



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Transformación dinámica de estructuras
de mercado para la generación de ediciones
digitales**

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

BAUTISTA LARA GUMERCINDO VICTOR

PÉREZ GARCÍA RAÚL



DIRECTORA DE TESIS:
ING. ARELLANO MENDOZA LUCILA PATRICIA

CIUDAD UNIVERSITARIA - MÉXICO A 23 DE ENERO DE 2008



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

*A mi padre[†], por siempre
a mi hermano
a mi compañera de vida
a mis profesores
y a mis amigos*

Victor

En Memoria:

*Los recuerdo como el mayor regalo que me dio la vida,
siempre los amaré: a mis padres.*

Estas en mi corazón, siempre te recordaré: Paty.

Por estar siempre a mi lado: a mis hermanos.

Por tu amor, por haberte conocido: Nidia.

Por cada momento compartido: a mis amigos.

Raúl

Índice

Introducción	15
Capítulo 1 - Antecedentes	17
1.1 Conceptos Básicos	17
1.1.1 Evolución de la World Wide Web	17
1.1.2 Tecnologías actuales de la WEB	18
1.1.3 Sistemas de hipermedia – hipertexto	19
1.1.4 Bibliotecas Digitales	20
1.2 Ediciones digitales	22
1.2.1 La tecnología en edición de libros	22
1.2.2 El libro electrónico	23
1.2.3 La edición electrónica o digital	24
1.2.4 Derechos de Autor	26
1.2.5 Ventajas y temores ante la digitalización	27
1.3 Metalenguaje	28
1.3.1 SGML	28
1.4 Lenguajes de programación para procesamiento de marcas	31
1.4.1 Lenguajes de marcas descriptivas y su procesamiento	31
1.4.2 Elementos para el procesamiento de marcas	32
1.4.3 Aplicaciones del procesamiento de marcas descriptivas	34
BIBLIOGRAFÍA DEL CAPÍTULO	36

Capítulo 2 - Información estructurada y su procesamiento	37
2.1 Información estructurada y metadatos	37
2.1.1 Información estructurada	37
2.1.2 Metadatos	38
2.1.3 Propuestas e iniciativas de metadatos: formatos, esquemas o estándares relacionados a ediciones digitales	39
2.2 Análisis de las herramientas a utilizar	40
2.2.1 XML, DTD, TEI y Hojas de estilo	40
2.3 Documentos XML	41
2.3.1 DTD	41
2.3.2 TEI	43
2.3.3 Hojas de estilo	45
2.4 Lenguajes de programación PHP Y JAVA	47
2.4.1 Lenguaje de programación PHP	47
2.4.1 Lenguaje de programación JAVA	47
2.5 Motores de transformación dinámica	48
2.6 Propuesta de solución	49
2.6.1 Análisis de información obtenida	49
2.7 Proceso de integración y solución propuesta	51
BIBLIOGRAFÍA DEL CAPÍTULO	54

Capítulo 3 - Análisis y diseño	55
3.1 Requerimientos de publicación	55
3.1.1 Descripción de requerimientos gráficos	55
3.1.2 Descripción de elementos Editoriales	60
3.2. Uso de TEI	61
3.2.1 Estructuración de la información	61
3.2.1.1 Elementos de división textual	62
3.2.1.2 Encabezados y cierres	64
3.2.1.3 Prosa y verso	64
3.2.1.4 Numeración de página y línea	66
3.2.1.5 Marcar frases enfatizadas	68
3.2.1.6 Citas y características similares	70
3.2.1.7 Notas	71
3.2.1.8 Listas	72
3.2.1.9 Cita bibliográfica	73
3.2.1.10 Tablas	75
3.2.1.11 Imágenes y gráficos	76
3.3 Implantación	77
3.4 Adaptación de DTD	79
3.4.1 Descripción de elementos	79
3.4.2 Aplicación	84
3.5 Estructuras prototipo	86

3.5.1	Diseño de estructura prototipo	88
3.6	Programación de hojas de estilo	90
3.6.1	Codificación dinámica en plantillas XSLT prototipos	90
3.6.2	Codificación estática de estilo en CSS	95
3.7	Módulo de transformación	95
3.7.1	Servidor Web: Tomcat	96
3.7.2	Lenguajes de soporte para el desarrollo del modulo	98
3.7.2.1	Motor de procesamiento del módulo	98
3.7.2.2	Paquetes de XSLT	99
3.7.3	Procesamiento XSLT con Java	100
	BIBLIOGRAFIA DEL CAPÍTULO	102
	Capítulo 4 - Transformación	103
4.1	Configuración de elementos de transformación	103
4.1.1	Instalación y configuración	103
4.1.2	Núcleo de transformación	105
4.2	Uso del módulo de transformación	106
4.2.1	Aplicación del módulo de transformación a recursos	106
4.3	Elementos interdisciplinarios para la transformación	108
4.3.1	Documento XML	108
4.3.2	Procesamiento de imágenes	109
4.3.3	Uso de hojas de estilo	111

4.4 Revisión de los elementos generados de la transformación	111
4.5 Validación de los productos generados	113
4.5.1 verificación del reporte generados	113
BIBLIOGRAFÍA DEL CAPÍTULO	116
Capítulo 5 - Publicación de edición digital	117
5.1 Implantación en caso de estudio Enciclomedia	117
5.1.1 Uso de Moprosoft	117
5.1.2 Antecedentes de Moprosoft	118
5.1.3 Análisis específico de implantación	120
5.2 Verificación, validación de código y diagrama de flujo	122
5.3 Enciclomedia y la edición digital	123
5.4 Estructura de elementos a publicar	125
5.4.1 Análisis del producto generado para su publicación	125
5.4.2 Requisitos del producto generado para la publicación	126
5.5 Publicación digital	127
BIBLIOGRAFÍA DEL CAPÍTULO	128
Conclusiones	129
Anexos	133

Introducción

La aparición de tecnologías para la creación de documentos digitales en diversos formatos ha permitido dar un gran salto en el desarrollo de los soportes de información. Y esto, a su vez, ha propiciado nuevas formas de almacenamiento, organización, edición, transferencia y acceso a la información.

Actualmente los documentos digitales¹ han adoptado mecanismos de estandarización que permiten su mejor y mayor aprovechamiento. Este proceso ha tomado tiempo: comenzó con la digitalización de libros impresos, en primer lugar aquellos que se consideraron de gran valor histórico; se trataba de *escanearlos* y de presentarlos como colecciones de imágenes ligadas mediante hipertexto² simulando hojear el libro; después, se mantuvo la práctica de *escanear* los documentos, pero mediante la tecnología OCR (*Optical Character Recognition*; Reconocimiento Óptico de Caracteres) con el fin de convertirlos en texto de formato digital; más adelante, se impulsó la creación de documentos empleando procesadores de texto con el objetivo de producir archivos impresos originados en una computadora.

Hoy en día, existen documentos con distintos formatos y cualidades, que nacen y se desarrollan únicamente en el ámbito digital. Los “metalenguajes” han jugado un importante papel en dicho ámbito, ya que sirven para crear, a su vez, aquellos lenguajes que codifican elementos que hacen posible: la identificación de documentos digitales, su manipulación con distintas aplicaciones y su intercambio en plataformas diferentes (Linux, Windows, Mac OS X, etc.); lo que también ha llevado a buscar nuevas propuestas para su organización, conservación y difusión, en las que se requiere que la cantidad no sea un obstáculo en el momento de su recuperación.

Con estos antecedentes, surgen las bibliotecas digitales.³ Al mismo tiempo, ellas han propiciado cambios en los procesos y estructuras tradicionales (un ejemplo, la forma de adquirir el acervo), así como en el comportamiento de los usuarios y en sus demandas de información, lo que obliga naturalmente a ofrecer servicios más complejos (no sólo contar con un acervo más completo o especializado, sino incluir el uso de la amplia gama de herramientas de las que se puede echar mano en el ámbito digital).⁴

Los proyectos con los avances antes mencionados requieren del uso de este soporte de tecnología, que a su vez requiere tener un grupo interdisciplinario que pueda generar un proceso eficiente y que permita enriquecer los aspectos evolutivos de la tecnología en materia editorial, esto permite generar estrategias sólidas que incorporen los diferentes estándares tanto editoriales como tecnológicos.

Los proyectos con estas características, son importantes para la creación, difusión, utilización y obtención de nueva información.⁵ Por lo anterior, podemos decir que las Ediciones Digitales contribuyen a popularizar el conocimiento toda vez que más personas

pueden tener acceso a él, lo que facilita la expansión de la cultura. El presente trabajo explica el proceso que apoya esta labor.

1 Entiéndase por documento digital todo aquel soporte electrónico (eléctrico, magnético u óptico) que contenga formas objetivadas del conocimiento y éstas sólo se compongan o se codifiquen por dos valores.

2 Texto que contiene elementos a partir de los cuales se puede acceder a otra información.

3 Colecciones organizadas de documentos digitales, almacenados y puestos a disposición de una comunidad con acceso a Internet a las que se agregan servicios tradicionales y recursos interactivos que la tecnología proporciona para la búsqueda y recuperación de la información.

4 CARRIÓN Rodríguez, Guadalupe. "El porqué de la importancia de la gestión de información", pp. 77-84 Ponencia presentada en las XXX Jornadas Mexicanas de Biblioteconomía, mayo de 1999, Morelia, Michoacán.

5 TRAMULLAS SAZ, Jesús, Dr. Propuestas de concepto y definición de la biblioteca digital. Disponible en Internet: <http://mariachi.dsic.upv.es/jbidi/jbidi2002/Camera-ready/SesionI/SI-I.pdf>.

Capítulo I

Antecedentes

En este capítulo se presentan los fundamentos que sirven como antecedentes al proceso de transformación dinámica, se cubren conceptos básicos, definiciones de ediciones digitales, Bibliotecas Digitales y conceptos tan específicos como son los metalenguajes.

1.1 Conceptos básicos

En el avance constante de los sistemas computacionales destaca la rapidez con la que se vuelve obsoleto un elemento. En esta sección presentaremos elementos de evolución, actuales, y componentes del entorno de creación de las ediciones digitales.

1.1.1 Evolución de la World Wide Web

La evolución de la World Wide Web se ha producido dentro una serie de elementos de diferente índole. Desde el diseño en las interfaces gráficas de las propias páginas Web y de los navegadores, pasando por las funcionalidades de la Web actual que permite transacciones financieras y de e-comercio, múltiples recursos multimedia, la potencia de los buscadores y de las bases de datos, los lenguajes hipertextuales que permiten numerosas aplicaciones por medio de la información estructurada, hasta desembocar en la explosión de la Web como enciclopedia y biblioteca universal en línea tanto para la lectura como para la edición y recuperación de información y cuyo culminación será la Web Semántica. Pero, sobre todo, debemos destacar la hegemonía de la World Wide Web como medio universal de comunicación entre personas y grupos de interés.

La evolución de la World Wide Web se puede observar a través de la múltiple bibliografía existente tanto en línea como fuera de ella, y también en forma gráfica, la World Wide Web es el hipertexto hipermedia por excelencia.

Existen datos que confirman que Internet es el medio de información que más crece y que la información acumulada tanto en la Web navegable como en la Web profunda es inmensa [1]. La Web, el gran hipertexto que reúne documentos y recursos de todo tipo y morfología (texto, imágenes, audio, videos, recursos audiovisuales), y sus utilidades son inmensas y diversas.

I.1.2 Tecnologías actuales de la Web

Las tecnologías actuales permiten que con una programación, a veces sencilla y otras veces un poco más compleja, se puedan disparar procesos electrónicos que efectúen dos pasos a la vez: actualicen la base de datos del producto y a su vez actualicen el sitio Web. En algunas circunstancias, es técnicamente factible (porque la base de datos reside en servidores físicamente cerca al servidor del sitio Web) que las páginas sean servidas directamente de la base de datos de producto, sin embargo hay muchas consideraciones de desempeño de la base de datos y de lo apropiado de la misma para ser colocada directamente en el servidor Web. Es recomendable entonces que cuando la información tenga un alto nivel de variación (alta volatilidad), se establezcan procedimientos automáticos que garanticen que la información estará actualizada en todos los puntos de contacto, interna y externamente.

Algunas tecnologías web representativas:

Navegadores web

- Internet Explorer
- Mozilla Navigator
- Mozilla Firefox
- Netscape Navigator
- Opera

Servidores Web

- Servidor HTTP Apache
- IIS (Internet Information Server)
- Tomcat (Libre, del proyecto Jakarta de Apache)

Otras tecnologías

- CFM Coldfusion
- DHTML
- PHP
- ASP
- CGI
- JSP (Tecnología Java)
- .NET

I.1.3 Sistemas de hipermedia – hipertexto

Un hipermedia es un documento digital que se puede leer, recorrer o navegar de manera no secuencial. La World Wide Web (WWW) es un ejemplo claro de un sistema hipermedia interactivo desarrollado sobre Internet. Para ver la información se utiliza una aplicación llamada navegador Web que extrae elementos de información (llamados "documentos" o "páginas Web") de los servidores Web (o "sitios") y muestra en la pantalla del usuario. El usuario puede entonces seguir hiperenlaces que hay en la página a otros documentos o incluso enviar información al servidor para interactuar con él. A la acción de seguir hiperenlaces se le suele llamar "navegar" por la Web.

El término hipermedia toma su nombre de la combinación de hipertexto y multimedia, una red hipertextual en la que se incluye no sólo texto, sino también otros medios: imágenes, audio, vídeo, etc. (multimedia).

Así pues, la hipermedia combina tanto la tecnología hipertextual, como la multimedia. Si la multimedia proporciona una gran riqueza en los tipos de datos, el hipertexto aporta una estructura que permite que los datos puedan presentarse y explorarse siguiendo distintas secuencias, de acuerdo a las necesidades y preferencias del usuario.

La estructura de un hipermedia es la misma que la de un hipertexto, formado por nodos que se conectan mediante enlaces. La única diferencia es que los nodos de hipermedia contienen elementos de diferentes medios o morfologías. Las anclas ya no sólo son palabras sino que pueden, por ejemplo, ser una imagen o un fragmento de ella, o pueden ser una secuencia de audio o de vídeo. La estructura de un hipermedia es, pues, más compleja que la del hipertexto. La interacción de los diferentes medios y la sincronización entre ellos suele ser uno de los aspectos más complejos en el desarrollo de aplicaciones multimedia.

Si, en teoría, hipermedia es la combinación de hipertexto y multimedia, existirán diferencias entre ambos sistemas. Hipertexto provee una estructura de navegación a través de los datos textuales, mientras que el multimedia nos ofrece no sólo esta estructura de navegación a través de datos textuales, sino a través de una gran variedad de tipos de datos de diferentes morfologías. Si en hipertexto se habla de *datos*, en multimedia es corriente hablar de *componentes*. Los documentos multimedia constan de una colección completa de componentes cada uno de los cuales puede estar compuesto, a su vez, de otros componentes o bien de elementos de datos llamados también *entidades*.

En conclusión, podemos destacar que los sistemas hipermedia, en cuanto a su generación como documentos, son mucho más complejos que los sistemas hipertextuales. En un hipertexto se pueden fragmentar los bloques de texto para ser enlazados, pero en un sistema hipermedia la asociación de un enlace con o dentro de un componente multimedia es mucho más compleja, ya que los datos la mayor parte de las veces no pueden fragmentarse ni indexarse. Además, los sistemas hipermedia pueden incorporar la llamada inteligencia embebida, es decir, son capaces de ejecutar otras aplicaciones o de tomar

decisiones de acuerdo con la actividad que desarrolla el usuario tanto al utilizar los enlaces como al acceder a los contenedores.

Hay que tener presente que no todas las aplicaciones multimedia son hipermedia, ya que pueden ser una simple presentación de pantallas en orden secuencial. No obstante, a menudo se utilizan indistintamente los términos hipermedia y multimedia, ya que se entiende que una buena aplicación multimedia debe ser en realidad hipermedia.

Resumiendo, existen tres conceptos diferentes:

Hipertexto: texto en formato no secuencial, compuesto de nodos y enlaces que los interconectan

Multimedia: unión de diferentes medios o morfologías de la información, como texto, gráficos, audio, video, otros recursos audiovisuales, etc.

Hipermedia: hipertexto + multimedia

1.1.4 Bibliotecas digitales

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

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

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

Biblioteca digital, es un repositorio de acervos y contenidos digitalizados, almacenados en diferentes formatos electrónicos por lo que el original en papel, en caso de existir, pierde supremacía. Generalmente, son bibliotecas pequeñas y especializadas, con colecciones limitadas a sólo algunos temas. En este tipo de biblioteca es en donde concentraremos este trabajo.

Existen diferentes definiciones para una biblioteca digital, en su concepto más simple, se representa como *un espacio en donde la información es almacenada y procesada en formato digital*.

Las bibliotecas digitales son organizaciones que proveen los recursos, incluyendo personal especializado, para seleccionar, estructurar, distribuir, controlar el acceso, conservar la integridad y asegurar la persistencia a través del tiempo de colecciones de trabajos digitales que estén fácil y económicamente disponibles para usarse por una comunidad definida o para un conjunto de comunidades. [2]

Funcionalmente, una biblioteca digital es una colección organizada de documentos almacenados en formato digital que a su vez ofrece los servicios de búsqueda y recuperación de información. Los documentos que se encuentran en una biblioteca digital pueden ser texto, imágenes, sonido, video o combinaciones de cualquiera de estas. Idealmente se debe de almacenar y recuperar documentos completos, y las búsquedas se realizan sobre el contenido completo de los documentos. Es decir, si el documento es texto, es posible realizar la búsqueda sobre cada palabra incluida en el documento completo y una vez localizado es posible obtenerlo de manera inmediata. En el caso de otros formatos como audio y video, la búsqueda se realiza sobre las fichas que definen las características de cada material.

Las bases de datos y la información pueden ser almacenadas de manera centralizada en caso de colecciones pequeñas. Aunque lo más común es contar con información distribuida, en este caso, las búsquedas y las consultas de un sistema a otro deberán ser transparentes para el usuario.

La interfaz por excelencia es el Web, ya que ofrece una plataforma abierta, principalmente para proporcionar el acceso a los usuarios, aunque también es utilizada por los administradores de la biblioteca y por los grupos encargados de generar contenidos.

Las bibliotecas digitales proporcionan sus servicios a través de alguna red de cómputo, ya sea una red sólo de uso local o tan abierta como Internet, esto dependerá de los servicios y las restricciones que deban hacerse para el acceso a la información.

En una biblioteca digital se involucran muchos grupos de trabajo. Es necesario contar con una logística de digitalización para diferentes casos, como el de las publicaciones que no pueden salir de su lugar de resguardo o de las obras que deben tener algún tratamiento especializado en su manejo, así mismo se necesita de programadores, bibliotecarios, editores, etc.; más adelante hablaremos con detalle de cada parte.

1.2 Ediciones Digitales

A lo largo de la historia el avance tecnológico ha originado diversas transformaciones en las áreas del trabajo humano. La actual masificación tecnológica –representada, entre otras cosas, por el uso de la computadora – ha influido en la metamorfosis del trabajo en muchos sectores y el editorial es uno de ellos. Se puede decir que nos encontramos en una situación similar a la que experimentaron los lectores de otros siglos al tener que cambiar del rollo de papiro al libro encuadernado, del libro manuscrito al impreso.

A la par de las nuevas tecnologías, nacen otras formas de concebir y escribir un libro. Ahora la pantalla electrónica ofrece una gama de dimensiones que permite manipular los textos a voluntad. El placer y beneficio de la lectura en papel es un acto que aún tardará en ser totalmente desplazado, a pesar de que las nuevas generaciones se adaptan mejor a los formatos electrónicos y contrario a lo que se piensa cada vez se acercan más a la lectura en la pantalla de una computadora.

La tecnología ha modificado la cultura que rodea al libro, las librerías, industria editorial, bibliotecas y difusión de libro. Simplifica las tareas, disminuye distancias y economiza costos y tiempo. Cada uno de estos temas nos lleva a múltiples reflexiones y, aunque es muy tentador especular sobre los cambios sociológicos que enfrentaremos, en este texto abordaremos solamente la manera cómo han irrumpido las nuevas tecnologías en el trabajo de edición y propone algunas recomendaciones de cómo ha de ser la edición electrónica de libros en línea, con el fin de ofrecer al lector mayor legibilidad al leer en pantalla.

1.2.1 La tecnología en edición de libros

Los avances tecnológicos han contribuido a renovar, por un lado, el método tradicional de edición y, por otro, han generado un nuevo concepto de trabajo editorial destinado a la edición electrónica. Es decir, para la manufactura de libros en papel existe un trabajo editorial basado en métodos tradicionales, pero realizado con la tecnología moderna. En cambio para los libros en disco compacto o en red se están conformando normas que, en algunos aspectos, recuperan parcialmente el trabajo tradicional de las artes gráficas.

En el trabajo editorial los aportes tecnológicos agilizan el proceso de publicación: escritura, captura, corrección, composición tipográfica, diseño, formación e impresión. Desde que el autor redacta su texto en la computadora está ahorrando una fase del trabajo editorial que era la captura. La máquina también permite llevar a cabo la corrección de estilo en pantalla y realizar comparaciones entre las distintas versiones de un escrito, todo apoyado con diccionarios y textos especializados que se cargan al software.

En cuanto a la composición tipográfica, los nuevos programas han eliminado la necesidad de la fotocomposición. Al contar con un documento previamente capturado y corregido, han disminuido los márgenes de error en la formación y, sobre todo, se erradica cada vez

más el fantasma de las erratas, los empastelados, letras rotas, líneas repetidas y demás monstruos tipográficos.

Los sistemas de impresión también se han beneficiado con la tecnología que permite la tirada directa desde archivos ubicados en la computadora. Ahora es posible imprimir libros al momento, proceso conocido como *impresión por demanda*, a menores costos y mejor calidad, puesto que cada libro es un original. Esta modalidad beneficia fundamentalmente a los trabajos académicos con temas especializados y a editoriales que manejan a autores cuya obra tiene poca demanda.

Como en cualquier oficio, se requieren conocimientos que eviten la improvisación y, por ende, los malos resultados. Por desgracia, muchos se han lanzado a editar libros atribuyendo a la computadora posibilidades mágicas, ignorando el conocimiento legado por los tipógrafos de antaño y que ahora son desconocidos por muchos de quienes aplican las nuevas tecnologías al trabajo editorial.

1.2.2 El libro electrónico

Respecto de los libros que se editan en soporte electrónico, para ser leídos en una pantalla, existen dos tipos: los que se publican en Internet y los editados en discos compactos. Ambos tipos se conocen como e-books o libros electrónicos, y comúnmente constituyen una nueva versión de la obra que había sido impresa en papel.

Los CD's también presentan sus variantes: a) aquellos concebidos para trabajar como sistemas multimedia o interactivos, que facilita la consulta o la hacen más atractiva –que básicamente incluyen atlas, diccionarios y enciclopedias–, y b) los que constituyen textos extensos en donde no se requiere tanto de la interactividad, conformados generalmente por estudios monográficos, obras literarias y trabajos académicos.

Entre los libros editados en páginas electrónicas también encontramos dos tipos:

a) aquellos que han sido llevados del papel a la pantalla, en cuyo caso se digitaliza un texto, se "sube" a una página electrónica y se le destina un uso específico, ya sea que sólo se pueda leer, se pueda comprar vía red o se pueda bajar a un disco duro. Aquí se incluyen clásicos de la literatura o de la ciencia política que ya se pueden consultar gratuitamente, como son las obras de Shakespeare, John Milton, Miguel de Cervantes, Dante, Esquilo, Maquiavelo, Rousseau, etcétera.

b) El otro tipo de libros en páginas electrónicas son aquellos escritos especialmente para la red. Éstos comprenden, en su mayoría, obras de autores que aprovechan el medio para promoverse sin demasiados costos [3].

También se incluyen aquí las cibernovelas donde los visitantes participan en la modificación de la trama, personajes y final de las historias, creando así una nueva modalidad de interacción en la lectura de un mismo texto. Por otro lado, ciertos autores reconocidos están valorando la importancia de estos medios para la venta de sus obras: Stephen King

tiene varios títulos dispuestos en línea donde se cobra por los derechos de lectura; sin embargo, el lector no puede imprimirla en papel ni bajarla al disco duro de su computadora. Simplemente se le vende la oportunidad de leerla en pantalla durante cierto tiempo.

Las nuevas tecnologías buscan cada día más emular el formato del libro tradicional, sólo que ahora con mayores posibilidades de almacenamiento de información en un mismo sistema de reproducción. Actualmente existen en el mercado máquinas que cuestan alrededor de 300 dólares que reproducen libros electrónicos en formatos fijos para ser leídos en pequeñas pantallas portátiles que imitan el proceso de lectura de un libro en papel: desde la forma de pasar las hojas, hasta la posibilidad de subrayar el texto mediante una pluma digital e incluso realizar anotaciones en la pantalla, para posteriormente recuperar la información en forma organizada.

1.2.3 La edición electrónica o digital

El proceso de edición de un libro para ser leído en medios digitales se conoce como *edición electrónica o digital*. Constituye un conjunto de procedimientos y pasos previos para la publicación de un texto en cd o en la red. Aún hay que definir muchos conceptos relacionados con la edición electrónica de libros.

Por ejemplo, aunque el término "libro" no se adecua a los nuevos sistemas –en el sentido de que se le define como el "conjunto de muchas hojas de papel, impresas y encuadernadas juntas, con una cubierta de papel, cartón, piel o madera"-, se sigue empleando dicho término para designar a las nuevas publicaciones en disco compacto o en red. Lo mismo sucede en materia de derechos de autor y asignación de números ISBN. En muchos campos se sigue trabajando con los conceptos propios de los libros tradicionales.

Ante las necesidades de redefinición, surgen nuevos términos relacionados con los sitios en el ciberespacio, como el de los *portales electrónicos* que son páginas que ofrecen al visitante la entrada a diversos servicios especializados relacionados con un mismo tema.

Los libros en red se encuentran mayormente vinculados a páginas de librerías, como en el caso de Amazon, que ofrece –además de la venta de libros impresos–, la lectura gratuita de publicaciones; o el Círculo de lectores, que posee un extenso catálogo de libros de actualidad en español.

La presentación de acervos bibliográficos es cada vez más común en la red. La tendencia es llevar el documento al usuario sin necesidad de que éste se desplace. Uno de los proyectos más interesantes, referentes a libros de habla hispana, lo representa el Portal Cervantes, que constituye una antesala a la obtención de servicios de información, consulta y venta de libros electrónicos en distintos países de hispano hablantes.

Recomendaciones para la edición electrónica o digital

Basándonos en el conocimiento tradicional de la edición de libros académicos en papel y el acercamiento, aplicación y análisis de los nuevos métodos, a continuación se enuncian algunas recomendaciones que deben ser consideradas por quienes se dedican a elaborar documentos para sistemas digitales:

- La composición en bandera o justificada a la izquierda, cuyas líneas no tienen una longitud rigurosa, es la más adecuada para los textos en libros electrónicos. Esta alineación ayuda al ojo para que no se pierda fácilmente al recorrer y pasar de una línea a otra.
- Las bibliografías pueden presentarse tal como se hace en papel, sólo que en aquellas ordenadas alfabéticamente se recomienda repetir el nombre del autor, para que el lector no tenga que estar regresando a la referencia. La primera línea deberá sobresalir por la izquierda en párrafo francés.
- Para la pantalla, el empleo de blancos se aplica basándose en algunos principios fundamentales con el fin de dar mayor comodidad de la lectura. En los libros electrónicos el campo la distribución de los blancos es variable, pero los editores de texto permiten modificar ciertos espacios de acuerdo con un diseño predeterminado que guarde el equilibrio visual en pantalla.
- Algunos elementos tipográficos siguen siendo útiles como: sangrías, coroneles, blancos entre párrafos, etcétera. Otros, como medianiles, líneas viudas o huérfanos, ya no son motivo de preocupación.
- En la composición de textos de carácter académico, se está eliminando el uso de números volados y se opta por incluirlos entre paréntesis. La ventaja es que – mediante un código html– al posicionar el *mouse* en la zona de número aparece un marco desplegable que presenta la información correspondiente, sin necesidad de trasladarse hasta el final del documento.
- Los diseñadores prefieren para estas páginas el uso de las familias Times o Arial, sin considerar otras posibilidades. Aunque los editores están predeterminados con la Times de 12 puntos, es recomendable emplear otros tipos como son las familias con patines, con tamaños menores a 12 puntos. Algunos optan por insertar los textos sin mayor codificación para que se adapte a la personalización que el lector dé a su pantalla. El editor debiera incluir recomendaciones al lector al inicio del texto para que ajuste el tamaño del visualizador y de la tipografía. Este texto propone que – mediante el visualizador de páginas Netscape–, el lector haga su propio ajuste para leer cómodamente un libro electrónico, publicado en Internet, de la siguiente forma: en el menú Edición>preferencias>apariencia>tipos de letra anchura variable elija cualquier familia tipográfica con patines como Garamond, Book Antigua, Bookman Old Style o Times New Roman, en 12 puntos. Existe otra forma sencilla de aumentar el tamaño de su tipografía en la pantalla: en el menú Ver>aumentar el tamaño de la letra, se pulsan las veces necesarias hasta lograr el tamaño deseado, siempre y cuando las líneas se vean completas en la pantalla.
- Los frisos o viñetas –tan utilizados en los siglos XVII y XVIII–, tienden a reaparecer cada vez más al principiar un capítulo en los libros electrónicos. Solamente que ya no tiene mucho sentido saber que no deben sobrepasar la octava parte de la altura de la

página. Lo mismo sucede con las filigranas o marcas de agua que estuvieron muy de moda a finales del XIX.

