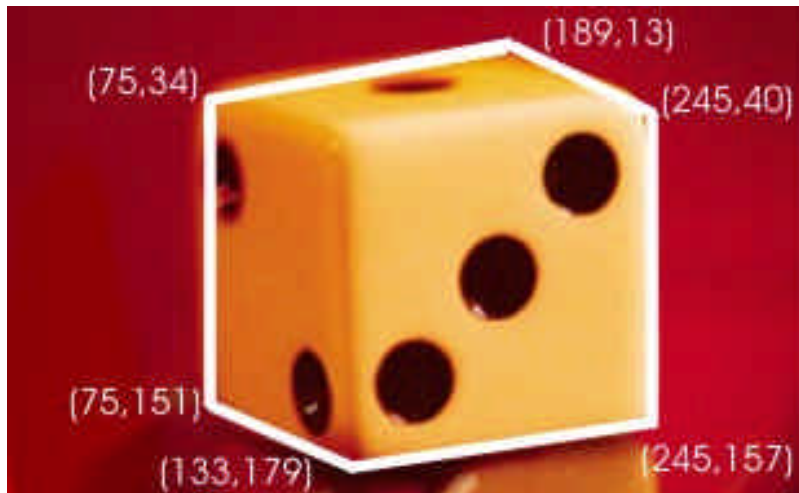
Si se quiere que únicamente el círculo del centro sea una liga, se utiliza *circ*:



`<AREA SHAPE=circ COORDS="163,149,20" HREF="http://www.dado.com">` Aquí los dos primeros números del atributo COORDS son la posición del centro del círculo y el tercer número es el radio en píxeles de éste.

Si se quisiera delimitar un área que no fuera ni un círculo y ni un cuadrado, entonces hay que usar un polígono. Por ejemplo, si se requiere que todo el dado sea una liga se tendrían que unir 6 puntos, por lo que el *tag* quedaría:

`<AREA SHAPE=poly COORDS="189,13,245,40,245,157,133,179,75,151,75,34" HREF="http://www.dado.com">`



Las coordenadas de cada vértice o punto del polígono se colocan en el mismo orden en que se recorrería el contorno de la figura (la dirección no importa).

Finalmente, existen varias regiones de la imagen que no se encierran de ninguna forma, por lo que existe la opción default para el atributo SHAPE. De esta forma se le indica que todo aquello que no está marcado sea otra liga.

```
<AREA SHAPE=default HREF="http://www.otraliga.com">
```

### Frames

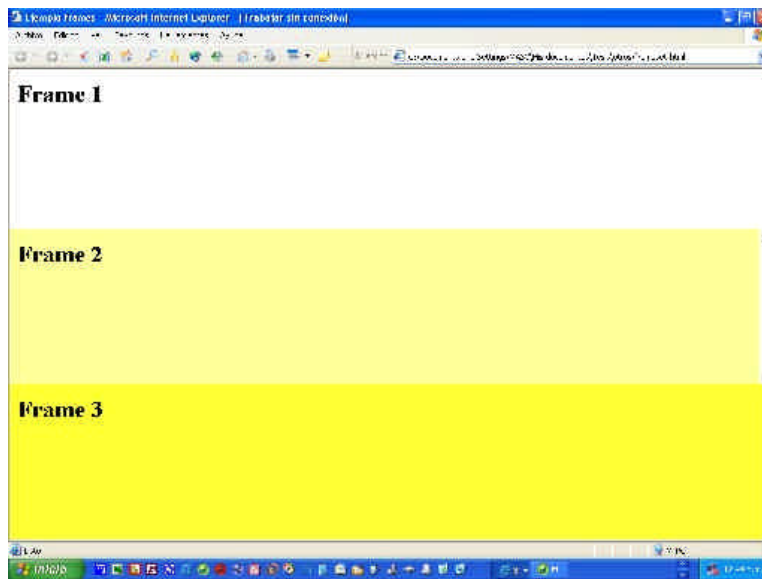
Los *frames* permiten mostrar más de una página a la vez en el *browser*. El siguiente es un ejemplo de cómo crear una página que tenga 3 *frames*.

```
<html>
<head>
<title>Ejemplo frames</title>
</head>
<frameset rows="34%,33%,33%" frameborder="no" border="0" framespacing="0"
noresize>
  <frame src="frame1.html" name="superior" scrolling="no">
  <frame src="frame2.html" name="medio" scrolling="yes">
  <frame src="frame3.html" name="inferior" scrolling="no">
</frameset>
</html>
```

Este es el código principal para la creación de *frames* y generalmente se guarda como frameset.html. Un ejemplo de las páginas que ocuparan los *frames* puede ser:

```
<html>
<head>
<title>frame 1</title>
</head>
<body bgcolor="#FFFFFF">
<h1>Frame 1 </h1>
</body>
</html>
```

Este texto se guarda con el nombre frame1.html. De este último archivo se modifica la parte donde dice Frame 1 (tanto en el título como en el cuerpo de la página) y se pone Frame 2; y se guarda como frame2.html y así sucesivamente.