- Los índices ya no refieren a una página sino a todo el texto mediante ligas o links.
- Aunque no existe propiamente una caja de composición, sí hay que marcar retículas para saber dónde y qué tamaño ocuparán algunos elementos; no hay límite para las medidas de alto de la composición, pero sí lo hay para el ancho de ella.
- Es recomendable usar de 10 a 12 palabras por línea, incluso 15 en textos de una sola columna. Para dos o más columnas, el promedio ideal es de 5 a 8 palabras por línea.

1.2.4 Derechos de Autor

En 1984, la Unión Internacional de Editores celebró en la ciudad de México su vigésimosegundo congreso. Al reconocer los complejos cambios que se estaban produciendo en el terreno de las nuevas tecnologías, ahí se sugirió a los editores la promoción de una legislación adecuada para la protección del derecho de autor, tanto en el caso de soporte impreso como en el de materiales magnéticos o electrónicos, así como los originados por cualquier innovación tecnológica [4].

En México, el interés de los editores tuvo respuesta trece años después, en 1997, cuando se modificó la ley del Derecho de Autor, que especifica la protección de obras de todo tipo, cualquiera sea el medio por el que se publiquen.

Aunque la ley no detalla la obligación de asignar seriales de ISBN para libros en red o ISSN para publicaciones periódicas electrónicas; pareciera que la misma ley de derechos de autor aplica a todas las obras en soporte electrónico y en internet.

A pesar de que en este tipo de ediciones han desaparecido dos elementos que por obligación deben incluirse en los libros impresos en papel -el colofón y la página legal-, a los libros y las revistas electrónicas se les deberían asignar esos seriales. No obstante, el hecho de que no exista información acerca de los derechos de autor en una publicación en línea no significa que no esté protegida.

En EUA los legisladores han dado un paso más. En octubre de 1998, el congreso estadounidense aprobó un proyecto de ley, denominado Digital Millenium Copyright que incluye algunas disposiciones para dar respuesta a dos tratados firmados en 1996 con la Organización Mundial de Propiedad Intelectual. Contiene protección contra la piratería, consistente en ciertos algoritmos encriptados diseñados para evitar que se pueda "bajar" un texto, música, video o imágenes que no hayan sido previamente pagados.

Califica como delito distribuir o vender cualquier mecanismo que pueda ser utilizado para burlar los sistemas de protección de derechos de autor [5].

No existe una ley internacional que cubra el uso de textos en Internet, sino que se aplican las leyes de cada país. Por ejemplo, si un texto francés encontrado en Internet es usado en México, estará protegido por las leyes mexicanas.

El problema real es que no es posible saber en dónde se usan, por no decir "pirateando", los textos. Lo que sí importa es que los autores sepan que al ceder sus derechos para la publicación de su texto en un libro impreso no conlleva el derecho del editor de publicarlo en soporte electrónico. Alguien puede ceder el derecho de publicar una obra en papel a un editor y dar el derecho a publicar un CD a otro editor distinto.

Relacionado con los derechos de autor, está el tema de las referencias a un libro electrónico dentro de los trabajos académicos, sigue rigiendo la norma de mencionar la fuente de donde se obtiene la información. Es decir, cuando se consulta un libro electrónico, hay que dar la referencia en una nota comenzando con el autor, el título del libro o del artículo que se haya consultado, el editor de la página donde se encuentre el libro y la fecha en que se consultó la página. Sin embargo, este tema es controvertido porque con frecuencia las fuentes de información citadas, por necesidad, son actualizadas constantemente o borradas de las direcciones electrónicas. De ahí que no se le otorgue mucha confiabilidad a las referencias a páginas electrónicas.

Parecería que en cuestión de edición electrónica, la historia ha dado la primera vuelta en reversa y empezamos a regresar al sitio en donde empezó la historia de la escritura y del libro.

Ya no se requiere de medianiles, ni la proporción áurea, formatos cuadrados o tabloides, tipómetro, retícula, pruebas azules, capillas, cromalín, encuadernador, prensista, retracilado, pastas duras, pliegos, lomos, etcétera, en cambio, las páginas electrónicas recuperan los frisos y las filigranas. Para el escritor moderno, la transmisión electrónica elimina la mediación de toda una cadena de agentes que intervenían entre la escritura y el momento de la escritura, y que retardaba la publicación del libro. No obstante, aún es muy temprano para determinar el rumbo que tomará el futuro de las publicaciones.

1.2.5 Ventajas y temores ante la digitalización

Entre las ventajas de la edición digital cabe mencionar la recepción inmediata, la posibilidad de interactividad, la facilidad de búsqueda de elementos textuales, el alcance a un gran número de usuarios, la accesibilidad al trabajo sin conexión una vez descargado el libro virtual, y el ahorro en papel, con sus consiguientes beneficios ecológicos, y de almacenaje.

Por otro lado, un tema discutido tan ampliamente como es el de la mayor facilidad de lectura de las ediciones tradicionales en papel, ofrece otra perspectiva inversa si pensamos en las posibilidades que ofrece la lectura de una obra digital en lo referente a las posibilidades de aumento del tamaño de la fuente, lo que la convierte en un medio óptimo para la gente con dificultades de visión.

Las posibilidades de traducción inmediata o de translación de caracteres alfabéticos convierten a la edición en línea en la excelente para la difusión internacional [4].

El enorme crecimiento que está experimentando la digitalización de libros a través de buscadores, editoriales, universidades, bibliotecas, etc., con la subsiguiente posibilidad de

apertura de estanterías virtuales por parte de los lectores, sin duda facilita la afición por la lectura a un amplio público, en especial a la juventud por ser los más duchos en el manejo de Internet.

También se acerca el conocimiento a un amplio sector de la población mundial que no dispone de los medios suficientes para adquirir ediciones en papel. Desde este punto de vista, la digitalización de libros supone una democratización de la sociedad en general y, también, de la sociedad del conocimiento [5].

1.3 Metalenguaje

La definición aplicada al metalenguaje computacional es muy simple, aparte de citarla es necesario explicar su importancia y aspectos generales.

“Lenguaje utilizado para describir un sistema de lenguaje”

1.3.1 SGML

Cuando empezaban a popularizarse las computadoras personales en los años ochenta IBM ideó un sistema de guía y enlace para sus PC mientras que Macintosh desarrollaba la Intermedia y la Hypercard.

El SGML (Standard Generalized Markup Language) fue publicado por la Organización Internacional de Estándares el 1986 ISO Standard 8879:1986. En español significa Lenguaje Estándar de Marcación General. Se trata de un sistema que organiza y etiqueta documentos. Con él se especifican las reglas de etiquetado aunque no impone ningún conjunto de etiquetas en especial. Derivada de GML (Generalized Markup Lenguaje), y es considerado un estándar internacional, sin pertenencia y abierto, que provee una codificación estándar para la transmisión de documentos entre sistemas de computadoras diferentes; además, tiene un inmenso potencial y campo de aplicación, su gran desventaja, y por lo cual no ha sido adoptado completamente, es su alta complejidad [6].

La industria de la publicación de documentos constituye uno de los principales usuarios del lenguaje SGML. Empleando este lenguaje, se crean y mantienen documentos que luego son llevados a otros formatos finales como HTML, Postscript, RTF, etc.

Cuando se concibe un documento electrónico en SGML se debe tener en cuenta que:

- El material que constituye un documento se puede distribuir en diferentes archivos, tantos como sean necesarios. Estos archivos además pueden estar almacenados en distintas computadoras.
- Un archivo puede contener la portada, otro la introducción, otro una parte de una hoja de cálculo, otro un gráfico, otro un organigrama, otro la bibliografía, etc.

- En SGML, cada uno de estos objetos recibe el nombre de *entidad*. Las entidades se conciben como objetos independientes.
- Las entidades pueden tener cualquier tamaño, haber sido creadas por cualquier programa de software o estar guardadas en cualquier computadora.
- Las entidades pueden estar compartidas por distintos documentos.
- Un documento estará definido en función de la estructura de las entidades que lo conforman.
- En el índice de materias de un documento no se encontrará ninguna referencia a los archivos que contienen las entidades.
- Las entidades se organizan en una estructura lógica, de manera jerarquizada, en la que se definen conceptos como capítulos, tablas y párrafos y que configuran lo que se denomina estructura de los elementos del documento.
- Elementos y entidades pueden coincidir: un elemento lógico como tabla puede ser también una entidad en un archivo hoja de cálculo

En toda comunicación se pueden distinguir dos niveles de información:

1. Por un lado, el conjunto de datos que componen lo que tradicionalmente entendemos por contenido.
2. Y por otro, una información más sutil, que se sobrepone a ese contenido. Esta otra información es la negrita en un libro, el subrayado en un manuscrito, un tono alto de voz en una conversación y se denomina etiquetado (markup)

La función del etiquetado es aportar información que por lo general refleja la estructura jerárquica de un documento, de forma que ayude al lector humano o al ordenador a procesar su contenido.

El lenguaje de etiquetado SGML permite distinguir entre el contenido o datos (compuestos por caracteres de datos, las letras del alfabeto, los números, la puntuación, etc.) y el etiquetado (compuesto por caracteres de etiquetado, los cuales, en este caso concreto, son también letras, números y signos de puntuación).

El procedimiento de etiquetado:

La idea de etiquetar un texto no es nueva. Hasta fechas muy recientes, los maquetadores de las imprentas marcaban los textos con instrucciones para que el cajista supiera cómo reflejar el diseño, esto es, si los títulos debían aparecer más grandes, en negrita o centrados, los renglones sangrados con una cierta anchura, etc.

Estas instrucciones eran una secuencia de signos ininteligibles para el profano y, muchas veces, sólo tenían sentido para la máquina concreta con la que se iba a imprimir.

Las instrucciones podían contener códigos de control específicos que trasladados a otro entorno podían bloquear la composición tipográfica.

Estas instrucciones, que estaban además intercaladas en el texto, imposibilitaban la reutilización posterior de la información.

Si el texto era revisado con intención de volver a editarlo, había que utilizar el mismo sistema de composición, para entonces probablemente obsoleto.

Si se deseaba cambiar el diseño, había que manipular los archivos para modificar todos los códigos, muchos encriptados u ocultos, como por ejemplo, el efecto de centrar un título escrito en negrita con letra Times Roman de 36 puntos.

La utilización de técnicas informáticas de búsqueda y recuperación globales podían resolver parcialmente el problema, pero a veces ocasionando efectos no deseados.

El etiquetado procedimental es esta técnica que estamos describiendo, por medio de la cual un operario utiliza instrucciones crípticas y dependientes del funcionamiento de un sistema determinado para que ejecute una acción, como por ejemplo activar la tipografía, poner en negrita, centrar, etc.

El etiquetado descriptivo por el contrario, identifica los elementos estructurales de un documento, determinando su estructura lógica, de manera que se solucionan muchos de los problemas mencionados.

Ventajas del etiquetado descriptivo

1. En primer lugar, el estándar SGML utiliza un conjunto de caracteres basado en el estándar ASCII (American Standard Coding for the Interchange of Information), reconocido de manera prácticamente universal por cualquier tipo de plataforma y de sistema informático. Los caracteres especiales, que no están contemplados en el conjunto de caracteres ASCII (propios de sistemas de escritura distintos del inglés, símbolos matemáticos, etc.) se transforman en representaciones ASCII y se denominan referencias de entidad. Así se evitan otros caracteres especiales o de control.
2. La segunda mejora tiene que ver con la subordinación del etiquetado a los aspectos lógicos de la estructura de los documentos. SGML parte del criterio de que existe una relación directa entre cuestiones como el cambio de tipografía y una cabecera, la utilización de la cursiva para resaltar un término, el dibujo de un cuadro con un gráfico, etc. En SGML todo el etiquetado es lógico, es decir, en lugar de utilizar códigos crípticos, se utilizan nombres de elementos, delimitados por marcas que indican el comienzo y final de los objetos lógicos [7].

SGML introduce tres conceptos básicos:

- El concepto de lenguaje de marcado generalizado como un metalenguaje que sirve para definir lenguajes concretos que pueden adaptarse a cada dominio mediante una gramática que describe formalmente un tipo específico de documento o DTD (Document Type Definition);

- El concepto de marcado descriptivo¹, frente marcado procedural. El marcado descriptivo describe, mediante las marcas o etiquetas definidas en la DTD, la estructura lógica de la información. La idea clave es que las marcas no determinan el procesamiento del documento de manera fija, ya que dicho procesamiento se determina a partir de las necesidades concretas, y se beneficia de la estructura lógica del documento caracterizada a través de sus marcas;
- El concepto de independencia de la plataforma. Como los documentos SGML únicamente contienen texto, éstos pueden ser procesados en distintas plataformas, trascendiendo el uso de dichos documentos a los sistemas que los crearon y utilizaron originariamente.

1.4 Lenguajes de programación para procesamiento de marcas

Cuando se elige un lenguaje de programación para implantar un sistema que permita procesar un lenguaje de marcas es importante evaluar aspectos relacionados a su integración al lenguaje de marcas que se necesita, lo eficiente que es al procesarlo, y dependiendo de las necesidades del sistema se deberá seleccionar el que cumpla con la mayoría de las necesidades. Mostramos a continuación una descripción de los conceptos en general que nos apoyarán a su selección.

1.4.1 Lenguajes de marcas descriptivas y su procesamiento

El marcado descriptivo o semántico utiliza etiquetas para describir los fragmentos de texto, pero sin especificar cómo deben ser representados, o en que orden. Los lenguajes expresamente diseñados para generar marcado descriptivo son el SGML y el XML.

Las etiquetas pueden utilizarse para añadir al contenido cualquier clase de metadatos. Por ejemplo, el estándar Atom, un lenguaje de sindicación, proporciona un método para marcar la hora "actualizada", que es el dato facilitado por el editor de cuándo ha sido modificada por última vez cierta información. El estándar no especifica como se debe presentar, o siquiera si se debe presentar. El software puede emplear este dato de múltiples maneras, incluyendo algunas no previstas por los diseñadores del estándar.

Una de las virtudes del marcado descriptivo es su flexibilidad: los fragmentos de texto se etiquetan tal como son, y no tal como deben aparecer. Estos fragmentos pueden utilizarse para más usos de los previstos inicialmente. Por ejemplo, los hiperenlaces fueron diseñados en un principio para que un usuario que lee el texto los pulse. Sin embargo, los buscadores los emplean para localizar nuevas páginas con información relacionada, o para evaluar la popularidad de determinado sitio web.

El marcado descriptivo también simplifica la tarea de reformatear un texto, debido a que la información del formato está separada del propio contenido. Por ejemplo, un fragmento indicado como cursiva (*<i>texto</i>*), puede emplearse para marcar énfasis o bien para señalar palabras en otro idioma. Esta ambigüedad, presente en el marcado de presentación

y en el procedimental, no puede soslayarse más que con una tediosa revisión a mano. Sin embargo, si ambos casos se hubieran diferenciado descriptivamente con etiquetas distintas, podrían representarse de manera diferente sin esfuerzo.

El marcado descriptivo está evolucionando hacia el marcado genérico. Los nuevos sistemas de marcado descriptivo estructuran los documentos en árbol, con la posibilidad de añadir referencias cruzadas. Esto permite tratarlos como bases de datos, en las que el propio almacenamiento tiene en cuenta la estructura, no como en los grandes objetos binarios (blobs) como en el pasado. Estos sistemas no tienen un esquema estricto como las bases relacionales, por lo que a menudo se las considera bases semiestructuradas.

1.4.2 Elementos para el procesamiento de marcas

Para los lenguajes de marcas es necesario tener un motor programado para su interpretación. En la actualidad existen diferentes tipos de lenguajes, principalmente orientados a objetos que cumplen este propósito, se mencionarán los más conocidos por su potencia e integración a los sistemas desarrollados actualmente.

El núcleo fundamental para el procesamiento de marcas tiene como antecedente la manipulación de datos de un sistema de información. En la actualidad se han generado estándares para el procesamiento de marcas, entre los más importantes se encuentran los siguientes:

DOM

DOM ("Document Object Model") es solamente una especificación definida por el "World Wide Web Consortium", esto es muy similar al J2EE de Sun, ya que permite a diversas compañías u organizaciones definir analizadores alrededor de esta especificación. La especificación DOM más reciente es 2.0 y la gran mayoría de los "analizadores" DOM disponibles ya cumplen con ella.

Uno de los analizadores de mayor uso es llamado Xerces y es desarrollado por la fundación Apache, Xerces es un "*fully-validating parser*" disponible en los lenguajes Java y C++, inclusive existe un "wrapper" para Perl lo cual permite manipular documentos en XML desde un programa en Perl.

DOM genera un árbol jerárquico en memoria del documento o información en XML. Basándose en el documento anterior cada elemento <Amigos>, <Nombre>, <Apellido> ..etc. es considerado un nodo dentro del árbol. Este árbol jerárquico de información en memoria permite que a través del "parser" (Xerces o algún otro) sea manipulada la información, las ventajas serían las siguientes:

- Puede ser agregado un nodo (Información) en cualquier punto del árbol.
- Puede ser eliminada información de un nodo en cualquier punto del árbol.

- Lo anterior se ejecuta sin incurrir en las limitaciones de manipular un archivo de alguna otra manera.

Debido a que DOM es sólo una especificación existen diversos "analizadores" DOM.

Un detalle importante de cualquier analizador es que la mayoría están escritos en Java , esta no es ninguna coincidencia ya que Java es uno de los lenguajes que permite mayor portabilidad entre Sistemas Operativos. Ahora bien, a pesar de esta portabilidad en Java, DOM es sólo una especificación y por ende existen diversas aplicaciones, esto lleva a otra pregunta clave el, saber si desarrolla una aplicación que utilice el analizador Xerces este programa puede ser utilizado posteriormente con un DOM desarrollado por Oracle?, la respuesta es negativa, por lo cual se presentan interfaces de este modelo para diversos lenguajes de programación.

SAX

SAX ("Simple API for XML") procesa el documento o información en XML de una manera muy diferente a DOM, SAX procesa la información por eventos. A diferencia de DOM que genera un árbol jerárquico en memoria , SAX procesa la información en XML conforme esta sea presentada (evento por evento), efectivamente manipulando cada elemento a un determinado tiempo , sin incurrir en uso excesivo de memoria. Por lo tanto puede notar las siguientes características:

- SAX es un analizador ideal para manipular archivos de gran tamaño, ya que no ocupa espacio para generar un árbol en memoria como es requerido en DOM.
- Es más rápido y sencillo que utilizar DOM
- La sencillez tiene su precio, debido a que SAX funciona por eventos no es posible manipular información una vez procesada, en DOM no existe esta limitación ya que se genera el árbol jerárquico en memoria y es posible regresar a modificar nodos.

La especificación más reciente de SAX es 2.0, y al igual que DOM 2.0 ésta se incluye en casi todos los analizadores disponibles en el mercado.

Debido a que las diversas corrientes que implantan estos modelos se tienen que enfrentar a la diversidad de los dos modelos es común que al implantar un modelo para su lenguaje en automático generen la otra interfaz del modelo. Algunas interfaces que se han creado son para los siguientes lenguajes:

- C++
- Java - W3C Document Object Model Level 2
- Lisp
- Pascal (Kylix)
- Perl
- PHP
- Python

- Ruby
- TCL

Otros aspectos que se tienen que considerar son las herramientas de edición de documentos de marcas así como elementos de estilo para las marcas descriptivas.

I.4.3 Aplicaciones del procesamiento de marcas descriptivas

Las aplicaciones están presentes en cualquier sistema que interprete la información estructurada. Los casos sobresalientes en su aplicación ya sea por innovación o por rendimiento son varios y algunos se mostrarán en esta sección.

Información distribuida

XML/RSS/RDF son términos bastante conocidos por los famosos “*Bloggers*”, personas que tienen su propio *Weblog*². Estas tecnologías tienen como base, documentos XML, en este caso la sindicación de noticias es un nuevo servicio que permite obtener información de un documento XML generado automáticamente por un sistema de publicación, como Movable Type ó B2 (CafeLog). Una de las bondades o servicios de RSS/RDF es poder syndicar las noticias de un sitio en mi sitio, es decir, si un sitio X permite la sindicación de noticias, con un Script en X lenguaje puedo hacer que esas mismas noticias aparezcan en mi Web.

También se puede notar los *FeedReaders* o lectores de noticias RSS, son programas que permiten agregar la URL de un sitio (exactamente, el archivo RSS o RDF que permite la sindicación) y sin necesidad de visitar ese sitio, recibir las novedades y las noticias de este.

Lectores de noticias RSS:

1. <http://www.feedreader.com/>
2. [http://www.feedster.com /](http://www.feedster.com/)
3. <http://www.bloglines.com/>

Algunos rubros sobresalientes de este procesamiento se pueden encontrar en temas como lo son:

1. Desarrollo web
2. Administración de contenidos
3. Documentación
4. Desarrollo de Bases de Datos

Sitios de usos interesantes:

- Aerospace: Spacecraft Markup Language

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

- Banking: Bank Internet Payment System

<http://www.fstc.org/>

- Advertising: AdXML

<http://www.adxml.org/>

- Petroleo: PetroXML

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

<http://es.wikipedia.org/wiki/Web>

<http://www.hipertexto.info/documentos/hipermedia.htm>

- Servicios Web

<http://www.w3.org/2002/ws/desc/>

BIBLIOGRAFÍA DEL CAPÍTULO

- [1] LYMAN, Peter and VARIAN, Hal R. How Much Information?
<http://www2.sims.berkeley.edu/research/projects/how-much-info2003/internet.htm>
Visitado en Mayo 2003
- [2] Digital Libraries Federation. <http://arl.cni.org> Visitado en Mayo de
- [3] <http://www.razonypalabra.org.mx/anteriores/n20/libros.html>
Visitado en Mayo de 2007
- [4] Senso, José A. (2003, Junio 1). El concepto de metadato. Algo más que descripción de recursos electrónicos.: Universidad de Granada. España.
- [5] Méndez Rodríguez, Eva (2002). Metadatos y recuperación de información estándares, problemas y aplicabilidad en bibliotecas digitales.: Trea.
- [6] Publicación electrónica abierta. Disponible en:
<http://www.enterate.unam.mx/Articulos/2003/mayo/publiabierta.htm> Visitado el 14 Mayo de 2007
- [7] SGML. Disponible en: <http://www.serv-inf.deusto.es/ABAITUA/konzeptu/sgml/sgmlI.htm#nueva> Visitado el 11 Mayo de 2007

¹ También se utiliza el término marcado declarativo con este mismo sentido pero, como puede inducir a confusión con los lenguajes de programación declarativos, se ha preferido utilizar el término descriptivo.

² Blog, también conocido como weblog o cuaderno de bitácora (listado de sucesos), es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.

Capítulo 2

Información estructurada y su procesamiento

Una buena práctica en el procesamiento de la información es un correcto manejo de su estructura, y una adecuada integración entre el sistema que la procesa y el esquema elegido de procesamiento. En este capítulo trataremos temas fundamentales para el manejo de la información que nos permitirán crear las bases para el manejo óptimo de la misma.

2.1 Información estructurada y metadatos

Los metadatos o la necesidad de etiquetar, catalogar y describir información estructurada, de tal forma que los documentos digitales se puedan procesar, almacenar, conservar, recuperar e intercambiar a través de Internet, protagonizarán, sin duda, un nuevo paradigma en los sistemas y servicios de información del siglo XXI. Este tema ha despertado un gran interés y debate científico en la comunidad bibliotecaria internacional y en todo el sector emergente de los sistemas de información electrónica, debido a la evidencia de Internet como fuente, depósito y recurso de información, así como a las nuevas formas de registrar el conocimiento que conlleva la explosión de la información digital. Todos estos cambios han superado los métodos tradicionales de procesamiento de la información (catalogación, clasificación e indización), e imponen una redefinición estratégica del papel del gestor de la información ante la biblioteca digital.

2.1.1 Información estructurada

Una distinción tradicional en el mundo de la computación alude a que hay dos tipos de información: la estructurada que encontramos identificada y agrupada bajo algún criterio y la no-estructurada en formatos muy diversos como texto, videos, audio o imágenes.

La información no estructurada (unstructured information), hasta hace poco aparecía en las bases de datos como binary large objects (BLOBs). Ahora ya se encuentra referida en las bases de datos pero con su propia entidad. En el ámbito de la información no estructurada se identifican seis áreas diferentes: manejo de documentos, manejo de contenidos Web, manejo de registros, manejo de derechos en el campo digital, control y seguimiento de la colaboración.

En este momento con el cambio de escenario se impone no sólo indexar y describir la referencia a los documentos por medio de registros de información estructurada sino llegar a administrar información no-estructurada. La respuesta es la sinergia de dos tipos. La indización asistida para el manejo de conocimiento tecnología: XML y SQL, es decir lenguaje de marcado y la gestión bases de datos, esto permite la representación y

administración de información no-estructurada con los recursos de manejo de la información estructurada.

Los documentos se estructuran con el lenguaje de marcado, así el límite entre información estructurada y no estructurada también va siendo progresivamente borrado, ya que no sólo se indican aspectos formales sino que se incluyen marcas con valor semántico, es decir que entramos ya en el campo de la indización temática.

La información estructurada permite distinguir entre el contenido, (palabras, imágenes, gráficos, etc.), y el papel que juega dentro del documento, (por ejemplo, no es lo mismo el contenido informativo de un encabezado, que el de un pie de página, ya que el primero es más relevante). Se puede asegurar que casi todos los documentos tienen una estructura definida. El siguiente paso es identificar esas estructuras. A partir de ahora la palabra documento tendrá para nosotros un significado añadido, además de información, palabras, imágenes y gráficos, tendrá una “estructura reconocible”.

Las tecnologías actuales de marcas buscan dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de pedazos de información. Ejemplos son un tema musical, que se compone de compases, que están formados a su vez con notas [1].

2.1.2 Metadatos

El concepto *metadato* nació en la década de los sesentas, y su creación se atribuye a Jack Myers, quien lo utilizó para describir datos correctamente [2]; actualmente, la definición más sencilla de metadatos es la que los define como “datos sobre datos” de tal forma que lo que básicamente representarán será metainformación. A pesar de esto, tradicionalmente, el término se ha empleado con referencia a datos sobre indización y catalogación creados por profesionales en la ciencia de la información; no obstante, su definición estará subordinada tanto a su labor como al campo profesional que los maneje.

Una definición más completa es la que se presenta a continuación:

“Metadatos es toda aquella información descriptiva sobre el contexto, calidad, condición o características de un recurso que tiene la finalidad de facilitar su recuperación, autenticación, evaluación, preservación o interoperabilidad [3]”

Por tal razón, su papel va más allá de lo simplemente descriptivo, puesto que también se usan en la elaboración de registros de adquisición, catálogos de exhibiciones, datos sobre utilización o incluso, información para la administración de información digital; todo esto en pro de la recuperación de información digital. Por esto, los metadatos se convierten en componentes primordiales de las Bibliotecas Digitales.

En términos generales, la palabra metadato, en el contexto de las bibliotecas, digitales o no, se utiliza comúnmente para referirse a información que:

- Proporciona (normalmente de forma breve) una caracterización de un recurso de información específico en las colecciones de la biblioteca
- Se almacena en los catálogos
- Se usa principalmente para ayudar al usuario a acceder a los recursos de información de su interés

2.1.3 Propuestas e iniciativas para metadatos: formatos, esquemas o estándares relacionados a ediciones digitales

En la actualidad existen numerosos sistemas que se están implementando en gran cantidad de proyectos. Dado que es prácticamente imposible recogerlos todos, se mencionarán aquellos que afectan directamente al procesamiento de la información, los que tengan un uso más extendido y los que, además, satisfacen los siguientes requerimientos:

- Identificación de documentos en un entorno distribuido
- Descripción de su contenido
- Localización y accesibilidad
- Gestión de derechos: copyright

Los estándares para metadatos que destacan son los siguientes:

- Los aceptados por la norma HTML.
- DC o DCMI (Dublin Core Metadata Initiative).
- RDF (Resource Description Framework).
- TEI (Text Encoding Initiative).
- MARC DTD (Machine Readable Cataloging Document Type Definition).
- EAD (Encoded Archival Description).
- PICS (Platform for Internet Content Selection).
- MCF (Meta Content Format).
- IAFA (Internet Anonymous FTP Archive).
- SOIF (Summary Object Interchange Format).

La mayoría de estos sistemas se utilizan de forma aislada, ya que su objetivo es, fundamentalmente, satisfacer unos requerimientos muy específicos (EAD para descripción de documentos de archivos, SOIF e IAFA como ficheros de intercambio en sistemas de indización distribuida, PICS para permitir o no el acceso a determinados contenidos).

Mención aparte merece RDF que, gracias a su orientación, permite la inclusión de otros sistemas de metadatos para favorecer el intercambio de información entre bases de datos heterogéneas [4].

2.2 Análisis de las herramientas a utilizar

Las posibilidades de seleccionar herramientas para un proceso de transformación dinámica son demasiado amplias en la actualidad, por este motivo debemos tener mucha cautela en la selección. Se analizarán las más competitivas con el fin de no volver exhaustivo el proceso de selección.

2.2.1 XML, DTD, TEI y Hojas de estilo

XML es un lenguaje que permite jerarquizar y estructurar la información y describir los contenidos dentro del propio documento, así como la reutilización de partes del mismo. La información estructurada presenta varios contenidos (texto, imágenes, audio, etc.) y formas: hojas de cálculo, tablas de datos, libretas de direcciones, parámetros de configuración, dibujos técnicos, etc. La forma da alguna indicación de qué papel puede jugar el contenido (por ejemplo, el contenido de una sección encabezada con un significado difiere del contenido de una nota a pie de página, lo que significa algo diferente entre el contenido de un pie de foto o el contenido de una tabla de datos). Más o menos todos los documentos tienen la misma estructura.

Como hemos afirmado anteriormente, un lenguaje de marcas es un mecanismo para identificar estructuras en un documento. La especificación XML define una manera estándar de añadir marcas a los documentos.

El lenguaje XML utiliza etiquetas (palabras entre corchetes agudos: '<' y '>') y atributos (name="valor"), para delimitar piezas de datos, dejando la interpretación de los datos a la aplicación que los lee. Así, pues, una etiqueta como <I> puede ser un libro, leche, libras, lotería, etc., dependiendo del contexto.

XML no especifica ninguna semántica o conjunto de etiquetas. De hecho, XML es realmente un metalenguaje para describir lenguajes de marcas. En otras palabras, XML facilita definir etiquetas y relaciones estructurales entre ellas. Desde que un conjunto de etiquetas no se predefine, no puede haber una semántica preconcebida. De todas las semánticas de un documento XML, ésta será definida por las aplicaciones del proceso o por las hojas de estilo.

XML es un estándar internacional desarrollado por un Grupo de Trabajo de XML (conocido como el Comité de Revisión Editorial de SGML) formado bajo el auspicio del World Wide Web Consortium (W3C) en 1996. [6] La Recomendación dice textualmente: "El Lenguaje Extensible de Marcas, abreviado XML, describe una clase de objetos de datos llamados documentos XML y parcialmente describe el comportamiento de programas de computador que pueden ser procesados."

Existen dos conceptos incluidos en la definición que es necesario explicar para poder tener un mejor entendimiento de lo que en realidad expresa dicha definición. A continuación están descritos:

- Las marcas o marcado son señales con un propósito que se añaden a un texto para ayudar a su procesado. Como por ejemplo HTML, o el sistema que utilizan los procesadores de texto para, por ejemplo, delimitar aquellas palabras que están en cursiva, o, en un estilo de letra particular.
- Extensible, quiere decir que se pueden definir las marcas como se desee, siempre pensando en la función que pueden desarrollar para no trabajar de más. XML se ajusta a las necesidades, se ha diseñado para adaptarse a una gran cantidad de situaciones diferentes, también, y como consecuencia, se adapta a los conocimientos de la persona que lo emplea, pudiendo elaborarse documentos XML extremadamente sencillos, y documentos complicados.

Según la especificación los objetivos de diseñar XML fueron los siguientes:

- XML debe ser directamente utilizable en Internet
- XML debe soportar una amplia variedad de aplicaciones
- XML debe ser compatible con SGML
- Deberá ser sencillo escribir programas que procesen documentos XML
- El número de las características opcionales en XML deberá ser el mínimo posible, a ser posible cero
- Los documentos XML deberán ser legibles por las personas y razonablemente claros
- El diseño de XML debe ser rápido
- XML debería ser simple, pero perfectamente normalizado
- Los documentos XML deben ser de fácil creación
- La concisión de las marcas XML tiene una importancia mínima

2.3 Documentos XML

Los documentos XML, tienen relación directa con diversos elementos para poder generar transformaciones y presentaciones de los metadatos. En esta sección analizaremos algunos de los elementos que servirían en el proceso de transformación dinámica.

2.3.1 DTD

La necesidad de jerarquizar y estructurar correctamente la información, no sólo para almacenarla, sino también para acceder a ella, se ha convertido en una labor que ha cobrado especial relevancia en los últimos años, en los que se han producido importantes avances en este campo.

DTD (Document Type Definition, Definición de tipo de documento) define los tipos de elementos, atributos, entidades y notaciones que se podrán utilizar en el documento, así como ciertas restricciones estructurales y de contenido, valores por defecto, etc. [7]

Este conjunto de marcas que contiene un documento debe escribirse siguiendo unas reglas, que indican la forma en que deben de incluirse estas.

La Declaración de Tipo de Documento (DTD-Document Type Definition):

Al definir el lenguaje XML ya nos referimos a la Definición del Tipo de Documento (Document Type Definition DTD)_que, en resumen, cumple las siguientes funciones:

Un DTD especifica la clase de documento

- Describe un formato de datos
- Usa un formato común de datos entre aplicaciones
- Verifica los datos al intercambiarlos
- Verifica un mismo conjunto de datos

Un DTD describe su composición:

- Elementos: cuáles son las etiquetas permitidas y cuál es el contenido de cada etiqueta
- Estructura: en qué orden van las etiquetas en el documento
- Anidamiento: qué etiquetas van dentro de cuáles

Los elementos de un DTD se clasifican en:

- Elementos con “contenido ELEMENT”:

Un elemento tiene contenido ELEMENT, si sólo puede contener a otros elementos, opcionalmente separados por espacios en blanco.

- Elementos con “contenido TEXT” :

Un elemento tiene contenido TEXT, si sólo puede contener texto (PCDATA = printable character data).

- Elementos con “contenido MIXED”

Un elemento tiene contenido MIXED, si puede contener texto u otros elementos.

- Elementos con “contenido EMPTY”

Un elemento tiene contenido EMPTY, si no puede contener otros elementos, y sus atributos son los siguientes:

CDATA: texto

ID: empezar con letra

IDREF: ser un ID

Así pues, el DTD especifica la clase de documento XML. El DTD indica sólo qué elementos, atributos, etc; tiene un documento y cómo se anidan. Contiene la información sobre los elementos que se puedan utilizar en un tipo de documento específico, cuáles de ellos son obligatorios, se puedan repetir en más de una ocasión, y los que pueden aparecer una sola vez, así como el orden en el que se deben de escribir los elementos y como pueden anidarse es decir, qué elementos pueden contener a otros.

2.3.2 TEI

Text Encoding Initiative (TEI) es un proyecto internacional que pretende crear un estándar de codificación que facilite el intercambio de texto en formato electrónico.

TEI surge en 1987 en un Simposio de la Association for Computers and the Humanities en Vassar Collage, USA.

Se definen tres objetivos principales para el proyecto:

- Definir un formato de intercambio de textos codificados electrónicamente
- Proporcionar un conjunto de recomendaciones o líneas maestras para la codificación de nuevo material textual. Éstas deberán codificar tanto los rasgos que tendrán que ser codificados como el modo de codificarlos.
- Documentarse sobre los sistemas de codificación ya existentes y desarrollar un metalenguaje que los describa. En este sentido se adoptará como metalenguaje, en la medida de lo posible, el estándar ISO 8879 (Standard Generalized Markup Language) [8].

Todos los textos que cumplan el TEI contienen (a) un encabezado (marcado con el elemento <teiHeader>) y (b) la transcripción del propio texto (marcado con el elemento <text>).

El encabezado TEI provee información similar a la de la cubierta de un texto impreso. Tiene hasta cuatro partes: una descripción bibliográfica del texto electrónico, una descripción de cómo ha sido etiquetado, una descripción no bibliográfica del texto (un perfil del texto), y una revisión de su historia (su creación).

Un texto TEI puede ser individual (una única obra) o compuesto (una colección de obras, como por ejemplo una serie). El texto puede tener front y back opcional. En medio está el body, cuerpo de la obra, que, en ciertos casos puede estar formado por groups, cada uno conteniendo a su vez más grupos o textos [9].

Un texto individual se etiquetará siguiendo una estructura genérica como esta:

```
<TEI.2>
  <teiHeader> [ Información del encabezado TEI ] </teiHeader>
  <text>
    <front> [ materia del front ... ] </front>
    <body> [ cuerpo del texto ... ] </body>
    <back> [ materia back ... ] </back>
  </text>
</TEI.2>
```

Como se ha indicado arriba, un documento TEI sencillo está formado por los siguientes elementos:

<front>:

contiene los materiales previos al comienzo del texto propiamente dicho: introducción, prólogos, dedicatorias, etc.

<body>:

contiene el cuerpo de texto individual(exceptuando la materia del front y del back).

<back>

contiene los apéndices, etc.

El cuerpo de un texto en prosa puede ser simplemente un conjunto de párrafos, o estos pueden estar agrupados en capítulos, secciones, subsecciones, etc. En el primer caso, cada párrafo está etiquetado con la etiqueta `<p>`. En el segundo caso, el `<body>` puede estar dividido en diferentes elementos `<div1>`, o en elementos `<div>`, y cada uno de estos subdivididos a su vez como se ve más abajo:

<p>

indica los párrafo en prosa.

<div>

contiene una subdivisión del front, body o back del texto.

<div1>

contiene una subdivisión de primer nivel del front, body o back de un texto (es la mayor si no se usa el `<div0>`, si se usara sería la segunda mayor).

Cuando hay subdivisiones estructurales más pequeñas que el `<div1>`, ésta puede dividirse en elementos `<div2>`, los elementos `<div2>` en elementos `<div3>` menores, etc., hasta el nivel de la `<div7>`. Si fueran necesarias más de siete nivel de divisiones estructurales, se debe modificar el conjunto de etiquetas del TEI para que acepte `<div8>`, etc., o para usar los elementos `<div>` sin numerar: un `<div>` puede subdividirse en `<div>` más pequeños sin ningún límite de anidamiento.

Estos elementos de división tienen los mismos tres atributos:

type

Indica el nombre convencional para esta categoría de división textual. Su valor será normalmente "Book", "Chapter", "Poem", etc. Otros posibles valores son "Group" para

grupos de poemas, etc., tratados como unidades sencillas sería "Sonnet", "Speech" y "Song". Los valores insertados en el atributo type del primer <div>, <div1>, <div2>, etc., en un texto se supone que es aplicable a los siguientes <div>, <div1>s, etc., dentro del mismo <body>. Esto supone que sólo se debe indicar el valor en el primer elemento de división de cada tipo, o donde haya cambio.

id

Especifica un identificador único para la división, que puede usarse para hacer referencias cruzadas o enlaces. A menudo es útil insertar un atributo id en las unidades estructurales de nivel superior de un texto y obtener los valores id de algún modo automático, por ejemplo añadiendo un número de sección a un pequeño código usado para el título de la obra en cuestión, como se verá en los ejemplos de abajo.

n

El atributo n especifica un corto nombre o número mnemotécnico para la división, que puede usarse para identificarla en lugar del id. Si existe un modo convencional de referencia o abreviatura para las partes de una obra (como el modelo 'libro/capítulo/versículo' para las citas bíblicas), el atributo n es el lugar para indicarlo.

2.3.3 Hojas de estilo

XSL

XSL o *Extensible Stylesheet Language* es una familia de recomendaciones del World Wide Web Consortium (<http://www.w3.org/Style/XSL/>) para hacer Hojas de estilo en lenguaje XML. El lenguaje XSL consta de tres secciones:

- XSL Transformations (XSLT): un lenguaje para transformar documentos XML (<http://www.w3.org/TR/xslt> y <http://www.w3.org/TR/xslt20/>)
- XML Path Language (XPath), lenguaje de expresión usado por XSLT para acceder o referirse a partes de un documento XML. (XPath se usa también en la especificación X Link). (<http://www.w3.org/TR/xpath> y <http://www.w3.org/TR/xpath20/>).
- XSL Formatting Objects (XSL-FO): un vocabulario XML para especificar formatos semánticos. <http://www.w3.org/TR/xsl>

XSL nos permite diseñar plantillas en las que definimos cómo es la salida que esperamos obtener al procesar el documento XML fuente. Básicamente, XSL es un lenguaje que define la transformación entre un documento XML de entrada, y otro documento XML de salida.

El XSL define la presentación o formato para un documento XML. Un mismo documento XML puede tener varias hojas de estilo XSL que lo muestren en diferentes formatos (HTML, PDF, RTF, VRML, PostScript, sonido, etc.)

La especificación XSLT (eXtensible stylesheet language-Transformations) describe un lenguaje basado en XML para transformar documentos XML a cualquier otro formato. XSLT forma parte de la especificación XSL (eXtensible stylesheet language), [10].

XSLT o XSL es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las hojas de estilo (aunque el término de hojas de estilo no se aplica sobre la función directa del XSLT) XSLT realizan la transformación del documento utilizando una o varias reglas de plantilla: unidas al documento fuente a transformar, esas reglas de plantilla alimentan a un procesador de XSLT, el cual realiza las transformaciones deseadas colocando el resultado en un archivo de salida o, como en el caso de una página web, directamente en un dispositivo de presentación, como el monitor de un usuario.

El procesador de hojas de estilo procesará de forma recursiva los elementos del origen para producir un completo árbol de objetos de flujo. Además de las reglas de construcción, XSL también soporta reglas de estilo, las cuales permiten la mezcla de características. Mientras que una sola regla de construcción puede ser invocada para un elemento en particular del origen, pueden ser invocadas todas las reglas de estilo aplicables, permitiendo la mezcla de características como en CSS.

Resumiendo, un hoja de estilo XSL describe el proceso de presentación a través de un pequeño conjunto de elementos XML. Esta hoja, puede contener elementos de reglas que representan a las reglas de construcción y elementos de estilo de estilos.

CSS

Una hoja de estilo (*style sheets*) es una colección de reglas que afectan a la apariencia de un documento. Estas reglas se refieren al modo en que aparecerá un documento en pantalla cuando el usuario utilice un navegador o explorador gráfico, controlando por ejemplo el color, el fondo, tipo de fuente, apariencia de los bordes, márgenes, alineación y espacio entre caracteres.

Las ventajas que ofrece la utilización de hojas de estilo es que se diseñan de forma independiente al documento HTML y que se se pueden aplicar, enteras o a alguno de sus niveles, y a los documentos que se desee, facilitando la consistencia y homogeneidad en el diseño y la imagen del sitio web. Además, las hojas de estilo se pueden adecuar a los distintos medios de presentación de documentos.

Las hojas de estilo en cascada (o CSS) es un mecanismo que describe cómo se va a mostrar un documento en la pantalla [11].

Es un estándar desarrollado por el World Web Consortium (W3C) para hacer que las páginas Web tuvieran mayor capacidad al ser visualizadas.

El W3C se refiere a estos documentos como “hojas de estilo en cascada” por que se pueden utilizar múltiples estilos para controlar las apariencias de nuestras páginas, y el navegador sigue las reglas en cascada para determinar la prioridad.

CSS, es una tecnología que nos permite crear páginas web de manera más precisa y homogénea. Gracias a las CSS controlamos más aún los resultados finales de la página, pudiendo hacer muchas cosas que no se podrían hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, etc.

2.4.1 Lenguajes de programación PHP Y JAVA

El desempeño, nivel de penetración en el mercado, facilidad de uso, escalabilidad, y otros factores son los que generar las dos posibilidades presentadas a continuación.

2.4.1.1 Lenguaje de programación PHP

Es un lenguaje para programar scripts del lado del servidor (bloques de código de programación que permiten ejecutar instrucciones de forma dinámica), que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma.

PHP es un acrónimo recursivo que significa "**PHP** Hypertext **P**re-processor" (inicialmente PHP Tools, o, *Personal Home Page Tools*). Es un lenguaje de programación gratuito e independiente de plataforma, rápido, con un gran conjunto de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores [11].

2.4.1.2 Lenguaje de programación JAVA.

Java es un lenguaje de programación de alto nivel que tiene las siguientes características:

- Orientado a objetos. Java acoge la moderna filosofía de programación orientada a objetos.
- Distribuido y dinámico. Java se diseñó teniendo en cuenta el ambiente de Internet. Los programas de Java se transportan con cierta cantidad de información que se utiliza para verificar y resolver los accesos a objetos en tiempo de ejecución.
- Robusto. Java favorece la programación libre de errores porque se digita estrictamente y realiza revisiones de alto rendimiento en tiempo de ejecución.
- Seguro. Java proporciona un medio seguro de crear aplicaciones para Internet.
- Portable. Los programas de Java se pueden ejecutar en cualquier ambiente donde exista un sistema de ejecución Java [12].

Java tiene la característica de ser, al mismo tiempo compilado e interpretado. El compilador es el encargado de convertir el código fuente de un programa en un código intermedio llamado *bytecode* que es independiente de la plataforma en que se trabaje y que es ejecutado por el intérprete de Java que forma parte de la Máquina Virtual de Java.

2.5 Motores de transformación dinámica

Han sido desarrollados dos métodos de analizar sintácticamente un documento XML. En el primer método, DOM (Document Object Model), se lee el documento completo y se identifica su estructura jerárquica. El segundo método, SAX (Standard API for XML), consiste en ir identificando las marcas de XML a medida que se va leyendo el documento. El segundo método es más rápido y consume menos recursos, pero tiene la desventaja de que cada vez que aparece una marca se debe decidir que hacer con ella, y no se puede regresar para atrás en el documento.

-SAX ha sido desarrollado con aplicaciones de servidor; el servidor debe suministrar rápidamente el resultado de transformar un documento XML. DOM fue desarrollado con aplicaciones de cliente; por ejemplo un editor de XML necesita poder navegar en cualquier dirección la estructura del documento; en este caso el método SAX no sería muy útil. Los modelos SAX y DOM son modelos genéricos de tratamiento de los archivos XML, pero no son sencillos de utilizar. Para ello existen en el mercado diversas APIs que permiten que la realización de programas que trabajan con XML sea mucho más sencilla. Existen varias alternativas para utilizar o manejar XML en aplicaciones Java. Se pueden destacar las siguientes:

-JDOM no es una envoltura alrededor de DOM, aunque comparte el mismo propósito que DOM con respecto a XML. Se ha hecho lo suficientemente genérico para dirigirse a cualquier modelo de documento. JDOM ha sido optimizado para Java y además, para el uso del API Java Collection. Los documentos JDOM pueden construirse y convertirse directamente a eventos SAX o árboles DOM, permitiendo que JDOM sea integrado en las tuberías de procesamiento XML y en particular con fuentes o resultados de transformaciones XSLT.

- DOM4J es otro API alternativo muy similar a JDOM. Además viene con una integración a XPath: el interface `org.dom4j.Node` por ejemplo define métodos para seleccionar nodos de acuerdo a una expresión XPath. `dom4j` también implanta el modelo de procesamiento basado en eventos lo que le permite un procesamiento eficiente de grandes documentos XML. Los manejadores pueden registrarse para ser llamados durante el análisis cuando se encuentran las expresiones XPath, permitiéndonos procesar inmediatamente y disponer de las partes del documento sin tener que esperar a que todo el documento sea analizado y cargado en memoria.

- StaX (Streaming API for XML): Es una API cuyo objetivo principal no es la representación en memoria del documento XML sino su tratamiento eficiente como flujo de caracteres con significado. Permite de esta manera el tratamiento rápido y a bajo nivel del XML sin tener que utilizar ni DOM ni la serialización del objeto DOM [13].

- JAXP sirve para procesar los datos XML usando aplicaciones escritas en el lenguaje de programación Java. JAXP soporta los procesadores (parsers) estándar SAX y DOM, elige entre procesar tus datos como un flujo de eventos o construir una representación objeto de éstos. JAXP también soporta el estándar XSLT, dándonos control sobre la

representación de los datos, así como la conversión de los datos a otros formatos, tales como HTML. Los principales APIs de JAXP están definidos en el paquete `javax.xml.parsers`. Este paquete contiene dos clases: `SAXParserFactory` y `DocumentBuilderFactory`, los cuales proporcionan un procesador SAX (`SAXParser`) y un constructor de documentos (`DocumentBuilder`). El constructor de documentos crea un compilador DOM de documentos.

2.6 Propuesta de solución

Una vez que analizados los elementos clave para generar la solución presentaremos las justificaciones y análisis para presentar una propuesta real que genere resultados.

2.6.1 Análisis de información obtenida

La información obtenida en la investigación indica que el camino para generar una edición digital requiere un método para publicar y herramientas para integrar y desarrollar cada uno de los elementos de la edición digital. Los principales aspectos detectados para generar una edición digital son los siguientes:

- Recepción de datos
- Análisis de datos para la publicación de ediciones digitales
- Procesamiento de datos
- Resultados (Obra o Edición digital)

Las herramientas necesarias para operar los procesos, son escogidas tomando en cuenta el uso estandarizado y la capacidad de escalabilidad de las mismas.

En lenguajes de programación, se propone generar la transformación con un lenguaje orientado a objetos, teniendo como mejor opción Java para el núcleo y JSP para la parte de la presentación.

En el lenguaje de marcas descriptivas, se propone al metalenguaje XML por su flexibilidad y difusión.

Para el servicio de publicación se eligió el Server Web Tomcat, por ser fácil y accesible en su configuración además de ofrecer soporte para las herramientas anteriores.

Para el soporte del estilo gráfico de la edición digital se optó por el lenguaje de estilo XSLT (eXtensible Style Sheet Transformation) para la parte dinámica y el CSS (Cascade Style Sheet) para la parte estática.

HTML Transitional 4.01 será la propuesta para el producto final de la edición digital, que optimiza el trabajo para Internet Explorer 6. Cada herramienta tiene su propia versión (esta depende de las compatibilidades entre ellas y la plataforma de desarrollo).

Tabla 2.1

VERSIONES PROPUESTAS PARA EL DESARROLLO		
HERRAMIENTA	VERSION	LIGA DE RECURSO
JAVA (jdk)	1.5	http://java.sun.com/j2se/1.5.0/docs/api/
JSP	2.0	http://java.sun.com/products/jsp/docs.html
XML	1.0	http://www.w3.org/TR/REC-xml/
XSLT	1.0	http://www.w3.org/TR/xslt
CSS	2.1	http://www.w3.org/Style/CSS/
TOMCAT	5.0	http://tomcat.apache.org/tomcat-5.0-doc/index.html
HTML	4.01	http://www.w3.org/TR/html401/
INTERNET EXPLORER	6	http://www.microsoft.com/spain/windows/ie/ie6/features.msp
IDE PARA DESARROLLO – NETBEANS	5.5.1	http://www.netbeans.org/

Algunos detalles como los formatos de imágenes se analizarán en la producción, se considerarán los formatos que tengan mayor difusión. Además se usarán validadores de código fuente no gratuitos.

Como justificación para usar los elementos de la *Tabla 2.1* podemos enumerar lo siguiente:

- I. El lenguaje JAVA (éste incluye a JSP), es un lenguaje robusto, dinámico y en crecimiento, además de tener un soporte amplio para la tecnología XML.

2. XML, se usa por ser el estándar con mayor aceptación en el mundo de los metalenguajes.
3. XSLT por ser una extensión adaptada totalmente a la transformación dinámica de XML.
4. CSS sirve para potenciar el estilo en el producto final, debido a que éste es HTML.
5. TOMCAT es un contenedor de Servlets lo que nos permite usar la tecnología JAVA fácil y eficientemente, además no agrega un costo en los proyectos.
6. HTML, es seleccionado como producto de salida por ser el lenguaje de marcas más aceptado por los exploradores actuales en Internet, y esto permite que las ediciones generadas no tengan problemas de compatibilidad.
7. Internet Explorer 6. Es una versión de explorador que tienen instalada la mayoría de las computadoras en este momento, y los cambios para futuras versiones se espera que no sean tan radicales para el sistema.
8. IDE-NETBEANS, es seleccionada por ser una interfaz para desarrollo de proyectos que no incrementa el costo, además de ser clara como interfaz (de fácil uso).

2.7 Proceso de integración y solución propuesta

Cada uno de los elementos necesarios para generar una edición digital se integrarán en un proceso final, que tiene como objetivo crear el productor final dependiente de cada proceso.

La metodología empleada en la implantación de la solución tendrá como apoyo el modelo de procesos MoProSoft Ver. 1.3 (Modelo de Procesos para la Industria de Software, Fig. 2.1 – Diagrama MoProSoft).

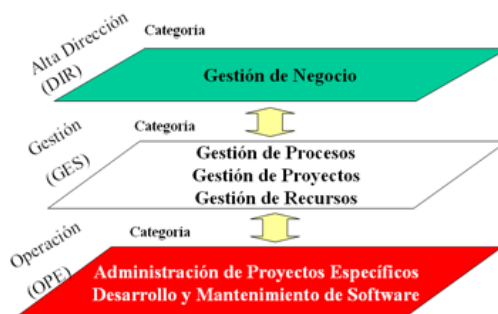


Fig. 2.1 - Diagrama MoProSoft [14]

¿Porqué usar MoProSoft?

MoProSoft es un Modelo de Procesos que esta creado para México y no es costosa su implantación dentro de un proyecto (comparada con la de otros modelos).

Para tener un control en el desarrollo del proceso de la edición digital nos apoyaremos en la capa de operación, específicamente en Desarrollo y Mantenimiento de Software. MoProSoft tiene un objetivo enfocado a los procesos y no a ser usado como una metodología de software, es por eso que nos apoyaremos sólo en la capa de operación. Para la solución apoyada en el modelo, se considerarán las siguientes estrategias:

En la capa de recepción de datos, el proceso tiene como responsabilidad el obtener una fuente de datos que sea capaz de proporcionar los medios suficientes para crear la versión digital del ejemplar. Se hace el análisis y diseño de la obra por parte de cada uno de los responsables, tanto de diseño gráfico, desarrollo editorial, y programación. Posteriormente se proporcionan los datos para nuevo análisis considerando ahora la estructura de marcado, de ésta se deriva una estructura prototipo de la edición digital.

En forma paralela a la creación de la estructura prototipo se genera el modelo gráfico adecuado.

Con base en los resultados prototipo de los análisis anteriores se genera una transformación parcial que posteriormente será el núcleo del modelo iterativo para generar la transformación dinámica de la edición digital.

La metodología interna que se usará tendrá como base parcial a RUP, usando el ciclo de vida como principal parte del modelo, este ciclo se explican en la sección [A3](#) de los anexos.

Se presenta el diagrama de solución propuesto (Fig. 2.2) de la transformación de la estructura de marcado para la generación de una edición digital.

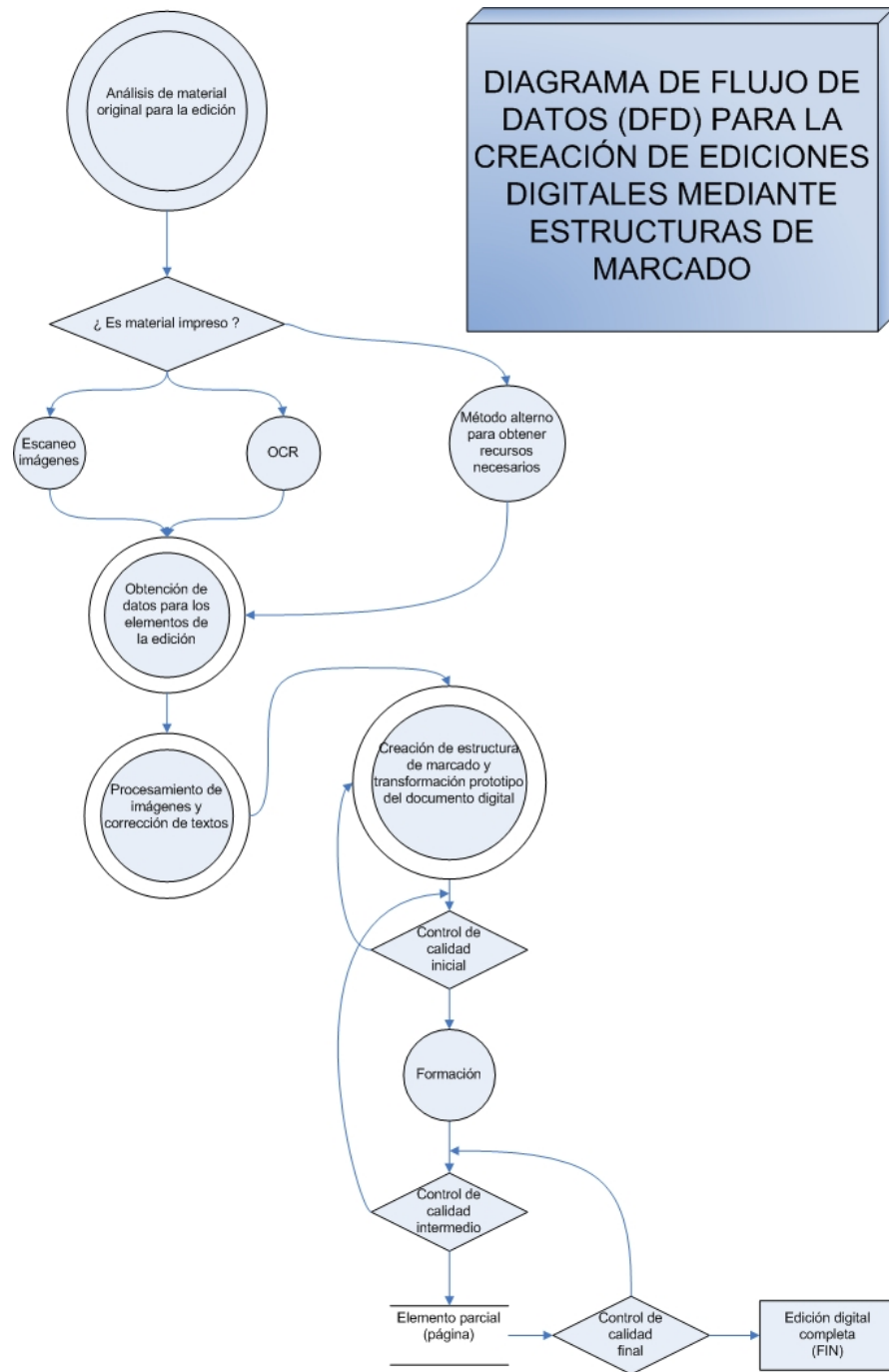


Fig. 2.2 – Diagrama de flujo de datos de edición digital completa

BIBLIOGRAFÍA DEL CAPÍTULO

[1] <http://metadatos-xml-rdf.awardspace.com/xml.html>
visitado en mayo de 2007

[2] <http://revista.unam.mx>
visitado en mayo de 2007

[3] <http://eprints.rclis.org/>
visitado en Mayo de 2007

[4] José A. Senso (2003, junio 1). El concepto de metadato. Algo más que descripción de recursos electrónicos.: Universidad de Granada. España.