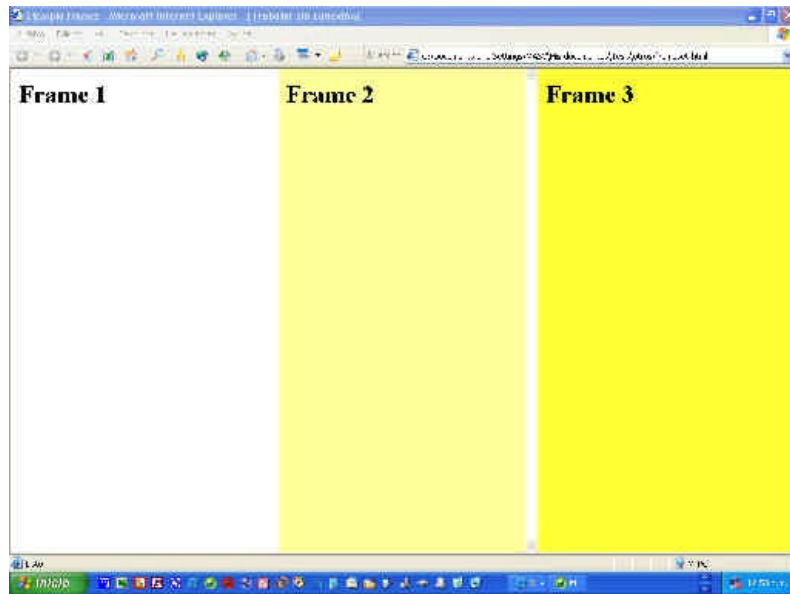
En la página principal (frameset.html) se le indica al *browser* que va a abrir un arreglo de *frames* que van a estar distribuidos en 3 renglones que ocupan aproximadamente la tercera parte de la ventana. También se le indica que los *frames* no pueden ser cambiados de tamaño y que no existe un espacio entre ellos. Lo único que delimita a un *frame* de otro es la barra para recorrer la ventana (scrolling bar). Todo esto se indica en la línea:

```
<frameset rows="34%,33%,33%" frameborder=0 framespacing=0 border=0 noresize>
```

Las siguientes 3 líneas indican qué páginas van a ser abiertas en cada uno de esos *frames* y cómo se va a llamar cada *frame*. También se indica si se va a poder recorrer la ventana en caso de que la información que se muestra allí, ocupe más espacio del asignado.

Para abrir lo 3 frames como columnas en vez de renglones, la línea cambia de:

```
<frameset rows="34%,33%,33%" frameborder=0 framespacing=0 border=0 noresize>
a, <frameset cols="34%,33%,33%" frameborder=0 framespacing=0 border=0 noresize>
```



Para combinar renglones y columnas se escribe lo siguiente:

```
<html>
<head>
<title>Ejemplo frames</title>
</head>
<frameset rows="*,80" cols="*" frameborder="NO" border="0" framespacing="0">
  <frameset cols="150,*" frameborder="NO" border="0" framespacing="0">
    <frame src="frame2.html" name="izquierda" scrolling="NO" noresize>
    <frame src="frame1.html" name="derecha">
  </frameset>
  <frame src="frame3.html" name="inferior" scrolling="NO" noresize >
</frameset>
</html>
```

Ahora, si se quiere poner una liga en el frame 2 que abra una página en el frame 1, se tiene que poner en el tag de la liga:

```
<A HREF="paginax.html" target="derecha">Liga</A>
```

### Formas

Las formas, son aquellos documentos en donde el usuario proporciona datos para ser registrados y usados posteriormente por el sitio.

La sintaxis de las formas es:

```
<FORM METHOD=POST NAME="NombreDeLaForma" ACTION="cgi-bin/archivo.pl">
<!-- Elementos de la forma--> </FORM>
```

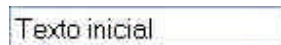
Es importante dar nombre a las formas, sobre todo si se va a usar javascripts. El atributo ACTION indica el lugar a donde toda la información se dirigirá. El archivo que maneja esta información debe estar en un subdirectorio llamado cgi-bin (con minúsculas) y es un archivo programado en perl, C o web-sql, que va a organizar la información.