[5] XML
<http://www.hipertexto.info/documentos/xml.htm>

[6] Características de XML.
XML a través de ejemplos. Abraham Gutiérrez, Raúl Martínez
Alfaomega, 2001

[7] Text Encoding Initiative (TEI)
<http://elies.rediris.es/elies3/cap132.htm>

[8] Estructura TEI.
http://www.tei-c.org/Lite/teiu5_sp.html

[9] XSL
Programación con XML, Ricardo Eito Brun.
Anaya Multimedia, 2001

[10] CSS
<http://www.w3c.es/Divulgacion/Guiasbreves/HojasEstilo>

[11] Lenguaje de programación PHP
<http://www.desarrolloweb.com/articulos/392.php>

[12] Características de Java.
http://www.fi-b.unam.mx/pp/profesores/carlos/java/java_basico1_1.html

[13] Motores de transformación dinámica.
<http://personal.oreto.inf-cr.uclm.es/fpromero/files/pdm/pr01.pdf>

[14] <http://www.software.net.mx/desarrolladores/directorios/asociaciones/amcis/> visitado en Mayo de 2007

Capítulo 3

Análisis y diseño

3.1 Requerimientos de publicación

Es necesario conocer y ubicar perfectamente los elementos involucrados en la publicación, esto nos servirá para determinar los pasos a seguir en cada proceso.

3.1.1 Descripción de requerimientos gráficos

Tomando en cuenta los requerimientos técnicos de visualización de la edición digital:

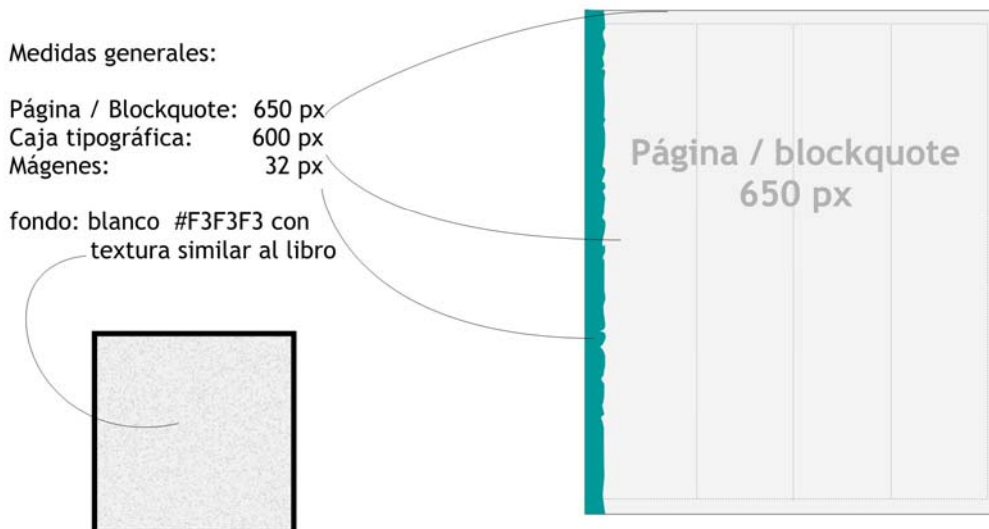
Monitor con resolución de 800 x 600 píxeles. Lo anterior representa el número de puntos que puede tener el monitor por pantalla, en horizontal por vertical. Así, un monitor cuya resolución sea de 800x600 puntos puede representar hasta 600 líneas horizontales de 800 puntos cada una.

La medida de nuestra página será de 650 píxeles y nuestra caja tipográfica de 600 píxeles, con 50 píxeles de margen horizontal. Para hacer un poco de contraste con el fondo de la pantalla, nuestra página tendrá un fondo gris muy claro para el texto, (el color en hexadecimal es #F3F3F3).

El análisis previo de los diferentes colores que contiene la edición digital, dio como resultado una lista con todos los colores en hexadecimal, estos colores se usan en títulos, texto en general, bordes y fondos de tablas.

También existe un ejemplo (Fig. 3.1) para las diferentes tipografías que contiene la edición digital, porque el texto suele resaltarse con colores diferentes, en ocasiones un color puede entenderse como una clasificación de un contenido o significado en particular.

Publicación: La Educación Intercultural Bilingüe



Números de los colores definitivos para las unidades del libro de: La Educación Intercultural Bilingüe

WEB

#009999
#FFCC33
#Ff6633
#660066
#33CC33
#CC99FF
#CC0000
#9999CC

Cuerpo de texto

Tipografía DOM DIAGONAL y GARML TC para las lecciones, tipografía: **Verdana o Arial** para los textos de todo el libro. **COMIC SANS** para algunos subtítulos y en cuadros

Toda la tipografía debe ser en **REDONDAS / (IMPORTANTÍSIMO) NO CURSIVAS**

Tipografía: Verdana o Arial
Puntos: 11 pts
Color: #000000 (negro)
Alineación: izquierda

Fig. 3.1 - Plantilla I

Números de los colores definitivos para los márgenes del libro de: La Educación Intercultural Bilingüe

NOTA Los colores de los márgenes serán correspondientes a cada unidad

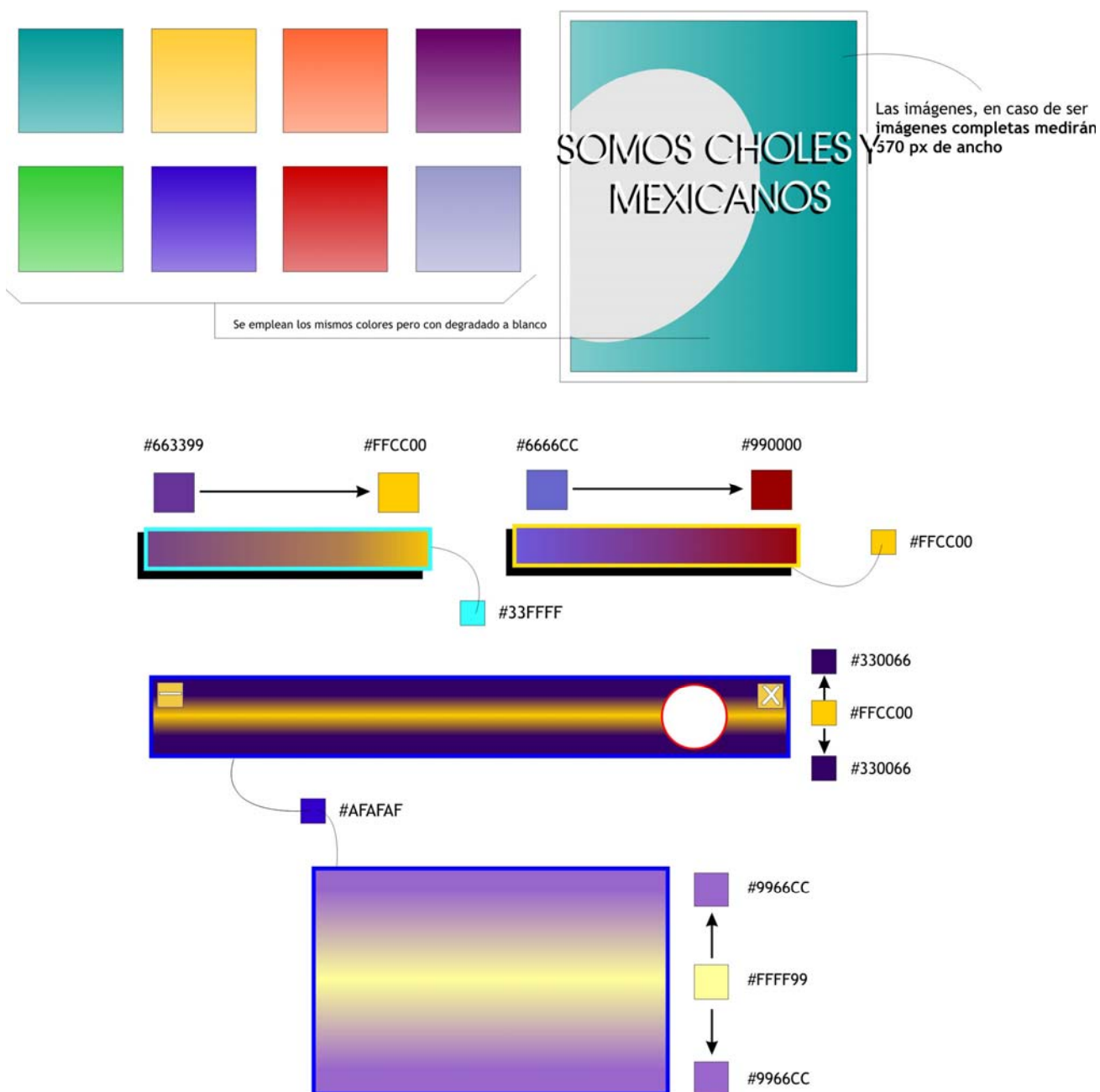


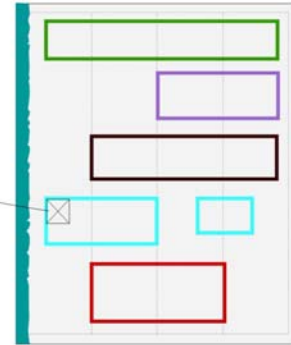
Fig. 3.2 - Plantilla 2

Cada inicio de unidad tiene una imagen con fondo de color diferente que indica una nueva lección (Fig. 3.2). Existen imágenes que miden 570 píxeles, y son de este tamaño por que van a servir como imagen de fondo para que se pueda sobreponer el título de cada

lección. Como se puede notar este tipo de imágenes está en el límite del tamaño para respetar el ancho de la página.

Colores y medidas para los recuadros del libro.

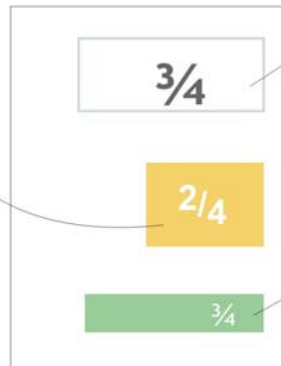
Nota: Algunas imágenes contienen el gráfico de un lápiz que en libro esta colocada afuera del recuadro, SE DEBERÁ INCORPORAR ESTE GRÁFICO DENTRO DEL RECUADRO



Medidas de los recuadros:

Recuadro del tamaño $\frac{3}{4}$ del ancho de la página: 400 px de ancho por lo que desarrolle de alto

Recuadro de $\frac{2}{4}$ del ancho de la página: 300 px de ancho por lo que desarrolle de alto



NOTA: para solucionar los recuadros que contengan imágenes, es recomendable usar los colores antes mencionados según la unidad

Fig. 3.3 - Plantilla 3

Existen imágenes de diferentes tamaños (Fig. 3.3), ya que algunas requieren ser de mayor tamaño por el tipo de contenido que se visualiza en ellas. Como podemos ver se hace una clasificación de ellas, mediante una división en 4 partes. 4/4 es el total del ancho de la pagina, le restamos 1 al numerador indicará que dicha imagen cubrirá esa proporción del ancho total de la pagina.

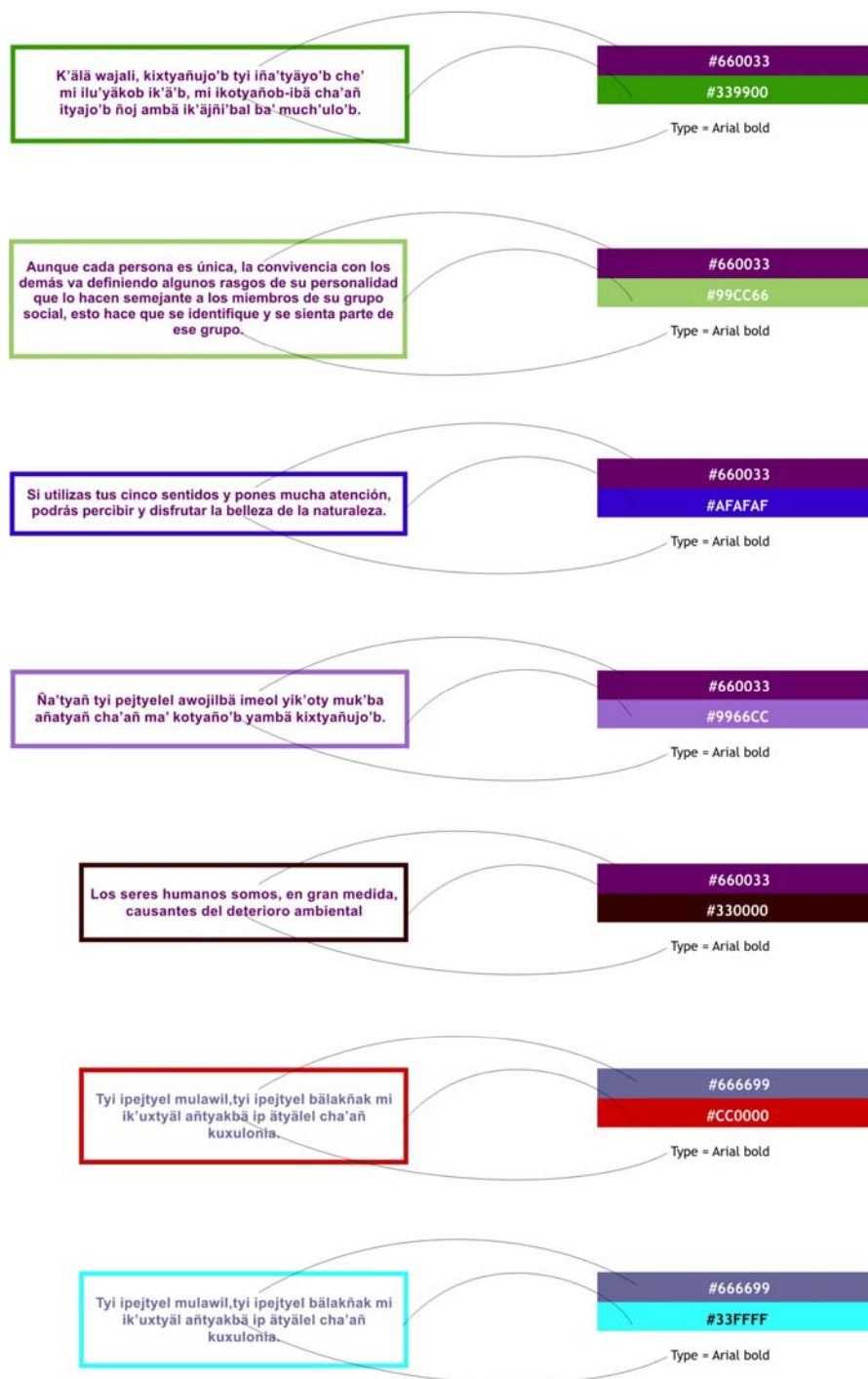


Fig. 3.4 Plantilla 4

Existen diferentes tipos de notas, las cuales aparecen rodeadas por un margen de diferentes colores, cada color hace referencia a un contenido en particular (Fig. 4C), así cuando visualizamos esta diferencia, sabremos de antemano a que hace referencia y esto en un texto es importante por que nos ayuda a determinar una parte de un texto en general.

3.1.2 Descripción de elementos Editoriales

Partimos del análisis de la que nos servirá como estructura general, hacemos mención nuevamente de ella para no perder de vista que es la base de toda la estructuración de las partes de nuestra edición digital, y es la siguiente:

```
<TEI.2>
  <teiHeader> [ Información del encabezado TEI ] </teiHeader>
  <text>
    <front> [ materia del front ... ] </front>
    <body> [ cuerpo del texto ... ] </body>
    <back> [ materia del back ... ] </back>
  </text>
</TEI.2>
```

El análisis el contenido total de la edición digital, da como resultado información sobre cada una de las partes que contiene, con lo cual empezamos a estructurar el contenido. Esta división de los contenidos simplifica las tareas, porque al hacer el análisis ahora lo hacemos como elementos por separado, sin perder de vista que el conjunto de estos forman el todo.

En los segmentos <body>, <front> and <back>, el texto puede subdividirse en: partes, capítulos, secciones; actos, escenas; cantos, estrofas; etc. haciendo uso del elemento <div>, que puede anidarse para marcar subsecciones (o bien usar elementos que van desde <div0> o <div1>, hasta <div7>). Los atributos "type" y "n" se usan para indicar que la división tiene un nombre o tipo particular y el número de ésta (p. ej., <div type="seccion" n="1">...</div>); las posteriores "div" tomarán por defecto el mismo valor de "type". Dentro de una división de texto habrá párrafos o elementos de este nivel como notas, listas, etc.

Si lo vemos en forma general sería así:

```
<TEI.2>
  <teiHeader> Título general del libro.</teiHeader>
  <text>
    <front>

    <div type="legal">
```

```
</div>

<div type="presentación">
</div>
o
<div type="introducción">
</div>
</front>
<body>
<div type="capitulos">
</div>
o
<div type="lecciones">
</div>
o
<div type="bloques">
</div>

</body>

<back>
<div type="bibliografía">
</div>
o
<div type="anexos">
</div>

</back>
</text>
</TEI.2>
```

3.2. Uso de TEI

TEI usa técnicas de búsqueda de texto, que localizan documentos que son importantes y relevantes. Cuando se recibe una solicitud de búsqueda, TEI analiza el contenido de los documentos que contienen las tablas u otros elementos de marcado y selecciona aquellos que contienen la(s) palabra(s) solicitadas.

3.2.1 Estructuración de la información

Los textos que cumplen el TEI contienen:

- a) un encabezado TEI (marcado con el elemento `<teiHeader>`)

b) la transcripción del propio texto (marcado con el elemento <text>).

El encabezado TEI provee información similar a la de la portada de un texto impreso. Tiene hasta cuatro partes: descripción bibliográfica del texto electrónico, descripción de cómo ha sido etiquetado, descripción no bibliográfica del texto (un *perfil del texto*), y revisión de su historia (su creación).

Un texto TEI puede ser *individual* (una única obra) o *compuesto* (una colección de obras). En los dos casos, el texto puede tener un *front* o *back* opcional. La parte principal es el *body*, cuerpo de la obra, que, en el caso de un texto compuesto, puede estar formado por *groups*, cada uno conteniendo a su vez más grupos o textos.

Un texto individual se etiquetará siguiendo una estructura genérica como esta:

```
<TEI.2>
  <teiHeader> [ Información del encabezado TEI ] </teiHeader>
  <text>
    <front> [ materia del front ... ] </front>
    <body> [ cuerpo del texto ... ] </body>
    <back> [ materia del back ... ] </back>
  </text>
</TEI.2>
```

Como se ha indicado arriba, un documento TEI está formado por los siguientes elementos:

<front>
contiene los materiales previos al comienzo del texto propiamente dicho: introducción, prólogos, dedicatorias, etc.

<body>
contiene todo el cuerpo de un texto (el contenido).

<back>
contiene los apéndices, bibliografía, anexos, conclusiones.

El cuerpo de un texto en prosa puede ser simplemente un conjunto de párrafos, lo común es que estén agrupados en capítulos, secciones, subsecciones, etc. Cada párrafo está contenido en la etiqueta <p>. En el segundo caso, el <body> puede estar dividido en diferentes elementos <divI>, o en elementos <div>, y cada uno de estos subdivididos a su vez [1].

3.2.1.1 Elementos de división textual

<div>
contiene una subdivisión del *front*, *body* o *back* del texto.

<divI>
contiene una subdivisión de primer nivel del *front*, *body* o *back* de un texto.

`<p>`
indica los párrafo en prosa.

Cuando hay subdivisiones estructurales más pequeñas que el `<div1>`, la `<div1>` puede dividirse en elementos `<div2>`, los elementos `<div2>` en elementos `<div3>` menores, etc., hasta el nivel de la `<div7>`. Si fueran necesarias más de siete nivel de divisiones estructurales, se debe modificar el conjunto de etiquetas del TEI para que acepte `<div8>`, etc., o para usar los elementos `<div>` sin numerar: un `<div>` puede subdividirse en `<div>` más pequeños sin ningún límite de anidamiento.

Estos elementos `<div>` tienen los mismos tres atributos:

- `type`
Indica el nombre convencional para esta categoría de división textual.
- `id`
Especifica un identificador único para la división, que puede usarse para hacer referencias cruzadas o enlaces. A menudo es útil insertar un atributo `id` en las unidades estructurales de nivel superior de un texto y obtener los valores ID de algún modo automático, por ejemplo añadiendo un número de sección a un pequeño código usado para el título de la obra en cuestión.
- `n`
El atributo `n` especifica un corto nombre para la división, que puede usarse para identificarla en lugar del ID. Si existe un modo convencional de referencia o abreviatura para las partes de una obra (como el modelo 'libro/capítulo/versículo' para las citas bíblicas), el atributo `n` es el lugar para indicarlo.

El valor de cada atributo `id` debe ser único dentro del documento. Un modo sencillo de asegurarnos de que es así es hacer que refleje la estructura jerárquica del documento. Podríamos definir valores para el `id` así:

```
<div1 id=BC1 n='I' type='tomo'>
  <div2 id=BC101 n='I.1' type='capítulo'>
    ... </div2>
  <div2 id=BC102 n='I.2' type='capítulo'>
    ... </div2>
  ...
  <div2 id=BC110 n='I.10' type='capítulo'>
    ... </div2>
  ...
</div1>
<div1 id=BC2 n='II' type='tomo'>
  ...
</div1>
...
```

3.2.1.2 Encabezados y cierres

Cada `<div>`, `<div1>`, `<div2>`, etc., puede tener un título o encabezado al principio, y no tan comúnmente un cierre como "Fin del Capítulo I". Se emplean los elementos `<head>` y `<trailer>` para transcribirlos:

`<head>`

contiene cualquier encabezado, por ejemplo, el título de una sección o el encabezado de una lista o glosario.

`<trailer>`

contiene un título o pie de cierre que aparece al final de una división de un texto.

3.2.1.3 Prosa y verso

Los párrafos indican una división textual y deben marcarse con la etiqueta `<p>`. Por ejemplo:

`<body>`

`<p>`En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lantejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda`</p>`

`</body>`

Existe un gran número de etiquetas para marcar los diferentes componentes de los textos poéticos:

`<lg>`

contiene un grupo de versos que funcionan como una unidad formal, p. ej., una estrofa, refrán, etc.

`<|>`

contiene un único verso, aunque pueda estar incompleto. Tiene el siguiente atributo:

`part`

indica si el verso está métricamente completo o no. Los valores válidos son: F para la parte final de un verso incompleto, Y si el verso está métricamente incompleto, N si el verso está completo o si no se hace referencia a esto, I para la parte inicial de un verso incompleto, M para la parte media de un verso incompleto.

`<sp>`

contiene un parlamento en un texto dramático o en un pasaje presentado como tal, ya esté en prosa o en verso. Tiene el siguiente atributo:

`who`

identifica al hablante insertando un ID.

<speaker>

contiene un tipo especial de encabezado o rótulo, en el que se da el nombre de uno o más hablantes de un texto o fragmento dramático.

<stage>

contiene cualquier tipo de acotación en un texto o fragmento dramático. Tiene el siguiente atributo:

type

indica el tipo de acotación. Se sugieren los siguientes valores *entrance*, *exit*, *setting*, *delivery*, etc.

Por ejemplo, este es el comienzo de un texto poético en el que los versos y estrófas están etiquetadas:

<lg n="1" >

<l>Altazor ¿por qué perdiste tu primera serenidad?</l>

<l>¿Qué ángel malo se paró en la puerta de tu sonrisa</l>

<l>Con la espada en la mano?</l>

<l>¿Quién sembró la angustia en las llanuras de tus ojos como el adorno de un dios?</l>

<l>¿Por qué un día de repente sentiste el terror de ser?</l>

<l>Y esa voz que te gritó vives y no te ves vivir</l>

<l>¿Quién hizo converger tus pensamientos al cruce de todos los vientos del dolor?</l>

<l>.....</l>

</lg>

Las etiquetas <l> indican versos, no líneas tipográficas.

A veces, especialmente en los textos dramáticos, las líneas de verso se dividen entre personajes. La forma más sencilla de etiquetar esto es usar el atributo *part* para indicar que las líneas partidas están incompletas, con se muestra en este ejemplo:

<div l type='Acto' n='II'><head>Acto segundo</head>

[...]

<stage type=italic>Dentro ruido de pastores, y sale Bato.</stage>

[...]

<sp><speaker>CUPIDO</speaker><l part=l>¿Cantáis vos?</l>

<sp><speaker>BATO</speaker><l part=M>Muy mal.</l>

<sp><speaker>CUPIDO</speaker><l part=F>¿Saltáis?</l>

<sp><speaker>BATO</speaker><l part=l>Mucho peor.</l>

<sp><speaker>CUPIDO</speaker><l part=F>¿Hacéis versos?</l>

<sp><speaker>BATO</speaker><l>Sí, señor; mas son perversos.</l>

El mismo sistema se puede aplicar a las estrofas divididas entre dos hablantes:

```
<sp><speaker>ELISA</speaker>
<lg type=estrofa part=I>
<I>Bien te supieras quedar</I>
<I>si me tuvieras amor.</I>
</lg>
<sp><speaker>FABIO</speaker>
<lg part=F>
<I>No hay amor donde hay señor,</I>
<I>ni quedar donde hay mandar.</I>
</lg>
```

El siguiente ejemplo muestra cómo se puede etiquetar un diálogo en prosa en un texto dramático. También muestra el uso del atributo *who* para indicar un código que identifique al hablante de ese fragmento de diálogo:

```
<sp who="MAT">
<speaker>MATILDE</speaker>
<p>Pero, querida tía, ¿es algún delito acaso interesarse en la suerte de Federico? Es tan bueno, tan amable, tan desgraciado... Un joven huérfano, aislado, que nunca ha conocido a sus padres... ¿Usted misma no le recogió en su casa desde su más tierna infancia? ¿No le ha dado usted una educación nada común?...</p>
</sp>
<sp who="ISA">
<speaker>ISABEL</speaker>
<p>Eres muy niña todavía, Matilde. Es verdad que no es un delito querer a Federico; que lo merece, ¡ah! sin duda; pero una joven de tus años debe ocultar sus sentimientos, y...</p>
</sp>
<sp who="MAT">
<speaker>MATILDE</speaker>
<p>Señora...</p></sp>
```

3.2.1.4 Numeración de página y línea

Los finales de página y línea pueden marcarse con los siguientes elementos vacíos.

```
<pb>
    marca el límite entre una página y la siguiente según un sistema de referencia estándar.
<lb>
    marca el inicio de una nueva línea (tipográfica).
```

Estos elementos indican un punto exacto del texto, no un fragmento de este. El atributo global *n* debe emplearse para indicar el número de la página o línea que empieza en la etiqueta. Además, estos dos elementos comparten el siguiente atributo:

ed

indica la edición o versión en la que se da el final de página en ese punto.

Cuando se trabaja con un texto original paginado, suele ser útil indicar esta paginación, aunque sea para simplificar más tarde la corrección. Indicar los finales de línea puede ser útil por el mismo motivo; el tratamiento de los guiones al final de línea en las fuentes impresas requerirá más atención.

Si se marca la paginación, etc., de más de una edición, se debe especificar cada edición en el atributo *ed*, e insertar tantas etiquetas como sean necesarias. Por ejemplo, en el siguiente pasaje indicamos dónde están los fines de página en dos ediciones diferentes (*ED1* y *ED2*)

<p>Pero otra vez, sin quererlo, me aparto de mi objeto, y no ha de ser así, sino que vuelvo atrás para deciros que el señor conde de Rumblar, luego que esparcía su ánimo en aquello del esqueleto, y hablaba por los codos durante una hora, iba en busca de entretenimientos más agradables, y aquí es donde viene como anillo en el dedo la ocasión de nombrar <pb ed=ED2 n='15'> a la Zaina, <lb> porque a eso de las once era cuando penetraba en sus salones el joven de que me ocupó, no acompañado sólo por el citado Mañara, sino también por D. Luis de Santorcaz, que <pb ed=ED1 n='10'> siempre se le unía en la Rosa-Cruz para seguir juntos hasta la madrugada.</p>

Los elementos <pb> y <lb> son casos especiales de la clase general de elementos *milestone* que marcan puntos de referencia dentro de un texto. El TEI Lite también tiene un elemento genérico <milestone>, que no está limitado a casos especiales sino que puede marcar cualquier tipo de punto de referencia: por ejemplo, un final de columna, el inicio de una nueva sección que no se marca de otro modo, etc. Este elemento tiene la siguiente descripción y atributos:

<milestone>

marca el límite entre secciones de un texto, tal y como se indica en los cambios de un sistema de referencia estándar. Tiene los siguientes atributos:

ed

indica la edición o versión a la que hace referencia el *milestone*.

unit

indica que tipo de sección cambia en ese *milestone*.

Los nombres usados para los tipos de unidad (atributo *unit*) y para las ediciones a las que hace referencia (atributo *ed*) pueden elegirse libremente, pero se deben indicar en el encabezado (*teiHeader*).

El elemento `<milestone>` puede usarse en lugar de otros, o los otros pueden emplearse como un conjunto; pero no deben mezclarse arbitrariamente.

3.2.1.5 Marcar frases enfatizadas

Las palabras o frases enfatizadas son aquellas que visiblemente son diferentes del resto del texto, normalmente por un cambio en la fuente, en el tipo de letra o en el color, con el fin de llamar la atención del lector.

El atributo global `rend` puede utilizarse en cualquier elemento cuando sea necesario para indicar cómo debe ser ese énfasis. Por ejemplo, un encabezado que deba ir en negrita podría etiquetarse como `head rend='Bold'`, y uno en cursiva como `head rend='Italic'`.

No siempre es posible o deseable interpretar las razones de dichos cambios en el formato del texto. En tales casos el elemento `<hi>` puede usarse para marcar un fragmento de texto enfatizado sin indicar su estatus o posición.

`<hi>`

marca una palabra o frase gráficamente distinta del texto que le rodea.

En el siguiente ejemplo se recoge aunque no se interpreta el uso de una tipografía distinta para el subtítulo y para el nombre incluido:

`<hi rend=gothic>El ama y la sobrina</hi>`

El lector tendrá la bondad de trasladarse con nosotros a una hermosa quinta del reino de Granada.

`<hi rend=italic>Don Alberto Ludueña</hi>` se había enriquecido en el comercio: y todo su conato, desde que perdió a una esposa querida...

Por otro lado, en los casos en que se sepa con certeza la causa del énfasis podemos usar otros elementos más específicos.

`<emph>`

marca palabras o frases que son enfatizadas por un efecto lingüístico o retórico.

`<foreign>`

identifica una palabra o frase que está en un idioma diferente del del texto de alrededor.

`<mentioned>`

marca palabras o frases mencionadas o citadas, no usadas.

`<term>`

contiene una palabra o expresión que se considera un término técnico.

`<title>`

contiene el título de una obra, ya sea un artículo, libro, periódico, etc., y los subtítulos. Tiene los siguientes atributos:

level

indica si es el título de un artículo, libro, periódico, serie o material inédito. Los valores son: m para un título monográfico (libro, colección u otro objeto publicado como un objeto independiente, incluyendo cada uno de los volúmenes de las obras); s (título de una serie o secuencia); j (título de un periódico); u para los títulos de material inédito (se incluyen las tesis y conferencias, excepto en el caso de que estén publicadas por alguna editorial); a para los títulos analíticos (artículos, poemas u otros objetos publicados como parte de una publicación mayor).

type

clasifica el título de acuerdo con alguna tipología. Algunos valores recomendados son: abbreviated, main, subordinate (para los subtítulos y los títulos de partes) y parallel (para los títulos alternativos, a menudo en otro idioma, por los que también se conoce la obra).

Algunos rasgos (especialmente citas o glosas) se pueden encontrar en un texto o marcado con algún énfasis o con comillas. En estos casos, se deben emplear los elementos `<q>` y `<gloss>` (como se verá en la siguiente sección). Si se debiese mantener el formato, usaríamos el atributo global *rend*.

He aquí un ejemplo de los elementos definidos:

Dafnis y Cloe, más bien que de novela bucólica, puede calificarse de novela campesina, de novela idílica o de idilio en prosa; y en este sentido, lejos de pasar de moda, da la moda y sirve de modelo aún, *mutatis mutandi*, no sólo a Pablo y Virginia, sino a muchas preciosas novelas de Jorge Sand, y hasta a una que compuso en español, pocos años ha, cierto amigo mío, con el título de *Pepita Jiménez*.

Si interpretamos la función del texto enfatizado, la frase quedaría así:

`<title>Dafnis y Cloe</title>`, más bien que de novela bucólica, puede calificarse de novela campesina, de novela idílica o de idilio en prosa; y en este sentido, lejos de pasar de moda, da la moda y sirve de modelo aún, `<foreign>mutatis mutandi</foreign>`, no sólo a Pablo y Virginia, sino a muchas preciosas novelas de Jorge Sand, y hasta a una que compuso en español, pocos años ha, cierto amigo mío, con el título de `<title>Pepita Jiménez</title>`.

Si sólo reproducimos el formato del original, quedaría entonces así:

`<hi rend=italic>Dafnis y Cloe</hi>`, más bien que de novela bucólica, puede calificarse de novela campesina, de novela idílica o de idilio en prosa; y en este sentido, lejos de pasar de moda, da la moda y sirve de modelo aún, `<hi rend=italic>mutatis mutandi</hi>`, no sólo a Pablo y Virginia, sino a muchas preciosas novelas de Jorge Sand, y hasta a una que compuso en español, pocos años ha, cierto amigo mío, con el título de `<hi rend=italic>Pepita Jiménez</hi>`.

3.2.1.6 Citas y características similares

Del mismo modo que los cambios en la tipografía, las comillas se usan convencionalmente para indicar varios rasgos dentro de un texto, entre los que destacan las citas. En la medida de lo posible, recomendamos que el rasgo mencionado sea etiquetado con los siguientes elementos, en lugar de simplemente reproducir las comillas del texto:

`<q>`

contiene aparentemente una cita -una representación de habla o pensamiento marcado como si fuera citado por otra persona (esté firmado); en prosa, suele ser el caso de los parlamentos de un personaje; en los diccionarios, el elemento `<q>` se puede usar para marcar ejemplos de uso reales o inventados. Tiene los siguientes atributos:

`type`

puede usarse para indicar si el texto citado es hablado o pensado, o para describirlo de modo más exacto. Valores recomendados son: `spoken` (para la representación de estilo directo, normalmente marcado con comillas o raya) y `thought` (para la representación del pensamiento, p. ej., un monólogo interno).

`who`

aquí se identifica al hablante de dicho texto en estilo directo.

`<mentioned>`

marca palabras o frases mencionadas o citadas, no usadas.

`<soCalled>`

contiene una palabra o frase en la que el autor o narrador niega su responsabilidad, por ejemplo mediante el uso de comillas o cursiva.

`<gloss>`

marca una palabra o frase que constituye una glosa o definición de alguna otra palabra o frase. Tiene los siguientes atributos:

`target`

identifica la palabra o frase asociada.

Ejemplo de cita:

Con esa profundización en las más íntimas realidades humanas hay que relacionar la permanencia de uno de los elementos trágicos fundamentales, la catarsis, que Bueno entiende como `<q>interior perfeccionamiento.</q>`

El estilo directo interrumpido por un narrador puede representarse simplemente cerrando la etiqueta de cita y abriéndola de nuevo tras la interrupción, como en este ejemplo:

`<p><q>Extraño es</q> - dijo fray Domingo - <q>que afirme todo eso nuestra reverencia por meras conjeturas.</q></p>`

Las citas pueden ir acompañadas por una referencia a la fuente o hablante, utilizando el atributo `who`, según se indique o no en el texto dicha fuente. Por ejemplo:

<q who=Quijote>Gracias doy al cielo por la merced que me hace, pues tan presto me pone ocasiones delante donde yo pueda cumplir con lo que debo a mi profesión, y donde pueda coger el fruto de mis buenos deseos.</q>

Este ejemplo también muestra cómo puede haber citas incluidas dentro de otras citas: un personaje (Wilson) cita a otro personaje (Spaulding).

El creador del texto electrónico debe decidir si las marcas de cita (comillas, guiones,...) son reemplazadas por las etiquetas o si las etiquetas son añadidas y se mantienen las marcas de cita (comillas, guiones,...). Si las marcas de citas son suprimidas del texto, se debe usar el atributo *rend* para indicar cómo venían dichas citas en el texto original.

Como con el resaltado, no siempre es posible ni deseable interpretar de este modo la función de las marcas de cita de un texto. En tales casos, podemos usar la etiqueta <hi rend=quoted> para marcar un texto citado sin indicar su estado o situación.

3.2.1.7 Notas

Todas las notas deben marcarse utilizando el mismo elemento, ya estén impresas como notas al pie, al final de capítulo, marginales o de otro modo:

<note>

contiene una nota o aclaración. Tiene los siguientes atributos:

type

describe el tipo de nota.

resp

indica quién es el responsable de dicha anotación: autor, editor, traductor, etc. El valor puede ser author, editor, etc., o las iniciales del quien ha hecho la anotación.

place

indica dónde aparece la nota en el texto fuente. Tiene los siguientes valores: inline, interlinear, left, right, foot, y end, para aquellas notas que aparecen como párrafos en el cuerpo del texto, entre líneas, al margen izquierdo o derecho, a pie de página o al final de capítulo o volumen, respectivamente.

target

indica el punto de enlace a una nota o el principio del fragmento al que la nota está referida.

targetEnd

apunta al final del fragmento al que la nota hace referencia, si la nota no está incluida en el texto en ese punto.

anchored

indica si el texto muestra el lugar exacto de referencia de la nota.

En la medida de lo posible, el texto de la nota debe insertarse en el texto en el punto donde su identificador o marca aparece en el texto fuente. Esto puede no ser posible por ejemplo con las notas marginales, que no pueden ser insertadas en un lugar exacto. Para simplificarlo, lo adecuado puede ser colocar las notas marginales antes del párrafo. Las

notas también pueden ser colocadas en una división independiente del texto (como notas al final de capítulo, en los libros impresos) y hacer enlaces al lugar donde hacen referencia por medio del atributo *target*.

El atributo *n* se usa para indicar el número o identificador de una nota si es necesario. El atributo *resp* debe usarse consistentemente para distinguir entre las notas del autor y las del editor, si la obra tiene de los dos tipos; de otro modo, el *TEI header* debe indicar de qué tipo son.

Ejemplos:

```
<p>Cuál encomiaba la defensa de Zaragoza <note place=foot n=1>El primer sitio de Zaragoza.</note>; cuál ponía la defensa de Valencia contra Moncey por cima de todos los hechos de armas antiguos y modernos; quién decía que nada podía igualarse a lo del Bruch;...</p>
```

3.2.1.8 Listas

El elemento `<list>` se utiliza para marcar cualquier tipo de *lista*. Una lista es una secuencia de elementos de texto, que pueden estar ordenadas, desordenadas, o puede ser un glosario. Cada elemento puede ir precedido por un título (`<label>`) de elemento (en un glosario, este título es el término definido):

`<list>`

contiene una secuencia de elementos organizada como una lista. Tiene los siguientes atributos:

`type`

describe la forma de la lista. Se recomiendan los siguientes valores: `ordered`, `bulleted` (para las listas con elementos numerados o con letras, y listas con puntos, guiones, etc., respectivamente), `gloss` (para listas consistentes en un conjunto de términos técnicos, cada uno marcado con el elemento `<label>` y acompañado por una definición o explicación marcada con un `<item>`), y `simple` (para las listas con elementos que no llevan números o puntos).

`<item>`

contiene un componente de una lista.

`<label>`

contiene el título asociado al elemento de una lista; en glosarios, indica el término definido.

Cada elemento de la lista se etiqueta con `<item>`. El primer `<item>` puede ir opcionalmente precedido de un `<head>`, que proporciona el título a la lista. La numeración de la lista puede omitirse (y luego reconstruirse, si se desea), usando el atributo *n* de cada elemento (`<item>`). Los siguientes ejemplos son en este sentido equivalentes:

`<list>`

`<head>Una pequeña lista</head>`

```
<item>Primer elemento de la lista.</item>
<item>Segundo elemento de la lista.</item>
<item>Tercer elemento de la lista.</item>
</list>
```

```
<list>
<head>Una pequeña lista</head>
<item n=1>Primer elemento de la lista.</item>
<item n=2>Segundo elemento de la lista.</item>
<item n=3>Tercer elemento de la lista.</item>
</list>
<list>
<head>Una pequeña lista</head>
<label>1</label><item>Primer elemento de la lista.</item>
<label>2</label><item>Segundo elemento de la lista.</item>
<label>3</label><item>Tercer elemento de la lista.</item>
</list>
```

En los casos en que la estructura interna de los elementos de la lista es más compleja, puede ser preferible tratar la lista como una *tabla*. En estos casos hay un conjunto de etiquetas TEI adicional para un etiquetado más específico.

Las listas de cualquier tipo pueden, por supuesto, pueden anidarse dentro de elementos de lista en cualquier nivel de profundidad. Aquí, por ejemplo, un glosario contiene dos elementos y cada uno de ellos es a su vez una sencilla lista:

```
<list type=numeral
<item n="1">
  <list type=simple>
    <item> Primer elemento de la lista anidada.</item>
    <item> Segundo elemento de la lista anidada.</item>
    <item> Tercer elemento de la lista anidada.</item>
  </list>
</item>
<item n="2">
  <list type=simple>
    <item> Cuarto elemento de la lista anidada.</item>
    <item> Quinto elemento de la lista anidada </item>
  </list>
</item>
</list>
```

3.2.1.9 Cita bibliográfica

A menudo es útil distinguir las citas bibliográficas cuando aparecen en textos transcritos para la investigación, en el caso en que tengan un formato apropiado en el texto impreso. Para este propósito existe el elemento `<bibl>`:

<bibl>

contiene una cita bibliográfica no muy estructurada en la que sus componentes pueden estar explícitamente etiquetados o no.

En los casos en que deban distinguirse los componentes de una referencia bibliográfica, debemos usar los siguientes elementos. Puede ser útil marcar como mínimo aquellas partes (como los títulos de artículos, libros y periódicos) que requieren de un formato especial. Los otros elementos se proveen para aquellos casos en que sea interesante destacar estos detalles.

<author>

en una referencia bibliográfica, contiene el nombre del autor(es) de una obra o la *declaración de responsabilidad* primaria de cualquier elemento bibliográfico.

<biblScope>

define el ámbito de una referencia bibliográfica, por ejemplo como una lista de números de página o una subdivisión de una obra mayor.

<date>

contiene una fecha en cualquier formato.

<editor>

declaración de responsabilidad secundaria de un elemento bibliográfico, por ejemplo el nombre de una persona, institución u organización (o varios) que desempeñan la función de editor, compilador, traductor, etc. Tiene el siguiente atributo:

role

especifica la naturaleza de la responsabilidad intelectual. Posibles valores son: *translator, compiler, illustrator*, etc.; el valor por defecto es *editor*.

<imprint>

agrupa información relativa a la publicación o distribución de un elemento bibliográfico.

<publisher>

da el nombre de la organización responsable de la edición o distribución de un elemento bibliográfico.

<pubPlace>

contiene el nombre del lugar donde se publicó un elemento bibliográfico.

<series>

contiene información sobre la serie en que apareció un libro u otro elemento bibliográfico.

<title>

contiene el título de una obra, ya sea artículo, libro, periódico o serie, incluyendo otros títulos alternativos o subtítulos. Tiene los siguientes atributos:

type

dice de algún modo el tipo de título, por ejemplo *main* (principal), *subordinate* (secundario), etc.

level

indica el nivel (*level*) bibliográfico o clase de título. Los valores posibles están descritos en el apartado 6.1 Cambios de tipografía, etc.

Por ejemplo, la siguiente nota editorial podría transcribirse así:

Fue miembro del parlamento por Warwickshire en 1445, y murió el 14 de Marzo de 1470, (de acuerdo con Kittredge, estudios de Harvard 5. 88ff).

Fue miembro del parlamento por Warwickshire en 1445, y murió el 14 de Marzo de 1470, (de acuerdo con < bibl > < author > Kittredge < /author >, estudios de < title > Harvard < /title > < biblScope > 5. 88ff < /biblScope > < /bibl >).

3.2.1.10 Tablas

Las tablas representan un gran reto para cualquier sistema de procesamiento de textos, pero las tablas sencillas aparecen en tantos textos que, incluso en el conjunto simplificado de etiquetas TEI que se presenta aquí, es necesario su marcado. Podemos utilizar los siguientes elementos para este propósito:

<table>

contiene texto mostrado en forma de tabla, con filas y columnas. Tiene los siguientes atributos:

rows

indica el número de filas que tiene la tabla.

cols

indica el número de columnas que tiene cada fila de la tabla.

<row>

es la fila de una tabla. Tiene los siguientes atributos:

role

indica el tipo de información contenida en las celdas de dicha fila. Se sugieren los siguientes valores: label para títulos o información descriptiva, y data para los valores reales de los datos.

<cell>

es la celda de una tabla. Tiene los siguientes atributos:

role

indica el tipo de información contenida en la celda. Se sugieren los siguientes valores: label para títulos o información descriptiva, y data para los valores reales de los datos.

cols

indica el número de columnas que ocupa esa celda.

rows

indica el número de filas que ocupa esa celda.

Por ejemplo, Defoe usa tablas de mortalidad como la siguiente en *Journal of the Plague Year* para mostrar el aumento y el descenso de la epidemia:

```
<table rend="marcoAzul">
```

```
<row>
```

```
<cell rend="encabezadoAzul" >Bloque 1 </cell>
```

```

    <cell rend="encabezadoAzul" >Bloque 2</cell>
    <cell rend="encabezadoAzul" >Bloque 3</cell>
    <cell rend="encabezadoAzul" >Bloque 4</cell>
    <cell rend="encabezadoAzul" >Bloque 5</cell>
</row>
<row>
    <cell cols="4">Secuencia de aprendizaje</cell>
</row>
</table>

```

3.2.1.11 Imágenes y gráficos

No todos los componentes de un documento son necesariamente textuales. Muchos textos suelen a menudo contener diagramas o ilustraciones y hay otros muchos en los que las imágenes y el texto están inseparablemente mezclados, o los recursos electrónicos en los que ambos (texto e imagen) se complementan.

Se puede simplemente indicar la presencia de un gráfico dentro del texto, con una breve descripción de su contenido, usando los elementos descritos en este apartado. Estos mismos elementos se pueden usar también para insertar versiones digitalizadas del gráfico en un documento electrónico.

<figure>

marca el punto en el que debe ser insertado un gráfico en un documento. Tiene el siguiente atributo:

entity

el nombre de una entidad del sistema predefinida que contenga una versión digitalizada del gráfico a insertar.

<figDesc>

contiene una descripción textual de la apariencia o contenido de un gráfico, para usarla cuando se documente una imagen sin mostrarla.

Cualquier información textual acompañada de un gráfico, como un título o encabezado, puede ser incluida dentro del elemento *<figure>*, en un *<head>* y en uno o más elementos *<p>*, tal y como aparecería cualquier texto dentro de un gráfico. Es muy recomendable que se dé una descripción de la imagen, como contenido del elemento *<figDesc>*, para aquellas aplicaciones que no pueden mostrar la imagen y para que el documento sea accesible para los lectores discapacitados visuales. (Este texto no suele considerarse parte del propio documento.)

(La etiqueta de cierre no puede omitirse, incluso aunque el elemento no tenga contenido). Generalmente, un gráfico tendrá, al menos, un título identificativo que debe etiquetarse con el elemento *<head>*. También es conveniente incluir una breve descripción de la imagen, como en el siguiente ejemplo:

```
<figure>
  <head>Don Quijote y los molinos</head>
  <figdesc>Grabado que muestra la lucha que mantuvo
    Don Quijote con unos molinos de viento.</figdesc>
</figure>
```

Cuando está disponible una versión digitalizada del gráfico, es preferible incluirla en el punto apropiado dentro del documento. Los elementos gráficos como las fotografías son típicamente almacenadas en entidades (archivos) separados de aquellos que contienen el texto del documento y utilizan una notación (formato de almacenamiento) diferente. La DTD del TEI Lite soporta gráficos codificados utilizando los estándares CGM, TIFF o JPEG bajo los nombre de notación SGML *cgm*, *tiff* y *jpeg*. Sin embargo, otras notaciones pueden usarse si se añade a la DTD una apropiada declaración de NOTACIÓN.

Cualquier formato que se use para codificar la imagen debe ser incluido dentro del documento del mismo modo. El primer paso es declarar una entidad SGML de un tipo concreto, que especifique un nombre para la entidad, un identificador externo (p. ej., un nombre de fichero) para ella y la notación usada. Por ejemplo, teniendo en cuenta que una imagen digitalizada de "Fuenteovejuna" fue guardada en formato TIFF en el archivo *fuelle.tff*, sería necesaria una declaración de entidad como esta:

```
<!ENTITY fuentePic SYSTEM "fuelle.tff" NDATA tiff>
```

Todas estas declaraciones deben procesarse antes del propio documento SGML; con la DTD del TEI Lite esto puede cumplirse incluyéndolas en un archivo llamado *litedecls.ent* o cualquier otro archivo que tenga el identificador público `-//TEI U5-1995//DTD TEI Lite 1.0 Extensions//EN`.

Con la declaración vigente, todo lo necesario para incluir la imagen digital en el punto apropiado del documento es insertar un valor en el atributo *entity* del elemento `<figure>`:

```
<figure entity=fuentePic>
  <head>Fuenteovejuna</head>
  <figdesc>Portada de la primera edición de Fuenteovejuna.</figdesc>
</figure> [2].
```

3.3 Implantación

Al acercarnos a un libro, necesitamos saber su título, quién lo escribió, qué partes lo integran y, sobre todo, cómo utilizarlo. Sea una obra literaria, o algún libro especializado, se debe conocer su organización para consultarlo con mayor facilidad.

Para descubrir cuál es el contenido de un libro, se requiere tomar en cuenta su estructura, que es la forma como están acomodadas las partes que conforman al libro. Tenemos que respetar dos cosas y ambas en el mismo nivel de importancia, la estructura de un libro y la

estructura del TEI, la unión de ambas dan como resultado la estructura de nuestra edición digital.

El objetivo del TEI es hacer explícitas ciertas características del texto de manera que faciliten su procesamiento automático.

Aplicamos la estructura genérica del TEI, pero ahora integrada con las partes detalladas de cómo vamos a etiquetar cada sección de la edición digital.

Estructura inicial:

```
<TEI.2>
  <teiHeader> [ Información del encabezado TEI ] </teiHeader>
  <text>
    <front> [ materia del front ... ] </front>
    <body> [ cuerpo del texto ... ] </body>
    <back> [ materia del back ... ] </back>
  </text>
</TEI.2>
```

Siguiendo esta estructura vamos a ejemplificar cada uno de los elementos que utilizamos.

<TEI.2>

<teiHeader>. Se encuentra el título general de la edición digital, Indica quién publica el texto electrónico.

<text>. Estructura el contenido del documento. Contiene a los elementos <front>, <body> y <back>, cada uno contiene información de la edición digital.

<front>. Contiene lo siguiente:

- a) Prefacio, introducción y/o presentación. Es el preámbulo, o parte que precede al cuerpo principal del libro. Páginas destinadas al escrito que sirve como preparación para lo que es la materia principal del libro
- b) Página legal o de derechos. Donde se anotan los derechos de la obra:
 - Título
 - Autor
 - El número de la edición y el año
 - Número de reimpresión
 - Año en que se reservaron los derechos
 - Lugar de impresión
 - La casa editorial

- El número internacional normalizador de libros (ISBN)
- Etc.

</front>

<body>

Cuerpo de la obra. Es la parte principal de un libro y puede estar dividido en lecciones, capítulos, etc.

</body>

<back>

Apéndices o anexos. Complementos o suplementos del cuerpo principal del libro constituidos por documentos importantes.

Bibliografía. Listado de las obras consultadas por el autor para la elaboración de su obra.

</back>

<TEI.2>

3.4. Adaptación de DTD

En un DTD se definen con precisión los elementos necesarios para la elaboración concreta de un documento o grupo de ellos estructurados de manera similar.

3.4.1 Descripción de Elementos

Declaraciones tipo elemento.

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

La declaración de cada tipo de elemento deben empezar con "<!ELEMENT" seguidas por el identificador genérico del elemento que se declara.

Por ejemplo:

<!ELEMENT div (head, div|>

En este ejemplo, el elemento <div> puede contener dentro elementos <head> y <div|>, 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:

```
<div>
  <head>...</head>
  <divI>...</divI>
</div>
```

La especificación de contenido puede ser de cuatro tipos:

EMPTY

Puede no tener contenido. Suele usarse para los atributos.

```
<!ELEMENT hi EMPTY>
```

ANY

Puede tener cualquier contenido. No se suele usar, ya que es conveniente estructurar adecuadamente nuestro documento XML.

```
<!ELEMENT div ANY>
```

Mixed

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

```
<!ELEMENT hi (#PCDATA)>
```

```
<!ELEMENT p (#PCDATA | hi)*>
```

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

Element

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

```
<!ELEMENT div (head, divI)>
```

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 p (hi)>`

Esto indica que `<p>` sólo puede contener un solo `<hi>`.

`<!ELEMENT div (head, div1)>`

La coma, en este caso, denota una secuencia. Es decir, el elemento `<div>` debe contener un `<head>` seguido de un `<div1>`.

`<!ELEMENT div (hi | div1)>`

La barra vertical "|" indica una opción. Es decir, `<div>` puede contener o bien un `<hi>` o bien un `<div1>`. El número de opciones es ilimitado, y se pueden agrupar usando paréntesis.

`<!ELEMENT div (head, (div1 | p))>`

En este último caso, el `<div>` debe contener un `<head>` seguido de un `<div1>` o un `<p>`.

Además, cada partícula de contenido puede llevar un indicador de frecuencia: que sigue 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, vamos a ver un ejemplo.

`<!ELEMENT div (head?, (div1+, p)*)>`

En este caso, `<div>` puede tener `<head>` o no (pero sólo uno), y puede tener cero o más conjuntos `<div1><p>`, `<p><p><div1>`, etc.

Declaraciones de lista de atributos

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>Alfredo Reino</de>
<a>Hans van Parijs</a>
<texto idioma="holandés">
Hallo Hans, hoe gaat het?...
</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 atributo de los elementos `<mensaje>` y `<texto>` haríamos lo siguiente:

```
<!ELEMENT table (head, row)>
<!ATTLIST table rend (color | normal) normal>

<!ELEMENT text (#PCDATA)>
<!ATTLIST text type 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 "rend" puede estar en el elemento `<table>` y puede tener el valor "color" o "normal", siendo "normal" el valor por defecto si no especificamos el atributo.

El atributo "type", pertenece al atributo text, y puede contener datos de carácter CDATA. La palabra `#REQUIRED` significa que no tiene valor por defecto, ya que es obligatoria 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 define la etiqueta `` de HTML:

```
<!ATTLIST img
rend CDATA #REQUIRED
alt CDATA #IMPLIED >
```

El atributo "rend" es obligatorio, mientras que el "alt" es opcional (y si se omite, no toma ningún elemento por defecto).

Tipos de atributos

Atributos CDATA y NMTOKEN

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 Diciembre de 1999">
```

```
<!ATTLIST mensaje fecha NMTOKEN #REQUIRED>  
<mensaje fecha="15-12-1999">
```

Atributos enumerados y notaciones

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

```
<!ATTLIST table rend (color | normal) 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">
```

Atributos ID e IDREF

El tipo ID permite que un tipo determinado tenga un nombre único que podrá ser referenciado por un atributo de otro elemento que sea de tipo IDREF. Por ejemplo, para implantar 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. [3].

3.4.2 Aplicación

Partiendo del análisis ejemplificaremos un elemento raíz TEI.2.

```
<!ELEMENT TEI.2 (teiHeader, text)>.
```

Declaramos al elemento TEI2 que es nuestro elemento raíz, el cual contiene los elementos `teiHeader` y `text`, estos elementos están separados por una coma (,) lo que indica que están seguidos de los elementos `<teiHeader>` seguido de un `<text>`.

El siguiente análisis es con los elementos que están dentro del paréntesis, empezamos con `<teiHeader>`

```
<!ELEMENT teiHeader (fileDesc, encodingDesc*, profileDesc*, revisionDesc?)>
```

En este caso, `<teiHeader>` debe contener un `<fileDesc>`, `<encodingDesc>` y `<profileDesc>` tienen asignado el signo (*) lo que indica que su ocurrencia es de cero o más veces y el último elemento `<revisionDesc>` tiene asignado el signo (?) lo que significa que puede o no estar. Los elementos están separados por una coma (,) denotan secuencia entre los elementos declarados.

Ejemplo 1.

```
<teiheader>
  <fileDesc></fileDesc>
  <encodingDesc></encodingDesc>
  <profileDesc></profileDesc>
  <revisionDesc></revisionDesc>
</teiHeader>
```

En este caso existe la ocurrencia de todos los elementos que son contenidos por el elemento `<teiHeader>`

Ejemplo 2.

```
<teiheader>
  <fileDesc></fileDesc>
  <profileDesc></profileDesc>
  <profileDesc></profileDesc>
  <revisionDesc></revisionDesc>
  <revisionDesc></revisionDesc>
</teiHeader>
```

En el código anterior notamos que hace falta el elemento <encodingDesc> y los elementos <profileDesc> y <revisionDesc> se repiten, por que así lo permite la regla.

Estos son solo dos ejemplos, pero existen más posibilidades de la generación de los modelos de contenido permitidos.

Para el elemento <text>

Iniciamos el siguiente análisis, existen 9 casos, los dividimos para que fuera aun más clara la explicación (Fig 3.5).

Caso (1).

Los elementos <index>, <interp>, <interpGrp>, <lb>, <milestone>, <pb>, <gap>, <anchor>, están separados por el signo “ | ” lo que nos indica que pueden estar contenidos cualquiera de los elementos descritos, es decir no existe orden de precedencia y como están dentro de un paréntesis seguido del signo “ * ” indica que los elementos contenidos pueden ocurrir 0 ó más veces.

Caso (2).

Se abre un paréntesis seguido del signo “ ? ” el cual contiene al elemento <front>, por lo tanto su ocurrencia puede ser nula u ocurrir una sola vez.

Caso (3).

Existe un doble paréntesis, primero los elementos <index>, <interp>, <interpGrp>, <lb>, <milestone>, <pb>, <gap>, <anchor>, están dentro de un paréntesis seguido del signo “ * ”, indicando que pueden ocurrir 0 ó más de una vez. El segundo paréntesis contiene al anterior y esta seguido del signo “ ? ”, así que la ocurrencia será de 0 o una vez de los elementos contenidos.

Caso (4).

Este caso esta unido a los dos anteriores, ya que comparten el mismo paréntesis. Este paréntesis tiene asociado el signo “ ? ” y contiene al elemento <front> y a los elementos del caso (3), pero como podemos ver están separados por una coma “ , ” lo que significa que son consecutivos, es decir deben de respetar ese orden de aparición, pero pueden o no ocurrir, pero en el caso de ocurrencia solo pueden aparecer una sola vez.

Caso (5).

Existe otro paréntesis, contiene a los elementos <body> y <group> separados por una línea vertical “ | ”, lo que nos indica que el orden de la ocurrencia es opcional pero al menos uno debe de ocurrir, ya que el paréntesis no tiene asociado ningún signo.

Caso (6).

Este caso es similar al caso (1).

Caso (7).

Hace referencia al elemento <back>, esta contenido por el paréntesis asociado al signo “ ? ”, por lo que su ocurrencia puede ser nula u ocurrir solo una vez.

Caso (8).

Este caso es similar al caso (1).

Caso (9).

Este caso es similar al caso (4).