Los elementos de las formas pueden ser los siguientes:

1) Cuadro en donde el usuario introduzca texto:

```
<INPUT TYPE=TEXT NAME="miTexto" SIZE=30 MAXLENGHT=40 VALUE="Texto inicial">
```

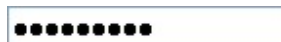
Con esto aparecerá un rectángulo del tamaño de 30 caracteres y que soporta una cantidad máxima de 40 caracteres que serán almacenados en la variable de nombre miTexto. Al momento de ser cargada la página y antes de que el usuario escriba algo en ella, dentro del rectángulo aparecerá: Texto inicial, que será eliminado en cuanto el visitante borre dicho texto.



2) Cuadro de password:

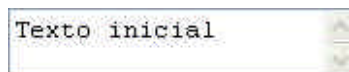
```
<INPUT TYPE=PASSWORD NAME="miPagina" SIZE=8 MAXLENGHT=8 VALUE="">
```

Aquí aparecerá un rectángulo de 8 caracteres de largo y que soporta sólo esa cantidad de caracteres. El valor de este campo será almacenado en la variable miPassword. El contenido del rectángulo aparece con asteriscos (\*).



3) Para tener un espacio más grande se coloca:

```
<TEXTAREA NAME=miTextoMasGRande COLS=40 ROWS=5> Texto inicial  
</TEXTAREA>
```



4) Para elegir opciones se usa:

```
<INPUT TYPE=CHECKBOX NAME="cuadro1" VALUE="1">opción 1<BR>  
<INPUT TYPE=CHECKBOX NAME="cuadro2" VALUE="2">opción 2<BR>  
<INPUT TYPE=CHECKBOX NAME="cuadro3" VALUE="3">opción 3<BR>  
<INPUT TYPE=CHECKBOX NAME="cuadro4" VALUE="4">opción 4<BR>  
<INPUT TYPE=CHECKBOX NAME="cuadro5" VALUE="5">opción 5<BR>
```

opción 1  
 opción 2  
 opción 3  
 opción 4  
 opción 5

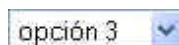
5) Existe la posibilidad de hacer lo mismo que en el punto anterior, pero de manera que el usuario sólo pueda escoger una sola opción. Un ejemplo de esto sería:

Opción 1  
 Opción 2  
 Opción 3  
 Opción 4  
 Opción 5

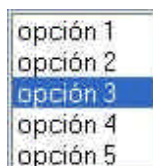
```
<INPUT TYPE=RADIO NAME="cuadro1" CHECKED VALUE="1">Opción 1<BR>  
<INPUT TYPE=RADIO NAME="cuadro2" VALUE="2">Opción 2<BR>  
<INPUT TYPE=RADIO NAME="cuadro3" VALUE="3">Opción 3<BR>  
<INPUT TYPE=RADIO NAME="cuadro4" VALUE="4">Opción 4<BR>  
<INPUT TYPE=RADIO NAME="cuadro5" VALUE="5">Opción 5<BR>
```

6) Otra manera de hacer esto es con:

```
<SELECT NAME ="lista">  
<OPTION VALUE="1">opción 1  
<OPTION VALUE="2">opción 2  
<OPTION SELECTED VALUE="3">opción 3  
<OPTION VALUE="4">opción 4  
<OPTION VALUE="5">opción 5  
</SELECT>
```

A screenshot of a web browser showing a dropdown menu. The menu is open, and the text 'opción 3' is visible and highlighted in blue. A small downward-pointing arrow is visible on the right side of the menu box.

Otra presentación se logra poniendo el atributo SIZE como se indica a continuación.

A screenshot of a web browser showing a list box. The list box contains five options: 'opción 1', 'opción 2', 'opción 3', 'opción 4', and 'opción 5'. The option 'opción 3' is highlighted in blue, indicating it is the selected option.

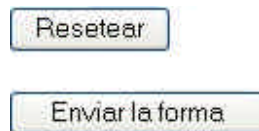
7) Para colocar un botón:

```
<INPUT TYPE=BUTTON VALUE="Botón">
```



8) Enviar la forma a través de la red y resetear

```
<INPUT TYPE=RESET VALUE="Resetear">  
<INPUT TYPE=SUBMIT VALUE="Enviar la forma">
```



### 3.2.1.3 XML

Extensible Mark-Up Language o XML es un metalenguaje que permite definir lenguajes de marcado; es un subconjunto simplificado del estándar SGML.