Ahora analizaremos a cada elemento que contiene el elemento <text>, por que el análisis anterior hizo referencia al como es la ocurrencia de cada uno de ellos, pero sería repetir la explicación anterior en muchos de los casos, así que por tal motivo la omitimos.

```
<!ELEMENT text (  
(index | interp | interpGrp | lb | milestone | pb | gap | anchor)*, ... (1)  
  
( front, ...(2) (index | interp | interpGrp | lb | milestone | pb | gap | anchor)* ...(3) )?, ...(4)  
  
(body | group), ...(5)  
  
(index | interp | interpGrp | lb | milestone | pb | gap | anchor)*, ...(6)  
  
( back, ...(7) (index | interp | interpGrp | lb | milestone | pb | gap | anchor)* ...(8) )? ...(9)  

```

Fig. 3.5 – Declaración del elemento text.

3.5 Estructuras prototipo

En esta sección analizaremos estructuras prototipos apegadas al libro digital tomando como referencia las definiciones hechas por la DTD, adoptada por la Biblioteca Digital, Miguel de Cervantes (www.cervantesvirtual.com).

En la edición digital de documentos hay diferentes necesidades para la publicación, aunque se sabe que prácticamente todo lo impreso puede tener su versión digital cubriendo las necesidades de la versión impresa, se destacan las siguientes publicaciones:

- libros
- revistas
- periódicos
- artículos
- folletos

Partiendo de la investigación acerca de los modelos tecnológicos existentes para el marcado de documentos digitales, elegimos XML, Extensible Markup Language; Lenguaje de Marcado Extensible, bajo el esquema de codificación TEI Lite (Text Encoding Initiative).

Las necesidades de la edición digital dependen de la biblioteca digital o medio que la publicará. El camino a seguir para el desarrollo de la edición digital es variado, se propone la metodología antes descrita, la cual tiene como inicio seleccionar y generar las descripciones necesarias para generar las estructuras de la edición digital.

La estructura prototipo base deriva de las especificaciones de TEI (Text Encoding Initiative). Se puede generar su propia especificación de estructura basada en el estudio y control y de acuerdo a las necesidades del usuario.

Las normas proporcionan un medio para hacer explícitos ciertos rasgos de un texto de tal modo que faciliten el procesamiento de dicho texto con programas informáticos ejecutados desde diferentes máquinas. Esto es lo que se denomina *marcado* o *codificado*.

Cualquier representación textual en un ordenador utiliza algún tipo de marcado; el TEI apareció, por dos razones: la gran variedad de esquemas de codificación mutuamente incomprensibles que abundan en los ámbitos de estudio, y por la creciente variedad de usos que están siendo identificados para los textos en formato electrónico.

Las normas TEI usan el *Standard Generalized Markup Language* (SGML) para definir su esquema de codificación. SGML es un estándar internacional (ISO 8879), usado cada vez más por todas las industrias de procesamiento de información, que posibilita una definición formal de un esquema de codificación, en función de *elementos* y *atributos* y reglas que controlan su aparición en un texto. El uso que hace el TEI del SGML es ambicioso por su complejidad y generalidad, pero fundamentalmente no es diferente del de cualquier otro esquema de marcado en SGML. Por lo tanto, cualquier programa preparado para SGML puede procesar los textos que cumplan el TEI.

Al seleccionar entre varios cientos de elementos SGML definidos por el TEI completo, hemos tratado de identificar un 'conjunto inicial' que recoja los elementos que casi cualquier usuario debería conocer. La experiencia al trabajar con el TEI Lite será de gran valor para entender la DTD del TEI completo y para conocer qué partes opcionales de la DTD completa son necesarias para trabajar con tipos de textos específicos.

Podemos definir los objetivos de este subconjunto del siguiente modo:

- Incluir la mayoría de las etiquetas principales del TEI, ya que estas contienen elementos importantes para todos los tipos de textos y todos los tipos de trabajo de procesamiento de los textos;
- Poder manejar correctamente una variedad de textos bastante amplia, a un nivel de detalle que se dé en la práctica real (tal y como ocurre en, por ejemplo, las bases del Oxford Text Archive);
- Ser útil para la producción de nuevos documentos así como para etiquetar los ya existentes;

- Ser utilizable por la mayoría de los programas de SGML ya existentes;
- Poder ser obtenido de la completa TEI DTD empleando los mecanismos de extensión descritos en las Normas del TEI;
- Ser lo más reducido y sencillo posible siempre y cuando sea coherente con los otros objetivos [4].

3.5.1 Diseño de estructura prototipo

La referencia bibliográfica [2], describe los elementos de TEI Lite que son candidatos para ser usados en la estructura prototipo. La estructura se genera con conocimientos editoriales que son aplicados para interpretar el material impreso, y poder hacer un correcto marcado descriptivo¹.

Una vez que los requerimientos de la edición digital son determinados se genera una estructura abstracta previa al prototipo de uso inicial. En esta estructura general se tienen considerados de forma general los contenedores que hacen coherente el análisis de las necesidades.

La estructura es siempre en apego a las partes que componen una edición impresa. Para citar un ejemplo concreto, los libros tienen por lo general una sección que contiene el índice, presentación, etc., un cuerpo el cual contiene las lecciones, capítulos, bloques etc., y algunos contienen también de una parte final en la que se encuentran los créditos, bibliografía y otros apartados.

El encabezado TEI provee información similar a la de la portada de un texto impreso. Tiene hasta cuatro partes: una descripción bibliográfica del texto electrónico, una descripción de cómo ha sido etiquetado, una descripción no bibliográfica del texto (un *perfil del texto*), y una revisión de su historia (su creación).

Un texto TEI puede ser *individual* si se utiliza para ser una única obra o *compuesto* si servirá para una colección o serie de obras, como por ejemplo una antología). En cualquier caso, el texto puede tener un *front* o *back* opcional. En medio está el *body*, cuerpo de la obra, que, en el caso de un texto compuesto, puede estar formado por *groups*, cada uno conteniendo a su vez más grupos o textos.

Un texto individual se etiquetará siguiendo una estructura genérica como esta:

```
<TEI.2>
  <teiHeader> [ Información del encabezado TEI ] </teiHeader>
  <text>
    <front> [ materia del front ... ] </front>
    <body> [ cuerpo del texto ... ] </body>
    <back> [ materia back ... ] </back>
  </text>
</TEI.2>
```

Tomando en cuenta lo anterior, se genera la estructura genérica que se implantará en las ediciones digitales que se implantarán.

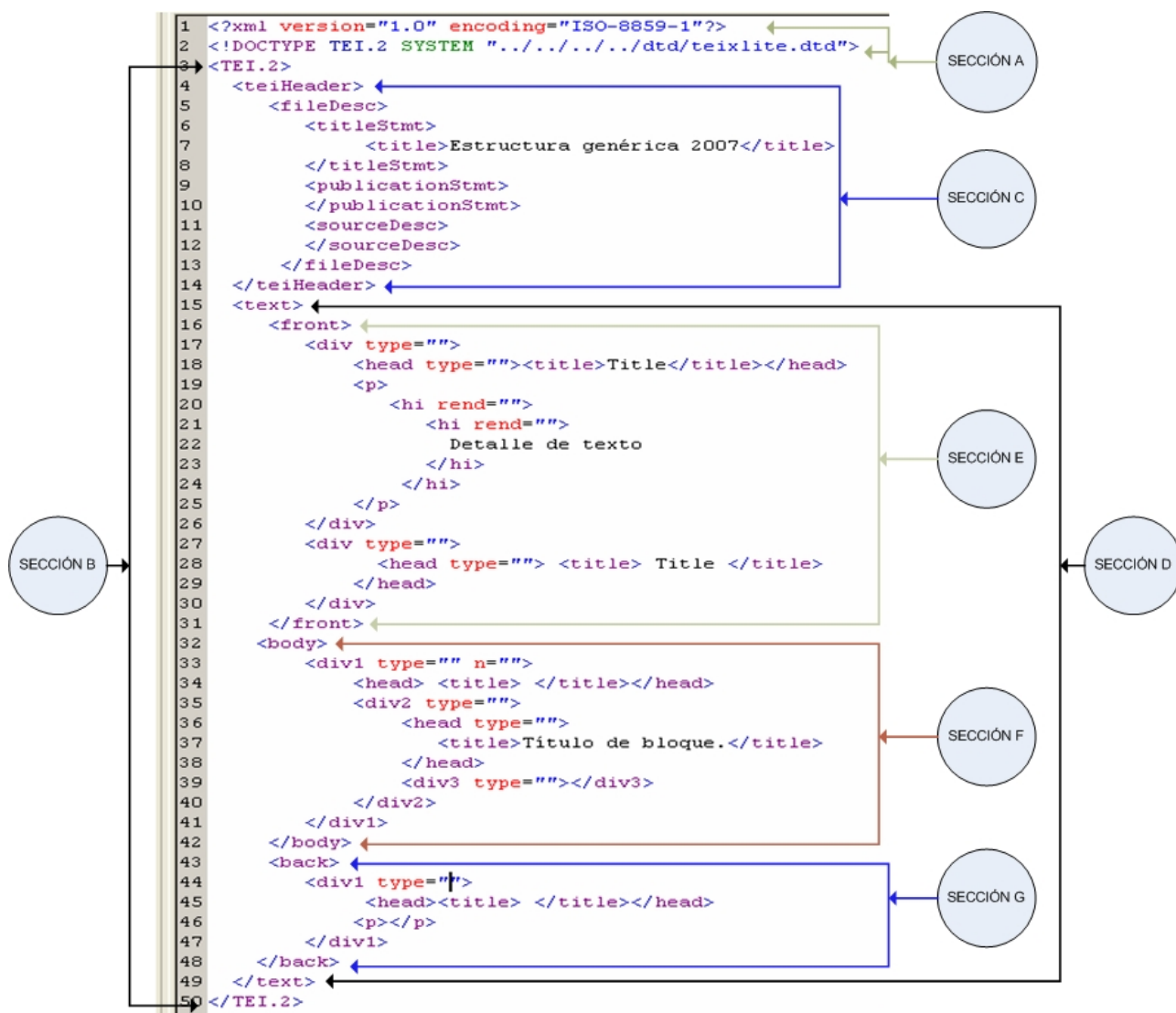


Fig. 3.6 - Diagrama de estructura genérica

En el diagrama anterior se muestra una estructura genérica creada en el lenguaje de marcado XML, ésta estructura se apega a lineamientos editoriales, a continuación explicaremos a detalle cada una de las secciones del *Diagrama de estructura genérica* (Fig. 3.6):

Sección A: Presenta las etiquetas de inicio del documento, que permiten identificar la versión, tipo de codificación y la definición del tipo de documento, que en nuestro caso es TEI Lite (DTD).

Sección B: Como etiqueta global se presenta <TEI.2> que es el contenedor general de cualquier edición digital que nosotros tengamos.

Sección C: Esta sección presenta los elementos que describen de forma general la obra, se podría decir que son equivalentes a los metadatos de una ficha bibliográfica.

Sección D: Etiqueta <text> de nivel global que contiene el “texto” de la edición digital, entendiéndose como texto a la edición digital compuesta.

Sección E: Contiene los materiales previos al comienzo del texto propiamente dicho: introducción, prólogos, dedicatorias, etc.

Sección F: Contiene todo el cuerpo de un texto individual, exceptuando la materia del *front* y del *back*.

Sección G: Contiene los apéndices, títulos complementarios como créditos, bibliografía etc., que siguen a la parte principal de un texto.

Nota: Es importante mencionar que se toman en cuenta elementos, estructura y anidamientos del DTD dichas condiciones se apoyan en expresiones regulares. Las herramientas que validan el documento XML cubren éste requisito.

Teniendo la estructura genérica que permite ser la base de un grupo de ediciones digitales si es que esto es necesario, la estructura prototipo se presenta en el apéndice A2 ya adaptada mediante análisis editoriales y tecnológicos.

La importancia de un correcto etiquetado descriptivo de la edición se refleja en la flexibilidad de transformación que la edición digital necesita en las siguientes etapas.

3.6 Programación de hojas de estilo

En el proceso de la edición digital es muy importante tomar en cuenta la productividad, parte fundamental ocupan las hojas de estilo, que junto con XML, permiten separar el contenido de la presentación.

3.6.1 Codificación dinámica en plantillas XSLT prototipos

Al seleccionar como medio de transformación a la tecnología XSLT (Extensible Stylesheet Language Transformations), se considera el uso de plantillas de transformación relacionadas a los datos XML. El conjunto de instrucciones del lenguaje que hacen posible la transformación es amplio, vamos a mencionar las instrucciones más importantes y las adicionales están mencionadas en la referencia [8].

Para procesar el documento XML mediante el lenguaje de transformación XSLT tiene que existir una relación partiendo desde la etiqueta raíz, para poder ser específicos mediante se recorre el documento XML. Para la plantilla prototipo de estilo, basta con definir estructuras generales sin formato como son, tablas, listas, párrafos, etc.

El objetivo de la plantilla prototipo es generar la primera transformación preliminar que presenta a los elementos con un formato inicial. A continuación se presenta la estructura prototipo con los detalles de cada uno de sus elementos.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:import href="./plantilla_alterna.xsl"/>
  <xsl:output
    doctype-public="-//W3C//DTD HTML 4.01 Transitional//EN"
    doctype-system="http://www.w3.org/TR/html4/loose.dtd"
    method="xhtml"
    media-type="text/html"
    indent="yes"
    encoding="ISO-8859-1" />
```

Fig. 3.6 - Sección I de plantilla prototipo

En esta sección se inicia la definición del tipo de documento, versión y tipo de codificación, en la parte de salida se define el tipo de documento que se va a generar, las definiciones del tipo de documento y parámetros necesarios para la codificación. El producto que se va a generar tiene como formato final un documento HTML 4.01 Transitional, el cual se define en esta parte de la codificación.

```
<xsl:template match="/" >
  <html>
    <head>
      <title>
        <xsl:value-of select="TEI.2/teiHeader/fileDesc/titleStmt/title"/>
      </title>
      <xsl:text disable-output-escaping="yes">
        <![CDATA[<meta http-equiv="Content-Type"
          content="text/html; charset=ISO-8859-1">]]>
        <![CDATA[<meta name="description" content="La Educación
          Intercultural Bilingüe. Cuaderno de Trabajo para las
          Niñas y los Niños de Educación Primaria Indígena.
          Quinto y Sexto Grados. Ch'ol">]]>
        <![CDATA[<meta name="keywords" content="lb01g56">]]>
        <![CDATA[<meta name="language" content="es">]]>
      </xsl:text>
      <xsl:text disable-output-escaping="yes">
        <![CDATA[<link rel="stylesheet" type="text/css"
          href=" ../libros/texto/lb01g56/xml/css/bloques.css">]]>
      </xsl:text>
    </head>
```

```

<body>
  <div class="principal">
    <table class="principal" border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td>
          <table class="navegaSup" border="0">
            <tr>
              <td></td>
            </tr>
          </table>
        </td>
      </tr>
      <tr>
        <td>
          <xsl:apply-templates select="TEI.2/text/descendant::div1[position()=$unidad ]"/>
        </td>
      </tr>
    </table>
  </div>
</body>
</html>
</xsl:template>

```

Fig. 3.7 - Sección 2 de plantilla prototipo

Se indica el punto de inicio de la estructura del documento XML, en la primera línea que presenta la sentencia `match=""`, le indicamos que va a iniciar desde la raíz del documento.

Esta sección contiene la estructura superior del producto final que en nuestro caso es un formato de salida HTML. En este inicio de template definimos las referencias del documento de salida como son CSS y scripts adicionales.

```

<xsl:template match="div1">
  <table width="650" border="0"
    cellpadding="0" cellspacing="0">
    <tr>
      <td class="">
        <xsl:apply-templates />
      </td>
    </tr>
  </table>
</xsl:template>

<xsl:template match="div2">
  <xsl:choose>
    <xsl:when test="$subtema = 1">
      <xsl:apply-templates select="child::head" />
    </xsl:when>
  </xsl:choose>
  <xsl:apply-templates select="child::div3 [position() = $subtema]"/>
</xsl:template>

<xsl:template match="div3">
  <xsl:apply-templates />
</xsl:template>

```

Fig. 3.8 - Sección 3 de plantilla prototipo

Se inicia con la definición de las divisiones generales de la edición digital, el nivel de profundidad está relacionado con la estructura de la edición impresa.

```

<xsl:template match="head">
    <xsl:apply-templates />
</xsl:template>
<xsl:template match="p">
    <xsl:element name="p">
        <xsl:choose>
            <xsl:when test="&#36;class or attribute::rend">
                <xsl:attribute name="class">
                    <xsl:choose>
                        <xsl:when test=" &#36;class ">
                            <xsl:value-of select=" &#36;class " />
                        </xsl:when>
                        <xsl:otherwise>
                            <xsl:value-of select="attribute::rend" />
                        </xsl:otherwise>
                    </xsl:choose>
                </xsl:attribute>
                <xsl:apply-templates />
            </xsl:when>
            <xsl:otherwise>
                <xsl:apply-templates />
            </xsl:otherwise>
        </xsl:choose>
    </xsl:element>
</xsl:template>

<xsl:template match="h1">
    <xsl:element name="span">
        <xsl:attribute name="class">
            <xsl:value-of select="attribute::rend" />
        </xsl:attribute>
        <xsl:apply-templates />
    </xsl:element>
</xsl:template>

<xsl:template match="lb">
    <xsl:text disable-output-escaping="yes">
        <![CDATA[<br>]]>
    </xsl:text>
</xsl:template>

<xsl:template match="list">
    <xsl:element name="ul">
        <xsl:attribute name="class">
            <xsl:value-of select="attribute::type" />
        </xsl:attribute>
        <xsl:apply-templates select="child::item" />
    </xsl:element>
</xsl:template>

```



```

<xsl:template match="list/child::item">
  <xsl:element name="li">
    <xsl:attribute name="class">
      <xsl:value-of select=" ancestor::list/attribute::type " />
    </xsl:attribute>
    <xsl:apply-templates />
  </xsl:element>
</xsl:template>

<xsl:template match="lg">
  <xsl:element name="table">
    <xsl:attribute name="class">
      <xsl:value-of select="attribute::rend" />
    </xsl:attribute>
    <tr>
      <td>
        <xsl:apply-templates select="child::1" />
      </td>
    </tr>
  </xsl:element>
</xsl:template>

<xsl:template match="l">
  <xsl:element name="p">
    <xsl:attribute name="class">
      <xsl:value-of select="attribute::rend" />
    </xsl:attribute>
    <xsl:apply-templates />
  </xsl:element>
</xsl:template>

<xsl:template match="table">
  <table class="transparente"
    border="0" cellpadding="5" cellspacing="0">
    <xsl:apply-templates select="row"/>
  </table>
</xsl:template>

<xsl:template match="row">
  <xsl:element name="tr">
    <xsl:apply-templates select="cell"/>
  </xsl:element>
</xsl:template>

<xsl:template match="cell">
  <xsl:attribute name="class">
    <xsl:value-of select="attribute::rend" />
  </xsl:attribute>
</xsl:template>

```