XML es una respuesta a la necesidad de mayor versatilidad en Internet. XML no es un nuevo lenguaje similar o mejor que HTML. Es una versión de SGML, más sencilla y más fácil de aplicar, con una mayor compatibilidad y adaptabilidad a las nuevas tecnologías y necesidades en Internet.

En XML según las necesidades del usuario o del proyecto a emprender se pueden crear lenguajes exclusivos para el formato de documentos, utilizando sus propias etiquetas, únicamente con el apego al estándar.

Con aplicaciones creadas del SGML como lo es el HTML sólo se especificaba información sobre la estructura del contenido, pero con XML se permite especificar elementos que resultan valiosos en el momento de recuperar información, pues identifican el tipo de contenido.

Por ejemplo, al crear un documento como el siguiente,

```
<autor>  
<nombre>Miguel Ángel</nombre>  
<apellidos>Sánchez Villegas</apellidos>  
</autor>
```

se podría delimitar búsquedas al indicar en que etiquetas queremos buscar, si en <apellidos> o en <nombre>.



Lo ideal es que existan estándares generales especializados en aplicaciones; por ejemplo el *DocBook* es un lenguaje para el formato de libros electrónicos.

### 3.2.1.4 XHTML

Extensible Hyper Text Mark-Up Language es una aplicación de XML. Es un lenguaje similar al HTML, pero con base en las reglas de XML. Las capacidades de XHTML son las mismas que el HTML, pero la diferencia que existe es que las reglas son más estrictas. En XHTML todos elementos necesitan una etiqueta inicial y una final. En HTML están permitidas etiquetas como `<br>` que representa un salto de línea, pero no es necesaria la etiqueta para cerrar, esto no es correcto en XHTML. Otra diferencia importante es escribir todas las etiquetas obligatoriamente en minúsculas; estas son diferencias formales que obligan a realizar documentos mejor formados que HTML.

Esto no quiere decir que se tenga que dejar de usar el HTML y emigrar todo a XHTML, simplemente XHTML fue en el fondo la forma que tuvo el W3C de mostrar las bondades y ventajas de la implementación de XML.

## 3.3 Bases de datos

La tecnología que provee los medios para almacenar, administrar y acceder a grandes cantidades de información son las bases de datos. Los antecedentes de estas se remontan a 1890 cuando Herman Hollerith mecanizó el almacenamiento del censo de EE.UU. en lo que se considera la primera base de datos significativa computarizada<sup>87</sup>.

La utilización de las bases de datos como medio de almacenamiento y recuperación de la información, reporta ventajas importantes como la eliminación de información redundante e inconsistente, permite compartir información, varios sistemas o usuarios pueden utilizar una misma entidad y logra la independencia de los datos.

Las bases de datos facilitan a organizaciones gubernamentales, instituciones académicas, e unidades de información, poner a disposición de los usuarios desde documentos de texto en lenguaje natural, catálogos, tablas estadísticas, datos financieros y objetos multimedia, hasta datos de naturaleza técnica y científica.

Los diseñadores y administradores de bases de datos enfrentan desafíos que reflejan la complejidad del floreciente entorno de la información. Las tecnologías de bases de datos deben manejar masivas cantidades de datos, extraer información útil desde estos repositorios, y tener la habilidad para reflejar las relaciones entre los datos mantenidos en diferentes bases de datos. Además de la arquitectura y sistema deben proveer integridad, recuperación, y seguridad.

---

<sup>87</sup> SCHNEIDER, G.M. *An invitation to computer science*. — 2ª ed. — EE.UU. : PWS, 1998. — 255 p.

Existen fundamentalmente tres estilos de diseño de bases de datos: el modelo jerárquico, el modelo de red y el modelo relacional.

En el modelo jerárquico la forma de esquematizar la información se realiza a través de representaciones jerárquicas, de manera similar a la estructura de un árbol. El modelo de red evita la redundancia en la información a través de la incorporación de un tipo de registro denominado conector. En el modelo relacional toda la información se representa a través de arreglos bidimensionales o tablas, utilizando operaciones como seleccionar renglones y columnas o unir información de varias tablas.

Al utilizar bases de datos para otorgar algún servicio de información, es necesario realizar un análisis de los modelos existentes y elegir el que mejor se adapte a las necesidades y al tipo de información. Un esquema de datos y una arquitectura bien definida aseguran un almacenamiento de datos lógico y eficiente, lo cual incrementa la capacidad de la base de datos y extiende las capacidades de los lenguajes de consulta y otros métodos de acceso.