```
<xsl:template match="figure">
  <table border="0" class="completa" cellpadding="0">
    <tr>
      <td>
        <img class="borde">
          <xsl:attribute name="src">
            <xsl:value-of select="concat ($baseImg, attribute::n )"/>
          </xsl:attribute>
          <xsl:attribute name="alt">
          </xsl:attribute>
        </img>
      </td>
    </tr>
  </table>
</xsl:template>
</xsl:stylesheet>
```

Fig. 3.9 - Sección 5 de plantilla prototipo

Es la parte en la que se definen cada uno de los elementos descriptivos y es más profunda conforme las iteraciones de la edición digital avanzan. Esta sección se va definiendo con la plantilla prototipo y la documentación visual de la edición digital.

3.6.2 Codificación estática de estilo en CSS

Un elemento necesario para el producto HTML que se va a generar es la hoja de estilo estática CSS, que hace posible que la parte visual tenga el soporte necesario para cumplir con los lineamientos de las definiciones visuales de la edición digital en formato HTML. La hoja de estilo estática se codifica por secciones específicas dependiendo de la estructura de la edición.

Un aspecto necesario para la definición de la CSS es el análisis visual de la edición impresa, es el punto de referencia para la codificación a lo largo de todo el proceso de creación y punto necesario para la codificación de elementos básicos de la CSS.

Los patrones que se siguen para la codificación de la CSS son contemplando de inicio posiciones de los elementos, posteriormente se consideran los estilos de cada uno de los elementos. Es importante recordar que el proceso al ser iterativo permite ir adecuando la codificación según sean las necesidades de presentación de la edición digital.

Como ejemplo se mostrarán las secciones básicas que se codifican en una estructura prototipo las cuales pueden ser vistas en el anexo [A1](#).

3.7 Módulo de transformación

Los elementos necesarios para la creación del módulo de transformación son presentados en dos partes, una la parte del soporte de publicación, la cual sería el Servidor Web. La

instalación se hace con los parámetros que tiene automáticamente asignado el paquete de instalación.

Otro aspecto que se debe de considerar es el lenguaje de soporte para el desarrollo de la interfaz y motor del módulo de transformación se usará Java como antes se mencionó en la elección de herramientas.

3.7.1 Servidor Web: Tomcat

Tomcat empezó siendo una implantación de la especificación de los servlets comenzada por James Duncan Davidson, que trabajaba como arquitecto de software en Sun y que posteriormente cooperó a hacer el proyecto open source y en su donación a la Apache Software Foundation.

Duncan Davidson inicialmente esperaba que el proyecto se convirtiese en open source y dado que la mayoría de los proyectos open source tienen libros de O'Reilly asociados con un animal en la portada, quiso ponerle al proyecto nombre de animal. Eligió Tomcat (gato), pretendiendo representar la capacidad de cuidarse por sí mismo, y de ser independiente.

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Entorno

Tomcat es un servidor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Como Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Estado de su desarrollo

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software Licence. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las

6.x, que implementan las especificaciones de Servlet 2.4 y de JSP 2.0. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

Estructura de directorios

La jerarquía de directorios de instalación de Tomcat incluye:

- bin - arranque, cierre, y otros scripts y ejecutables
- common - clases comunes que pueden utilizar Catalina y las aplicaciones web
- conf - ficheros XML y los correspondientes DTD para la configuración de Tomcat
- logs - logs de Catalina y de las aplicaciones
- server - clases utilizadas solamente por Catalina
- shared - clases compartidas por todas las aplicaciones web
- webapps - directorio que contiene las aplicaciones web
- work - almacenamiento temporal de ficheros y directorios

Características del producto

Tomcat 3.x (distribución inicial)

- Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
- recarga de servlets

Tomcat 4.x

- Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
- contenedor de servlets rediseñado como Catalina
- motor JSP rediseñado con Jasper
- conector Coyote
- Java Management Extensions (JMX), JSP Y administración basada en Struts

Tomcat 5.x.

- implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- recolección de basura reducida
- capa envolvente nativa para Windows y Unix para la integración de las plataformas [\[5\]](#)

En la transformación de la edición digital se configurará tomando en cuenta las instalaciones por default. Los directorios en los cuales se trabajará serán los siguiente:

- webapps/ROOT: directorio donde se almacenará el módulo de transformación
- common/ib: directorio que contiene los paquetes de clases necesarios para el módulo

El Servidor Tomcat nos sirve exclusivamente como plataforma para publicar el módulo de transformación, no es necesario configurar librerías o elementos alternos a la instalación original para que funcione el módulo de transformación.

3.7.2 Lenguajes de soporte para el desarrollo del modulo

JAVA y XSLT son tecnologías muy diferentes que en lugar de competir entre ellas, se complementan con una unión que permite aprovechar los recursos de portabilidad y eficiencia, dando como resultado productos robustos y escalables en los desarrollos actuales.

Las ventajas de Java son su portabilidad, su vasta colección de bibliotecas estándar y su amplia aceptación por parte de la mayoría de las empresas. Pero tal vez no sea la mejor tecnología para convertir simplemente archivos XML a otro formato como XHTML o Lenguaje de Marcado Inalámbrico (WML). XSLT está diseñado especialmente para realizar dichas transformaciones. Con XSLT, se pueden transformar los datos XML a cualquier otro formato de texto, como HTML, XHTML, WML e incluso formatos inesperados como el código fuente JAVA-.

3.7.2.1 Motor de procesamiento del módulo

El motor de transformación tiene como base el API de Java JAXP (Java API for XML Processing). La principal razón detrás de JAXP es precisamente la interoperabilidad "Write once run everywhere" (Escribalo una vez ejecútelo en todos lados) de Java, al existir diversos "Parsers" escritos en Java lo anterior resultaría falso y por esta razón Sun ideó JAXP; JAXP no es "Parser" sino un grupo de clases integradas a todo "Parser" , esto garantiza que un programa escrito hacia un "Parser" con soporte JAXP puede ser migrado a otro sin ningún problema.

JAXP sirve para procesar los datos XML usando aplicaciones escritas en el lenguaje de programación Java. JAXP soporta los procesadores (parsers) estándar SAX y DOM, de modo que se puede elegir entre procesar los datos como un flujo de eventos o construir una representación objeto de éstos. JAXP también soporta el estándar XSLT, dando control sobre la representación de los datos, así como la conversión de los datos a otros formatos, tales como HTML. Los principales APIs de JAXP están definidos en el paquete javax.xml.parsers. Este paquete contiene dos clases: SAXParserFactory y DocumentBuilderFactory, los cuales te proporcionan un procesador SAX (SAXParser) y un constructor de documentos (DocumentBuilder). El constructor de documentos crea un compilador DOM de documentos.

Paquetes:

Las librerías que definen a las APIs de SAX, DOM y XSLT son: javax.xml.parsers: Estas APIs proveen una interfaz común para diferentes proveedores de los procesadores (parsers) SAX y DOM org.w3c.dom: Define la clase Document (un DOM), también como todos los componentes de un DOM org.xml.sax: Define las APIs básicos de SAX

javax.xml.transform: Define las APIs para XSLT que permiten transformar XML a otros formatos

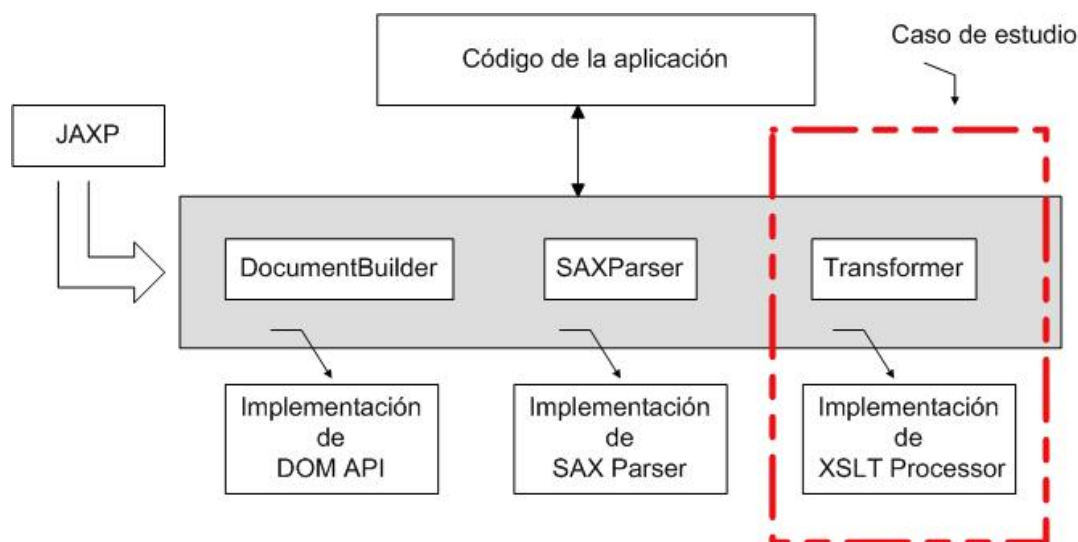


Fig. 3.10 - Diagrama de arquitectura de JAXP 1.1 [6]

Como se puede observar JAXP, cubre varias vías para el desarrollo de la aplicación, dependiendo de las necesidades de procesamiento se combinan o manejan por separado.

3.7.2.2 Paquetes de XSLT

Las APIs de XSLT son las que se utilizarán debido a las necesidades del tipo de transformación y están definidas en los siguientes paquetes:

Paquetes de XSLT	
Paquete	Descripción
javax.xml.transform	Define las clases TransformerFactory y Transformer, las cuales se usan para obtener un objeto capaz de realizar transformaciones
javax.xml.transform.dom	Clases que crean una entrada (fuente) y una salida (resultado) a partir de un objeto DOM
javax.xml.transform.sax	Clases para crear una entrada (fuente) a partir de un procesador SAX y una salida a partir de un manejador de eventos SAX
javax.xml.transform.stream	Clases para crear una entrada (fuente) y una salida (resultado) a partir de un objeto I/O stream

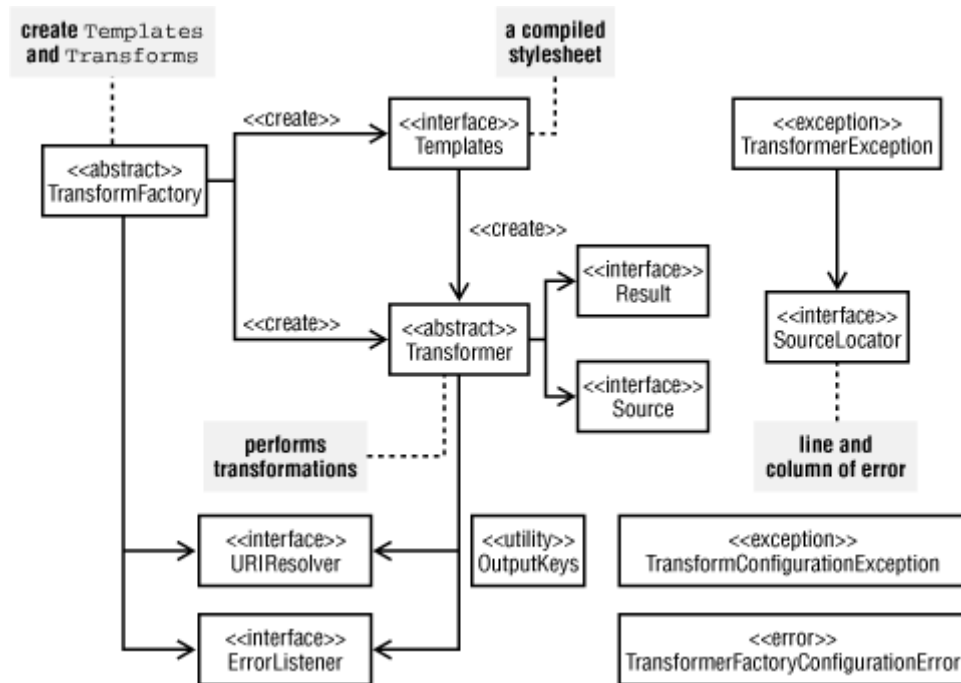


Fig. 3.11 - Diagrama de clases para javax.xml.transform [7]

3.7.3 Procesamiento XSLT con Java

El núcleo central del procesamiento de la información procesada se basa en el motor que está programado en el lenguaje JAVA usando el API JAXP, que ya se mencionó y usando como parte visual la un interfaz JSP (Java Server Page).

La configuración es la parte más simple del proceso, en el siguiente diagrama se muestran los elementos a los cuales dicha interfaz tiene acceso para la configuración inicial.

ELEMENTOS DEL MÓDULO DE TRANSFORMACIÓN

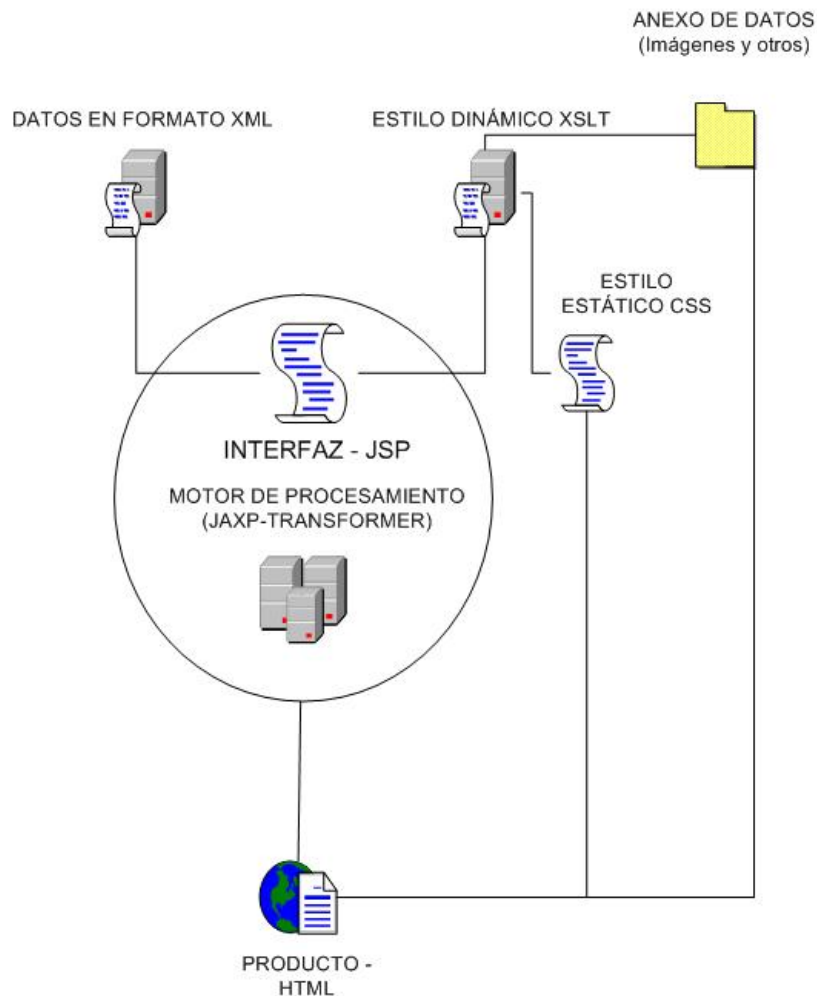


Fig. 3.12 – Elementos del módulo de transformación

Es importante mencionar que la edición digital tiene elementos que se van desarrollando paralelamente (Fig. 3.12).

El módulo de transformación genera automáticamente el índice de la edición digital una vez que todos sus elementos se encuentren definidos en el archivo de XML, posteriormente la hoja de estilo dinámica (XSLT) y el motor los transformen correctamente.

En la creación real, se trabaja mediante un procesamiento evolutivo, en el cual cada transformación es un “render” o visualizaciones parciales según las necesidades de la edición digital. Se colocan elementos y se prueba la capa para ver si es correcta la transformación hasta llegar a la edición digital terminada.

BIBLIOGRAFÍA DEL CAPÍTULO

- [1] TEI - <http://www.bib.uc3m.es/~nogales/cursos/tei.html> - Visitado junio de 2007
- [2] http://www.tei-c.org/Lite/teiu5_sp.html - Visitado en junio de 2007
- [3] DTD - <http://www.ulpgc.es/otros/tutoriales/xml/DTD.html> - Visitado en junio de 2007
- [4] http://www.tei-c.org/Lite/teiu5_sp.html#IDI - Visitado en junio de 2007
- [5] <http://es.wikipedia.org/wiki/Tomcat> - Visitado en Julio de 2007
- [6] http://www.onjava.com/pub/a/onjava/excerpt/java_xslt_ch5/index.html?page=3 - Visitado en julio de 2007
- [7] http://www.onjava.com/onjava/excerpt/java_xslt_ch5/graphics/jxt_0502.gif - Visitado en julio de 2007
- [8] <http://www.w3.org/TR/xslt-> Visitado en julio de 2007

ⁱ Se busca que el profesional que interviene tenga un nivel de licenciatura en letras, comunicación, filosofía, filología, bibliotecología o ramas afines.

Capítulo 4

Transformación

Los procesos que se realizan por motores específicos tienen sentido cuando se muestra detalladamente su funcionamiento, este capítulo presenta el inicio de ese funcionamiento, tratando de explicar los elementos claves a un nivel de programación elemental.

4.1 Configuración de elementos de transformación

Es importante saber que una configuración cercana a la óptima nos presentará menos complicaciones a lo largo del proceso. Si en el proceso a largo plazo contemplamos el factor escalabilidad, esta configuración será más que importante.

4.1.1 Instalación y configuración

Los procedimientos de instalación necesarios para la transformación de los elementos de la edición digital, son los que están automáticamente configurados en los diferentes paquetes que se usan para la transformación.

La base de la operación de la transformación está compuesta por un servidor Tomcat 5.0 usado como Servidor Web que opera con JRE 1.4 que contiene la instalación de JAXP 1.3, y dentro de la API JAXP encontramos el núcleo XALAN 2.7.0 para la transformación XSLT de la edición digital. [1]

Esta configuración soporta XSLT 1.0 y XML Path Language (XPath) Versión 1.0

La localización de los recursos es considerada la configuración para la edición digital y se presenta a continuación el esquema de directorios usados (*Fig. 4.1 esquema de directorios*).

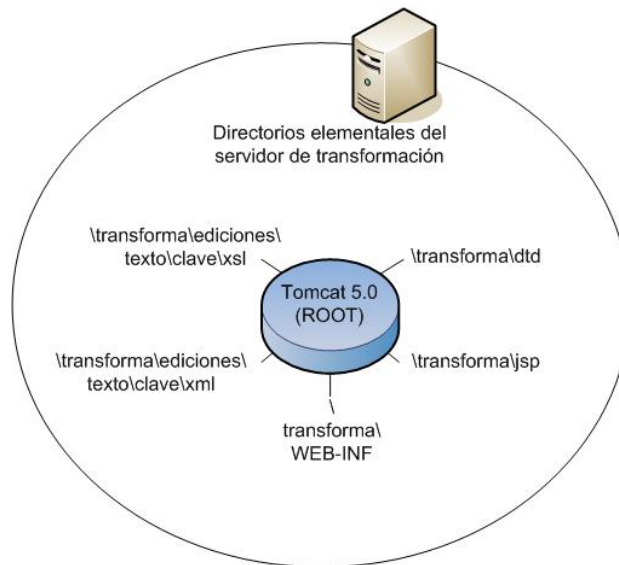


Fig. 4.1 – Esquema de directorios

Conteniendo los directorios de las rutas lo siguiente:

\transforma\dtd: Almacena la *DTD* (Document Type Definition), documento que tiene la estructura manejada por el XML de la edición digital. El *DTD* está basado en la *TEI* y puede contener ciertas modificaciones elementales siempre apegadas al *DTD Lite* de *TEI* original.

\transforma\jsp: Contiene las *Java Server Pages* (JSP's) que tienen programado la referencia al motor de transformación, así como la interfaz de configuración de datos.

\transforma\WEB-INF: Contiene los archivos de información de la aplicación que se encuentre en ese nivel de directorio, algunos ejemplos de archivos que contiene son: paquetes exclusivos para esa aplicación, configuraciones de directorios, etc.

\transforma\ediciones\texto\clave\xml: Contiene los archivos de recursos de la edición digital, el archivo fuente *XML* y algunos recursos generales.

\transforma\ediciones\texto\clave\xsl: Contiene los archivos de hojas de estilo dinámicas (*XSLT*) necesarias para la transformación de la edición digital.

La instalación de los editores no son considerados como parte fundamental de la transformación de la edición digital. Los procesos que no inciden en la transformación directa como lo son: diseño gráfico y cuidado de la edición no son relevantes para el proceso de transformación, si no como generadores de productos necesarios para la transformación.

La composición y distribución de estos directorios puede variar, siempre y cuando, la interfaz de configuración de elementos del módulo de transformación lo permita o se generen en el código de la transformación las rutas necesarias.

4.1.2 Núcleo de transformación

Las instalaciones y configuraciones anteriores son el primer paso del ambiente necesario para la transformación dinámica de la edición digital.

Es en este momento en el que el motor interno de transformación mezcla cada uno de los elementos para generar una capa de la edición digital. El concepto de *capa* se genera por cada sección de la edición digital.

La idea fundamental de generar la transformación por capas o iteraciones esta basada en la visualización inmediata de lo creado, con esto se controlan los cambios inmediatos y se evita generar consecuencias negativas en cascada.

En el siguiente diagrama (Fig. 4.2 – *Capa de transformación*) se muestra el concepto de la transformación para una iteración del desarrollo de la edición digital.



Fig. 4.2 – Capa de transformación

La parte que controla la transformación es sin duda el *script* del lado del servidor, en nuestro caso se usa JSP. Una vez que el motor de transformación da la orden de generar el documento mediante los parámetros, entra en acción la hoja de estilo dinámica de transformación (XSLT), como ya explicamos la XSLT es la hoja encargada de tener las extensiones necesarias que operan controladas por los parámetros recibidos, usando principalmente XPATH.

Los parámetros que el servidor recibe son asignados por el usuario dependiendo del nivel o nodo seleccionado para ser transformado. Cada uno de los parámetros puede indicar un tema, unidad, sección, índice, etc. La elección correcta de estos parámetros es el resultado de un correcto análisis de la estructura del ejemplar.

Siempre es importante recordar que la edición impresa lleva un orden de páginas, la edición digital puede llevarlo o no, eso depende del análisis editorial, el caso que se presenta maneja un orden por temas, según la longitud que sea adecuada para la correcta lectura en pantalla.

El formato de salida será un documento en HTML, respetando las reglas de W3C para HTML 4.01 Transitional. Es posible generar documentos en otro formato modificando la hoja de estilo dinámica (XSLT) para indicar el tipo de salida del documento.

4.2 Uso del módulo de transformación

Los documentos necesarios para la transformación serán seleccionados por el módulo de transformación para generar las rutas de acceso a los mismos, el proceso tiene la ventaja de ser iterativo, por lo mismo se prueba en cada iteración, esto permite verificar los avances gradualmente sin necesidad de esperar a concluir cada uno de los recursos de la edición digital. El producto final de la transformación será un archivo en formato HTML por iteración.

4.2.1 Aplicación del módulo de transformación a recursos

Los recursos que el módulo necesita para generar la primera ruta *índice* son los siguientes:

- *Directorio del documento XML*: Este archivo contiene la estructura de la edición digital, la clave se toma de su nombre.
- *Directorio de hoja de estilo dinámica XSLT*: Uno o varios archivos se tomarán del directorio donde se encuentre la hoja de estilo dinámica seleccionada.
- *Directorio de hoja de estilo estática CSS*: Uno o varios archivos se tomarán del directorio donde se encuentre la hoja de estilo estática seleccionada.
- *Configuración del tipo de estructura*: Es posible que la edición digital no cuente con alguna de las partes comunes como son preliminares, bibliografía, etc. Por tal motivo es posible configurar de inicio alguna de las propuestas para no trabajar una estructura completa.

En la figura siguiente (*Fig. 4.3 – Configuración de datos*), se muestra un caso real de una configuración, y el resultado de generar el script (JSP) necesario para comenzar la transformación gradual de la edición digital.

Por establecer un orden en el inicio de transformación se recomienda comenzar por el índice, debido a que es posible, en este punto, ver el avance del marcado de la edición digital en bloques mayores.

MÓDULO DE TRANSFORMACIÓN DINÁMICA	
<i>CONFIGURACIÓN DE DATOS</i>	
Directorio del documento XML:	<input type="text" value="forma\libros\texto\lb01g56\xml\lb01g56.xml"/> <input type="button" value="Examinar..."/>
Directorio de hoja de estilo dinámica XSLT:	<input type="text" value="ansforma\libros\texto\lb01g56\xsl\indice.xsl"/> <input type="button" value="Examinar..."/>
Directorio de hoja de estilo estática CSS:	<input type="text" value="ma\libros\texto\lb01g56\xml\css\indice.css"/> <input type="button" value="Examinar..."/>
<i>CONFIGURACIÓN - EDICIÓN DIGITAL</i>	
Seleccionar la estructura del archivo	<input type="text" value="Index,Preliminares,Unidad,Bic"/> <input type="button" value="v"/>
<input type="button" value="Generar"/>	
LIGA RESULTADO	
http://localhost:8080/transformaljsp/lb01g56.php?clave=lb01g56&seccion=indice	

Fig. 4.3 – Configuración de datos

De forma interna el módulo opera con dos elementos, la configuración de datos, explicada ya, y el motor de transformación, que es el encargado de generar el script de forma dinámica usando los parámetros seleccionados. Y por cada clic del usuario se generará una iteración produciendo un documento en HTML.

El motor de transformación, contiene la lógica necesaria para obtener un script adecuado a cada edición digital. En él se contemplan elementos como:

- Formato de tipo archivo (HTML)
- Formato de navegación (índice, preliminares, etc.)
- Generación física de archivos y/o dinámica

- Lógica de parámetros para seccionar el documento en la transformación progresiva
- Formato de nombres de archivos (u01t02.html, p01t02.html, etc.)

La importancia de la configuración de los elementos está en el orden para iniciar una publicación, sí bien la transformación puede iniciar manualmente, el módulo garantiza un inicio correcto generando rutas en los scripts de transformación y verificando de forma automática rutas de imágenes, hojas de estilo, documento base XML.

4.3 Elementos interdisciplinarios para la transformación

El marcado debe cumplir ciertas reglas de construcción básicas, aquellos documentos que cumplen se pueden denominar como “bien formados”, pueden entonces ser llamados “documentos XML”. Estas restricciones que todos los documentos XML deben cumplir aseguran un mínimo de coherencia y precisión a la hora de analizarlos o leerlos y simplifican los programas que deben procesar XML, de manera que todos los programas que procesan XML comprobarán que se cumplan estas restricciones y no trabajarán con documentos que no las cumplen.

4.3.1 Documento XML

La declaración del documento XML debe ser la primera línea del documento, incluirla en otro lugar supone un error de formación que impedirá que el documento sea procesado.

Hay que tener en cuenta que estas distintas declaraciones, siguen un orden estricto, no se pueden especificar en otro orden. Primero se especifica la declaración XML, después es necesario declarar la versión para la que se elaboró el documento, por último la codificación por especificación de XML, no admite acentos ni otros caracteres comunes en español, para incluirlos hay que utilizar una codificación de 8 bits, llamada ISO-8859. La declaración completa quedará como aparece abajo (Fig. 4.4).

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

Fig. 4.4 – Xml versión

La declaración del tipo de documento siempre comienza por la cadena “<!DOCTYPE Nombre_del_tipo SYSTEM “Ubicación_de_la_DTD”>”, luego la declaración queda como aparece abajo (Fig. 4.5).

```
<!DOCTYPE TEI.2 SYSTEM "../../../dtd/teixlite.dtd">
```

Fig. 4.5 – Doctype

La siguiente estructura (Fig. 4.6) muestra lo que hemos explicado en los capítulos anteriores acerca de la forma de etiquetar la información, podemos visualizar de forma general los elementos que contiene nuestro documento.

```
<TEI.2>
  <teiHeader>
  </teiHeader>
  <text>
    <front>
    </front>
    <body>
    </body>
    <back>
    </back>
  </text>
</TEI.2>
```

Fig. 4.6 – Estructura TEI.2

Lo que se ejemplifica abajo es cómo quedará etiquetado la parte del front (Fig. 4.7). Existen dos divisiones, las podemos diferenciar con el atributo “type”; generalmente el valor del atributo hace referencia al contenido, y esto se puede ver en la sección de *front*. Una división hace referencia a la página legal o de derechos y la otra a la presentación, ambas poseen un encabezado y un cuerpo de texto.

```
<front>
  <div type="legal">
    <head type="indice">
      <title>Hoja Legal</title>
    </head>
    <p>texto</p>
  </div>
  <div type="presentacion">
    <head type="indice">
      <title>Presentación</title>
    </head>
    <p>texto</p>
  </div>
</front>
```

Fig. 4.7 – Estructura front

4.3.2 Procesamiento de imágenes

Las diferentes imágenes que serán utilizadas, se analizan para determinar qué tipo de extensión será el óptimo, ya que la extensión afecta el tamaño y calidad del tipo de imagen, afectando el tiempo de transferencia, se espera que sea el mínimo, los aspirantes son gif y jpg.

Utilizamos imágenes tipo GIF cuando la imagen tiene transparencias, es decir cuando comparte el mismo fondo de la página, además de que es un formato comprimido y reduce el tiempo de transferencia a través de la Web (Fig. 4.8 – Formato web).

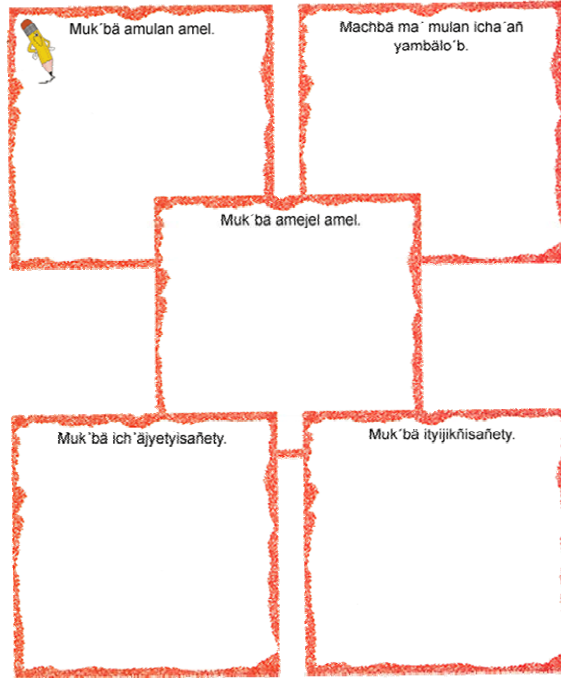


Fig. 4.8 – Formato web

Pero no todas las imágenes comparten el mismo fondo de la página, debemos de tener alternativas, para este caso el tipo de imagen será JPG, a veces será necesario trabajarlas como fotografía digital, debido a que a veces esto permite comprimirlas sin necesidad de bajar su calidad en la resolución y dicha compresión no afecta a la calidad de las imágenes, siguen siendo de alta calidad ya que ese detalle no es susceptible al ojo humano (Fig. 4.9 – Imagen modificada).



Fig. 4.9 – Imagen modificada

La forma en que están nombradas las imágenes tiene una razón, pensemos en el caso de que la cantidad sea de más de 100, si iniciamos nombrando a cada una por la información que muestra sería complicado. La solución es nombrar a la imagen consecutivamente dependiendo del orden de aparición.

4.3.3 Uso de hojas de estilo

Las hojas de estilo permiten dar los márgenes, sangrías, diferentes tamaños, colores de fondo, características de formato dadas por supuesto por los diseñadores pero que no están soportadas directamente por el conjunto de etiquetas estándar.

La definición de un estilo o regla de estilo contiene dos elementos principales:

- El selector: Es el elemento al cual se va a aplicar el estilo.
- La declaración: Es la descripción de las propiedades que conforman el estilo.

La declaración del selector quedaría de la siguiente forma:

```
Nombre_etiqueta {  
    Nombre_propiedad_1: Valor_1;  
    Nombre_propiedad_1: Valor_1;  
    Nombre_propiedad_1: Valor_1;  
    .....  
    Nombre_propiedad_1: Valor_1;  
}
```

Dicha declaración empieza con el nombre de la etiqueta, tratando de respetar una relación hacia el elemento al que se va a aplicar, ya que esto ayudará a que sea más claro el código. Después se abren un par de llaves “{ }” que son las encargadas de contener a la lista de propiedades; en cuanto a las propiedades estas son declaradas “nombre_propiedad : valor ;”,

4.4 Revisión de los elementos generados de la transformación

Primero se hace una revisión de los productos que han sido generados en el punto 4.2.1, los cuales son depositados en la carpeta ya configurada, se verifica que se generaron correctamente, y que los nombres son los correctos (Ver Fig 4.11 - Documentos HTML).



Fig. 4.11 – Documentos HTML

Podemos ver el conjunto de elementos incluidos, vemos que la página tiene un fondo de color y posee un título, que nos servirá como página de inicio mostrando información del contenido de la lección.



Fig 4.11 - Inicio de lección

Esta es la apariencia final (Fig. 4.11 – Inicio de lección), del inicio de la lección.

4.5 Validación de los productos generados

Hablar de *Accesibilidad Web* es hablar de un acceso universal a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios. Y al hablar de accesibilidad se tiene que hablar de validación para poder lograrla.

Con esta idea de accesibilidad nace la Iniciativa de Accesibilidad Web, conocida como WAI (Web Accessibility Initiative). Se trata de una actividad desarrollada por el W3C, cuyo objetivo es facilitar el acceso de las personas con discapacidad, desarrollando pautas de accesibilidad, mejorando las herramientas para la evaluación y reparación de accesibilidad Web, llevando a cabo una labor educativa en relación a la importancia del diseño accesible de páginas Web, y abriendo nuevos campos en accesibilidad a través de la investigación en este área. Para lograr la meta de este punto tenemos que usar una herramienta para facilitarnos el trabajo basta con tener alguna que cumpla con la WAI.

4.5.1 verificación del reporte generados

La idea principal radica en hacer la Web más accesible para todos los usuarios independientemente de las circunstancias y los dispositivos involucrados a la hora de acceder a la información. Partiendo de esta idea, una página accesible lo será tanto para una persona con discapacidad, como para cualquier otra persona que se encuentre bajo circunstancias externas que dificulten su acceso a la información (en caso de ruidos externos, en situaciones donde nuestra atención visual y auditiva no esté disponible).

Para hacer el contenido Web accesible, se han desarrollado las denominadas Pautas de Accesibilidad al Contenido en la Web (WCAG), cuya función principal es guiar el diseño de páginas Web hacia un diseño accesible, reduciendo de esta forma barreras a la información. WCAG contiene 14 pautas que proporcionan soluciones de diseño y que utilizan como ejemplo situaciones comunes en las que el diseño de una página puede producir problemas de acceso a la información. Las pautas contienen además una serie de puntos de verificación que ayudan a detectar posibles errores.

Cada punto de verificación está asignado a uno de los tres niveles de prioridad establecidos por las pautas.[\[2\]](#)

- **Prioridad 1:** son aquellos puntos que un desarrollador Web tiene que cumplir ya que, de otra manera, ciertos grupos de usuarios no podrían acceder a la información del sitio Web.
- **Prioridad 2:** son aquellos puntos que un desarrollador Web debería cumplir ya que, si no fuese así, sería muy difícil acceder a la información para ciertos grupos de usuarios.
- **Prioridad 3:** son aquellos puntos que un desarrollador Web debería cumplir ya que, de otra forma, algunos usuarios experimentarían ciertas dificultades para acceder a la información.

Existen herramientas que evalúan el nivel de prioridad con el que se esta cumpliendo, ya que analizan que todos los elementos involucrados cumplan con el nivel de prioridad especificado.

Se genera un resumen como el que aparece abajo, y como se puede ver los elementos cumplen con las prioridades (Ver Fig. 412 Verificación de accesibilidad).

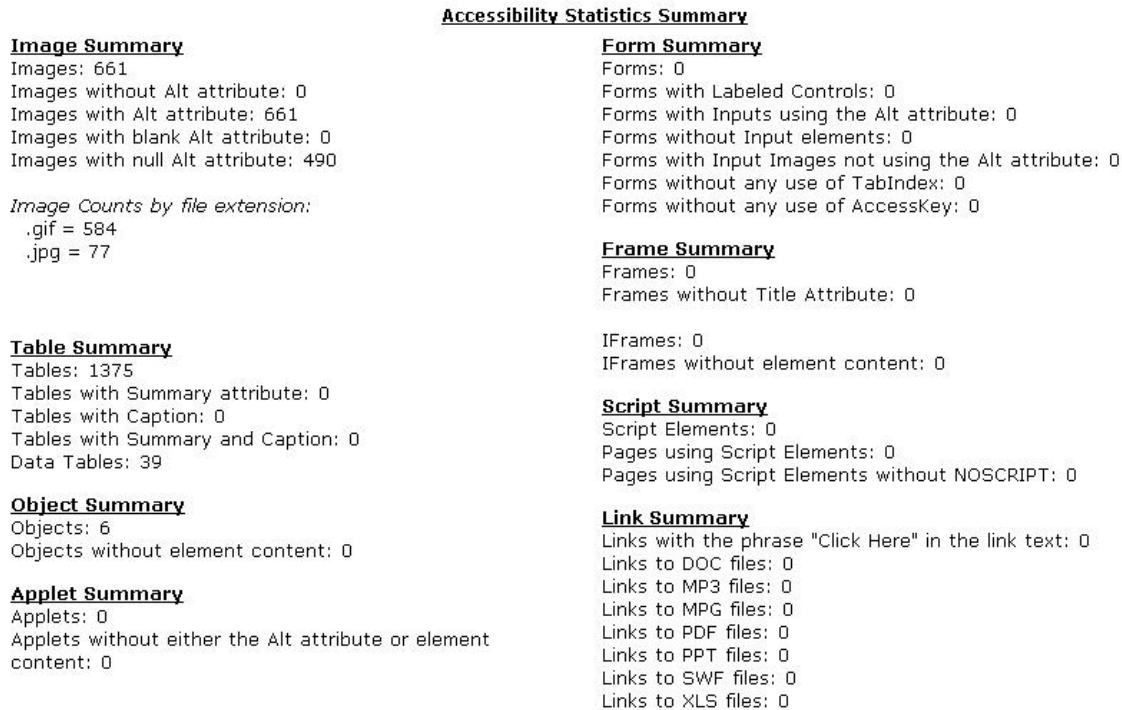


Fig. 4.12 - Verificación de accesibilidad

El HTML Transitional incluye mecanismos para hojas de estilo, scripts, objetos incluidos, soporte mejorado para textos de izquierda a derecha o direcciones mezcladas, y mejoras en los formularios para facilitar la accesibilidad por parte de personas discapacitadas. [3]

Existe una herramienta que utilizamos para poder analizar los elementos que cumplan con los lineamientos establecidos (Ver Fig. 4.13 - Herramienta de verificación de HTML Transitional 4.0). Esta herramienta (A Real Validator) contienen un motor interno que valida las reglas de estructura del DTD de la versión HTML 4.01 Transitional, marcando los posibles errores y sugerencias para su correcta implantación en el código fuente.

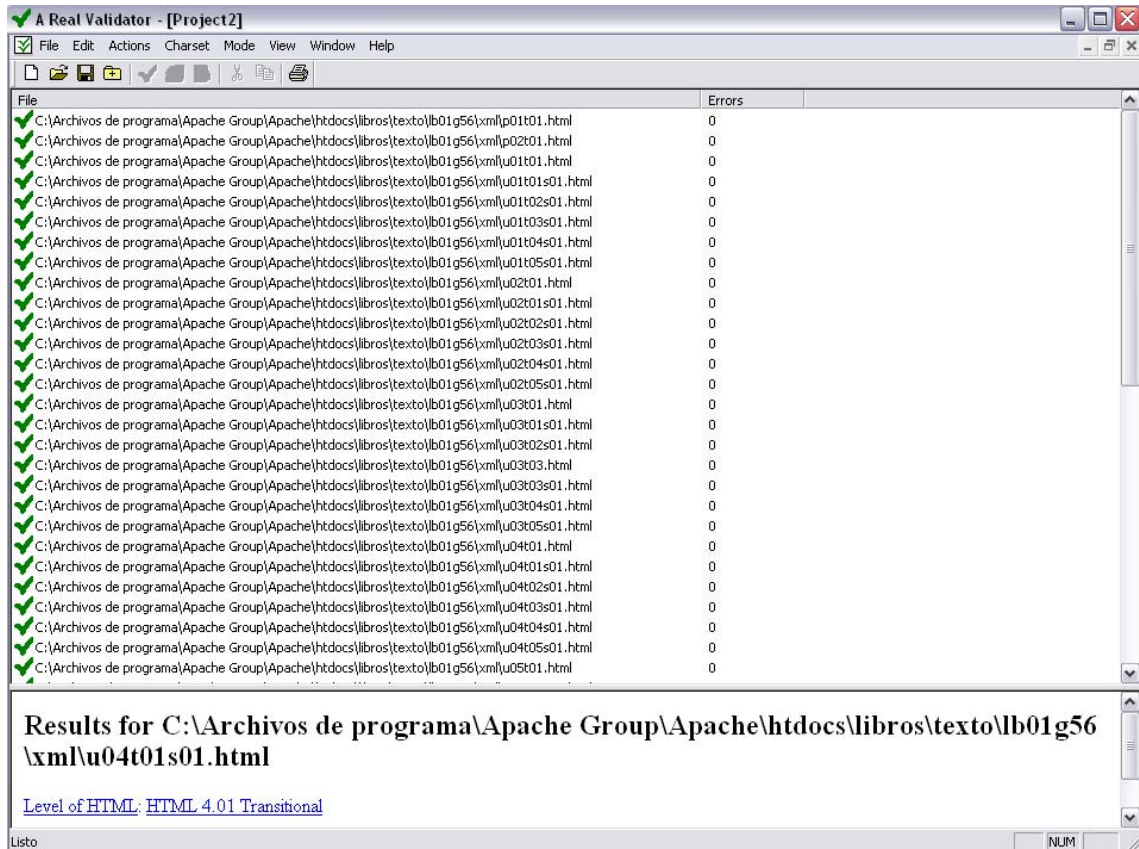


Fig. 4.13 Herramienta de verificación de HTML Transitional 4.01

BIBLIOGRAFÍA DEL CAPÍTULO

[1] <http://xml.apache.org/xalan-j/> visitado en Agosto de 2007

[2] <http://www.w3.org/TR/WCAG10-HTML-TECHS/> visitado en Agosto de 2007

[3] <http://www.w3.org/TR/html401/sgml/loosedtd.html> visitado en Agosto de 2007

CAPÍTULO 5

Publicación de edición digital

Hoy en día consultar un documento vía Internet es algo relativamente fácil. En la última década el acceso a diferentes documentos a través de la red ha ido evolucionando. Un claro ejemplo de ello es la publicación digital, en la que se usan la computación e Internet para crear y divulgar la información.

Por medio de una publicación digital es posible divulgar el conocimiento a través de Internet, estableciéndose una comunicación entre el autor y el usuario. Con el apoyo de multimedia se facilitan la transmisión de los contenidos y apoya la comprensión de la información.

El término publicación digital surge como un intento para diferenciar la información plasmada en papel, de toda aquella que se produce en la computadora. Los documentos digitales se publican en un determinado soporte y formato, que para ser consultados el usuario requiere de un procesador.

Se necesitan recursos humanos, tecnológicos y financieros, además de comprender los procesos, la transformación, la catalogación, la agrupación, la presentación y la distribución de información útil para los propósitos de la publicación digital, así como los procedimientos de autorización [1].

5.1 Implantación en caso de estudio Enciclomedia

El proceso de implantación de la edición digital creada por medio de la transformación dinámica de estructuras de marcado, se apoya en el modelado de procesos MoProSoft (Modelo de procesos para la industria de software). Se mencionarán algunas de las definiciones del modelado de procesos usadas en el proyecto Enciclomedia, proyecto implantado a nivel nacional, explicado posteriormente más a detalle.

5.1.1 Uso de MoProSoft

Un aspecto importante es mencionar por qué optar por el modelo de procesos *MoProSoft*, está relacionado a que en la actualidad el desarrollo de software sigue metodologías específicas o patrones de diseño como son RUP, AGILENT, XP, etc., y si se desea desarrollar un proyecto robusto que involucre a todos los procesos relacionados en el desarrollo del software es necesario seguir un modelo de evaluación de procesos.

Diferente a otros modelos de evaluación de procesos como son CMM y CMMI, SIX SIGMA, ISO, entre estos algunos son más complejos y además no están “tropicalizados” o

adaptados a México, aspectos explicados más adelante. Debido a la complejidad del modelo de proceso, es importante mencionar que exclusivamente mencionaremos la categoría de operación y específicamente “*Desarrollo y Mantenimiento de Software > Actividades y Diagrama de Flujo*”.

5.1.2 Antecedentes de MoProSoft

Para comenzar con la explicación del apartado que se utiliza para el desarrollo de la edición digital en MoProSoft es necesario proporcionar algunos antecedentes.

Podemos definir un proceso como un conjunto de prácticas relacionadas entre sí, llevadas a cabo a través de roles y por elementos automatizados, que mediante recursos y a partir de insumos, producen un satisfactor de negocio para el cliente. La madurez de un proceso es el nivel al cual está explícitamente documentado, gestionado, medido, controlado y continuamente mejorado.

Se asume que un proceso maduro tendrá alta capacidad; ésta última hace referencia al rango de resultados esperados que pueden obtenerse al realizarlo, lo que permite predecir el desempeño de futuras ejecuciones.

Un modelo de procesos es un conjunto estructurado de elementos que describen las características de procesos efectivos y de calidad, indicando “qué hacer”, no “cómo hacer” ni “quién lo hace”. Actualmente existen modelos internacionales como CMMI e ISO.

CMMI implica un amplio marco de trabajo donde se describen las prácticas esenciales de un proceso efectivo de desarrollo de software; fue creado en Estados Unidos y se ha convertido en una referencia internacional de la capacidad de los procesos de desarrollo de software. Define 25 áreas de proceso del ámbito de la ingeniería de software y la ingeniería de sistemas.

Por su parte, ISO 9000 es una familia de normas que sirve como guía a las organizaciones —de todo tipo y tamaño— en la implantación y operación de Sistemas de Gestión de la Calidad. ISO 9000 en su versión 2000 tiene un fuerte enfoque en los procesos y en la satisfacción del cliente; además de que es certificable en México. Sin embargo, por su generalidad, es abstracto y difícil de adecuar al desarrollo de software.

Si bien estos modelos internacionales permiten determinar la capacidad de procesos de una organización que desarrolla software, no es fácil adaptarlos al contexto de las empresas mexicanas, que en su mayoría son PyMEs (Pequeñas y medianas empresas). De acuerdo con un estudio hecho en 2004, 92% de las empresas que desarrollan software en México son PyMEs, con menos de 100 empleados.

Es así como surge MoProSoft, por iniciativa de la Secretaría de Economía y gracias al trabajo de académicos y empresarios mexicanos, encabezados por la Dra. Hanna Oktaba, profesora de la Facultad de Ciencias de la UNAM. MoProSoft es un modelo de procesos

para la industria de software nacional, que fomenta la estandarización de su operación a través de la incorporación de las mejores prácticas en gestión e ingeniería de software. La adopción del modelo permite elevar la capacidad de las organizaciones que desarrollan o mantienen software para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad. Es también aplicable en áreas internas de desarrollo de software de las empresas de diversos giros.

Características de MoProSoft

- Es específico para el desarrollo y mantenimiento de software.
- Es sencillo de entender y adoptar.
- Facilita el cumplimiento de los requisitos de otros modelos como ISO 9000:2000, CMM y CMMI.
- Se enfoca a procesos.
- Se le considera práctico en su aplicación, principalmente en organizaciones pequeñas, con bajos niveles de madurez.
- Comprende un documento de menos de 200 páginas que, al compararlo con otros modelos y estándares, lo hace bastante práctico.
- Resulta acorde con la estructura de las organizaciones mexicanas de la industria de software.
- Está orientado a mejorar los procesos, para contribuir a los objetivos de negocio, y no simplemente ser un marco de referencia o certificación.
- Tiene un bajo costo, tanto para su adopción como para su evaluación.

¿Para qué sirve MoProSoft?

- Mejora la calidad del software producido por la empresa que adopta el modelo.
- Eleva la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.
- Integra todos los procesos de la organización y mantiene la alineación con los objetivos estratégicos.
- Inicia el camino a la adopción de los modelos ISO 9000 o CMMI.
- Sirve para implantar un programa de mejora continua.
- Permite reconocer a las organizaciones mexicanas por su nivel de madurez de procesos.
- Facilita la selección de proveedores.
- Permite obtener acceso a las prácticas de ingeniería de software de clase mundial.

Estructura de MoProSoft

El modelo pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua. Sintetiza las mejores prácticas en un conjunto pequeño de procesos que abarcan las responsabilidades asociadas a la estructura de una organización que son: la Alta Dirección, Gestión y Operación.

MoProSoft es un modelo integrado donde las salidas de un proceso están claramente dirigidas como entradas a otros; las prácticas de planeación, seguimiento y evaluación se incluyeron en todos los procesos de gestión y administración; por su parte los objetivos, los indicadores, las mediciones y las metas cuantitativas fueron incorporados de manera congruente y práctica en todos los procesos; las verificaciones, validaciones y pruebas están incluidas de manera explícita dentro de las actividades de los procesos; y existe una base de conocimientos que resguarda todos los documentos y productos generados [2]:

5.1.3 Análisis específico de implantación

El proceso que se analiza está diseñado para pertenecer a la capa de operación. A continuación se muestran una sección de las definiciones de las actividades del proyecto Enciclomedia.

Planeación:

Se maneja la parte conceptual del proceso, se realizan actividades de recepción, análisis, descripción, etc, para poder iniciar con la ejecución del proceso.

A1. (O1)	1 Planeación de contenidos
	Selección del material impreso que se desarrollará como libros digitales
A2. (O1)	2 Recepción de materia impreso y/o digital
	Verificar mediante <i>Lista de verificación</i> la integridad de la información impresa y la operatividad de los archivos digitales, y valida si es una publicación que ya se haya trabajado.
A3. (O1)	3 Catalogación del libro
	Descripción del contenido del libro conforme a estándares de metadatos
A4. (O2)	4 Análisis del libro en papel
	<i>Análisis</i> estructural de los elementos gráficos y de contenido concernientes a la <i>Especificación de requerimientos</i> , en función del <i>Plan de desarrollo</i> . De éste análisis se derivan el <i>diseño</i> y las siguientes actividades paralelas o consecutivas (ver Fig – E1. <i>Diagrama de flujo de proceso</i>).
AMI (O3)	MI Análisis de nuevos requerimientos
	<i>Análisis</i> estructural de los elementos gráficos y de contenido concernientes a la actualización de la publicación digital.

Ejecución:

Se inicia la construcción del producto.

A5. (O2) 5 Transformación previa	
	<p>Generación progresiva de plantillas preliminares para visualizar el resultado de las actividades intermedias. Se realiza o se asigna la elaboración de los instrumentos base del trabajo que desarrolla cada departamento: Estructura prototipo en XML (Desarrollo Editorial) Documentación de elementos gráficos prototipo o <i>documentación visual</i> (Diseño Gráfico) Plantillas de transformación prototipo o previas (Desarrollo Tecnológico) <i>Componentes de presentación</i> de uso transitorio en la codificación de contenidos Incorporar a la base de conocimiento intermedia</p>
A6. (O2) 6 Codificado y procesamiento	
	<p>Se integra la <i>configuración de software</i> que determinará las actividades y los <i>reportes de actividades</i> subsecuentes Codificación de textos en lenguaje XML. Procesamiento de imágenes y desarrollo de animaciones e interactivos. Programación de elementos de transformación (VERIFICACION – VER1).</p>
A7. (O2) 7 Diseño Editorial	
	<p>Integración de los elementos anteriores para asegurar la consistencia en el soporte digital.</p>
AM2 (O3) M2 Mantenimiento de edición digital	
	<p><i>Mantenimiento</i> estructural de los elementos gráficos y de contenido concernientes a la actualización de la publicación digital.</p>

Monitoreo y control:

Consiste en asegurar que se cumplan los objetivos del proceso, se supervisa y evalúa el progreso para detectar desviaciones y realizar acciones correctivas.

A8. (O3) 8 Control de calidad	
	<p>Revisa los aspectos editoriales de la publicación digital en cumplimiento con la <i>Especificación de requerimientos</i>. Incorporar a la base de conocimiento final</p>
	<p>El programador realiza un monitoreo de los elementos que integra en la transformación (VERIFICACIÓN – VER2)</p>
AM3 (O3) M3 Control de calidad del mantenimiento	
	<p>Control de cambios en el mantenimiento de la publicación digital</p>

Cierre:

Su objetivo es entregar los productos de acuerdo al protocolo de entrega y dar por concluido el ciclo de entrega.

A9. (O3)	9 Revisión y validación tecnológica
	El PR verifica los aspectos de funcionalidad de la publicación digital para cumplir con la <i>Especificación de requerimientos</i> en el servidor de transformación preliminar. (VERIFICACIÓN – VER3), el RDT valida (VALIDACION – VALI), si la validación fue satisfactoria el programador transfiere la publicación digital al servidor de transformación final. El RDT valida (VERIFICACIÓN – VER4) la correcta integración al servidor de transformación final.
A10. (O3)	10 Entrega
	Define y establece la logística de entrega. Crea una copia de seguridad y hace la entrega en un medio digital (CD, HD, E-MAIL, etc.).
	Se incorporan a la Base de conocimiento final las modificaciones que se hayan incorporado al proceso
AM4 (O3)	M4 Entrega de publicación actualizada
	Cierre del mantenimiento de la publicación digital.

Nota: Para más detalles, el anexo [A2](#) presenta el proceso completo de la producción de ediciones digitales usando MoProSoft.

5.2 Verificación, validación de código y diagrama de flujo

La verificación del código se hace en varias fases del proceso, se aplican plantillas de verificación y validación, podemos definir estos dos conceptos de la siguiente forma:

Validación: La técnica para evaluar un componente o producto durante una fase o proyecto, o al finalizar los mismos, **a fin de garantizar que cumpla con los requisitos especificados.**

Verificación: La técnica de evaluar un componente o producto al final de una fase o proyecto **para asegurar o confirmar que cumple con las condiciones impuestas [3].**

Estos pasos tienen como objetivo garantizar las condiciones impuestas en el plan de trabajo y generar un producto que cumplan en forma. Qué tan estricto puede ser un sistema de verificación y validación dependerá del diseño del producto y los riesgos que implique el dejar pocos controles.

Se considera en este proceso no saturar los controles para no obstaculizar la producción, es necesario tener un análisis para saber exactamente en donde van a ser más eficientes.

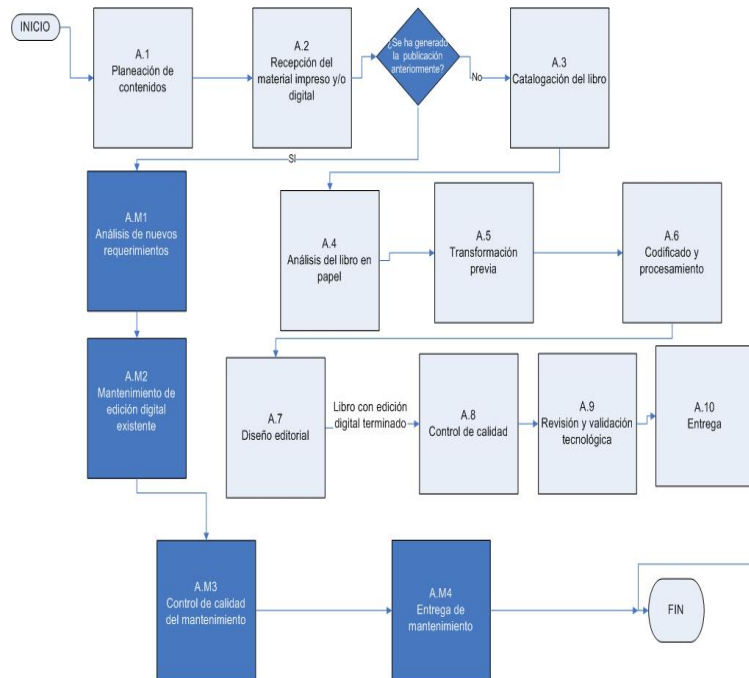


Fig - 5.1. Diagrama de flujo de proceso

5.3 Enciclomedia y la edición digital

El caso de estudio Enciclomedia es tomado por ser un proyecto producido en esta época, el proceso realmente está diseñado para crear ediciones digitales que sean adaptadas a cualquier ambiente. Es importante notar que todos los casos que se presentan están realizados para requerimientos muy semejantes al proyecto Enciclomedia que tiene derechos de autor sobre la interfaz y algunos recursos. Todos los códigos presentados anteriormente no tienen patente ni derechos de autor son codificaciones genéricas que pueden ser adaptadas a otros proyectos.

Enciclomedia constituye la edición digital de los Libros de Texto Gratuitos de la Secretaría de Educación Pública (SEP). Su característica principal es que ha vinculado las lecciones de los libros (con los que año con año trabajan niños y maestros en todo el país), con diversos recursos didácticos como imágenes fijas y en movimiento, interactivos, audio, videos, mapas, visitas virtuales, recursos de la enciclopedia Microsoft Encarta®, entre otros.

Enciclomedia ha recuperado e integrado la experiencia y el esfuerzo que por años ha realizado la SEP, pues muchos de los recursos pedagógicos que incluye el programa son resultado de proyectos eficazmente probados como Red Escolar, Sepiensa, Biblioteca Digital, Enseñanza de la Física con Tecnologías (EFIT), Enseñanza de las Matemáticas con Tecnologías (EMAT) y Biblioteca del Aula.

Enciclomedia ha integrado un amplio acervo educativo con novedosas rutas de acceso a la información, a fin de generar aprendizajes más significativos, congruentes con la realidad que viven las generaciones actuales, así como acercar el uso de las Tecnologías de la Información y la Comunicación (TIC) de manera equitativa y gratuita a las primarias y secundarias públicas del país.

En atención a las necesidades y actividades de los profesores se creó, además de una estrategia permanente de capacitación y actualización, el *Sitio del Maestro*: un espacio que cuenta entre otras cosas con sugerencias didácticas, la digitalización de sus herramientas de trabajo —avance programático, libro del maestro, ficheros— y un apartado de materiales para el desarrollo profesional; todo ello encaminado a que el maestro aproveche al máximo este recurso educativo y su labor pedagógica sea más útil.

Enciclomedia ha generado un espacio de colaboración social, al enriquecer permanentemente sus contenidos gracias a la participación de individuos, organizaciones públicas y particulares, en los ámbitos estatal y federal [4].

Los requerimientos de las ediciones digitales para poder ser integrados en el sistema son muy simples, a continuación se muestra una lista de los más significativos.

1. Código HTML, validado en estándar 4.01 Transitional
2. Configuración previamente acordada para la distribución de carpetas (misma que ya se explicó en la configuración del servidor)

Los lineamientos editoriales son determinados donde son creadas las ediciones digitales, la complejidad de las cuestiones editoriales como son: definición de extensión de archivo, agrupamiento de elementos por contexto y en general cosas generales del cuidado de la edición hacen que sea mucho más eficiente decidir esto con la plantilla creadora de la edición digital.

El motor de *Enciclomedia* tiene como base la edición digital de los Libros de Texto Gratuitos, si no existe un producto base que tenga relaciones hacia los recursos que enriquecen a la herramienta que esta pensada para el profesor, no tendría sentido hacer un sistema complejo con miles de recursos cuando no existe forma de tenerlos disponibles en el momento de la enseñanza de la base.

Gracias al tipo de contenido generado en la edición digital (HTML 4.01 Transitional), es posible que los motores que son usados para ligar la información funcionen correctamente para la parte académica encargada de este proceso.

El proceso de la Edición digital, XML tiene por objetivo preservar una capa de datos que puede regenerar en cuestión de minutos cambios generales en toda la edición sin necesidad de modificar cada uno de los elementos (equivalentes a una página de la versión impresa), con esto la actualización de las ediciones digitales está reducida a cambios generales en la mayoría de los casos optimizando el trabajo de una manera que sería

complicado hacerlo de otra forma.

Las Ediciones Digitales en el proyecto Enciclomedia son fundamentales, además de estar el desarrollo pensado en obtener un esquema robusto que pueda ser clave en el mantenimiento del mismo proyecto.

5.4 Estructura de elementos a publicar

Un sitio web no suele ser necesariamente equivalente a un documento, el problema está en que se han confundido las tecnologías de acceso a la información (infraestructura de acceso) con las tecnologías de publicación (infraestructura de publicación); en concreto se confundido el uso de HTML para ambas infraestructuras.

5.4.1 Análisis del producto generado para su publicación

Suponiendo que realmente no se pueda prescindir de HTML (e incluso de XML) como infraestructura de acceso a la información, es necesario separar funcionalmente el documento de la infraestructura de acceso. La forma recomendable de publicar un producto es la siguiente:

- Documento transparente
- Código fuente estructurado
- Formatos de publicación:
 - versiones navegables (html)
 - versiones imprimibles (ps, pdf)
 - otras versiones (rtf, word)
 - versiones de los anteriores empaquetados (tar/gz, zip)
- Incorporación de meta-información (rdf, OMF, Dublin Core)
- Publicación a través de bibliotecas digitales

Con estas condiciones podemos establecer que la edición digital basada en las estructuras de marcado ya explicadas con detalle son lo suficientemente robustas para cubrir la totalidad de las condiciones.

Comúnmente en los proyectos se verifica una línea de desarrollo que puede estar soportada en sistemas de bases de datos, administradores dinámicos, etc. cosas que en general funcionan para el proyecto, cuando se intenta dar un giro para la salida del formato por ejemplo cuando los constructores del proyecto se dan cuenta que tan limitada tienen la parte “dinámica”.

Si se analiza que el proyecto parte de separar en un inicio los conceptos de los grupos de trabajo, que es semejante a tener por ejemplo un modelo MVC (Model View Controller), muy usado en esta época, nos podemos dar cuenta de la integridad de cada esfuerzo de

trabajo perfectamente definido puede evitar re-ingeniería o code refactoring, manejo de tiempos, etc.

Analizadas estas posibilidades para generar un producto adecuado para su publicación, podemos definir que el producto dependerá de las necesidades de la plataforma sin los costos ocultos, esto gracias a la estrategia tomada.

El producto que se genera esta diseñado para un sitio de publicación elemental por cuestiones de simplicidad, el producto se define como Edición digital en formato HTML. Los recursos que la Edición digital tiene son dependientes a las relaciones de las rutas programadas en los documentos de formato HTML. Las características de la Edición digital o producto generado son las siguientes:

- Páginas codificadas en lenguaje HTML 4.01 Transitional
- Imágenes en formatos de fácil soporte para los navegadores tradicionales (jpg, gif, png)
- Animaciones en formato SWF, para soporte con los visualizadores de la época.
- Hojas de estilo en cascada (CSS), apegadas al estándar CSS 1.0
- Ocasionalmente uso de código javascript 1.2

Solución de salida simple soportada por una estructura robusta que permite modificar desde la integración interdisciplinaria hasta la parte específica del producto.

5.4.2 Requisitos del producto generado para la publicación

Dependientes de los requisitos de publicación se encuentran los del producto, si estos no son visualizados de forma correcta se puede caer en una inconsistencia que generará muchos de los problemas de la ingeniería de software.

Para una publicación tradicional que soporte nuestro producto generado tenemos las siguientes condiciones o características:

- Soporte para publicación de contenidos HTML 4.01 Transitional
- Reproducción de formatos elementales gráficos soportados por HTML 4.01 Transitional.
- Soporte de contenidos multimedia para la tecnología Flash Player
- Ejecución de código javascript 1.2

Notablemente parecen importar más las condiciones del cliente que de la misma

infraestructura de publicación lo que descarta una compleja configuración del lado del servidor y dejando las complejidades en estándares de navegadores para la publicación.

5.5 Publicación digital

Descartadas la mayoría de las complejidades de la publicación para el caso en uso, debemos considerar otros factores de publicación digital como son:

- Derechos de autor
- Tiempos de acceso al producto
- Tipo de acceso a los contenidos
- Distribución de los contenidos

El entorno de la Edición digital juega un proceso crucial ya que puede ser un entorno aislado o de acceso abierto. Las ventas de la obra pueden ser muy restrictivas o puede ser una obra clásica de difusión libre y abierta. Podemos mencionar dos aspectos de tipo de publicación, el primero será para el proyecto Enciclomedia y el segundo para una Biblioteca Digital.

Para el caso del proyecto Enciclomedia la Edición digital debe ser adaptada a las condiciones de la interfaz y herramientas que generan los vínculos dinámicos hacia la Edición digital.

El proyecto Enciclomedia tecnológicamente está pensado como una aplicación web, usando versiones en HTML y Aplicaciones Flash para la reproducción de sus contenidos, en la parte dinámica se apoya de un manejador de bases de datos y scripts del lado del servidor para la comunicación y manipulación de los contenidos, por ejemplo las referencias o vínculos hacia la edición digital, herramientas de la interfaz, administradores, etc. El factor de aplicación web requiere presentar una edición digital que cumpla con la integración en un servidor web, condiciones explicadas anteriormente y que son equivalentes para este proyecto.

Cuestiones como los derechos de autor autorizados para el proyecto son gestionados y no existe restricción para su publicación dentro del sistema de Enciclomedia. El tiempo y tipo de acceso del producto no se convierte en un factor a considerar debido a que es una aplicación desarrollada para ser independiente de un sistema de comunicación como Internet, no limitada para ello y no pensada para ese propósito. La distribución de los contenidos es por unidad de instalación lo que permite tener un control relacionado al número de instalaciones de Enciclomedia.

Finalmente se tiene una publicación web dentro de una aplicación web local no distribuida, abriendo y cerrando posibilidades.

Terminado el producto con los requisitos de publicación, la plataforma esta relacionada con el tipo de instalación, el proyecto Enciclomedia tiene su plataforma orientada a una instalación amigable y con un sistema operativo común.

Si la Edición digital es generada para otro tipo de proyecto, como una Biblioteca Digital, se tiene que controlar el tipo de acceso, los derechos de autor, y algún requisito en particular que requiera el servidor de publicación. Todo requisito será previamente analizado al inicio del proceso de creación de las estructuras de marcado y su programación.

BIBLIOGRAFÍA DEL CAPÍTULO

[1] <http://www.mati.unam.mx/> - Visitada en Octubre de 2007

[2] <http://www.enterate.unam.mx/Articulos/2006/marzo/moprosoft.htm> - Visitada en Septiembre de 2007

[3] PMI (2004). Guía de los fundamentos de la Dirección de Proyectos: Global STANDAR - PMI.

[4] <http://www.enciclopedia.edu.mx> - Visitada en Septiembre de 2007

Conclusiones

Una integración de elementos adecuada genera una producción controlada y eficiente. A lo largo de este trabajo se presentan los factores que favorecen dicha integración en un esquema de producción editorial en soporte digital. El resultado de esta investigación es una guía para la conformación de un equipo interdisciplinario en un proyecto de publicaciones digitales, que cumplan los requerimientos más estrictos.

En la vida profesional se tiene que convivir con personas de diferentes disciplinas, ya sean profesionales o no. Lo que es poco común de mostrar son métodos o estrategias, para que éstas disciplinas funcionen adecuadamente para un proceso. Los productos editoriales como los libros, se han trabajado en entidades editoriales normalmente integrando colaboradores como impresores, diseñadores, etc. Regularmente formando un producto dependiente de la imprenta para su elaboración masiva y con la complejidad de generar sólo un formato debido al costo de producción.

La publicación electrónica ha tomado fuerza gracias a la integración de las computadoras a gran parte de las tareas académicas y comerciales, lo que presentó un aumento extraordinario en el número de usuarios de Internet. Básicamente, existen, de acuerdo a sus soportes, dos tipos de publicaciones digitales. La publicaciones fijas, que son montadas en un soporte fijo transportable, y las publicaciones en línea, que son colocadas en un servidor para su consulta por Internet. Ambos tipos de publicaciones utilizan estructuras en red para organizar la información y pueden hacer uso de la multimedia, aunque en este aspecto las publicaciones en línea deben optimizar los archivos, para que así su transmisión no sea muy lenta. Es importante señalar que es muy sutil la frontera entre ambos tipos de publicaciones, dado que una publicación puede combinar aplicaciones de ambos soportes para su distribución, actualización o consulta.

Debemos entender que hacer uso de la tecnología nos facilitará el camino para llegar al objetivo esperado. La combinación de las nuevas tecnologías con el proceso editorial da como resultado la posibilidad de disfrutar de grandes avances.

La producción de una publicación digital puede ser muy sencilla. Basta con el conocimiento de un lenguaje de marcas y una manipulación de imágenes. No obstante, para explotar mejor el potencial de las herramientas que actualmente nos proporciona la tecnología informática, para este tipo de publicaciones, es necesario constituir equipos multidisciplinarios con conocimientos altamente técnicos y especializados. Lo más recomendable, para la edición de una publicación de calidad, es dividir el proceso de producción en tres áreas de trabajo: un área de producción, encargada de generar y corregir los distintos archivos que utilizará la publicación; un área de diseño, encargada de diseñar la interfaz a través de la cual el usuario interactúa con la información, y un área de programación, encargada de generar las aplicaciones de automatización, la seguridad, las

bases de datos y, en general la comunicación entre las distintas partes de la obra. Deberá haber un encargado que coordine y supervise la integración del trabajo de cada área.

En el proceso para la *transformación de estructuras de marcado para la generación de ediciones digitales*, se presentó una estrategia que permite separar tres grupos de diferente alcance profesional, desarrollo tecnológico, desarrollo editorial y diseño gráfico, capaces de generar un producto digital con las ventajas de tener un soporte tecnológico escalable e incluyente, el cual es punto de referencia para poder desarrollar la edición digital.

Dos aspectos importantes en las ediciones digitales son, la preservación y la mejora de las condiciones de accesibilidad. La preservación es un problema que se ha resuelto no importa el número de consultas que se hagan, el aspecto siempre será el mismo, no habrá modificación alguna. En cuanto a la accesibilidad el paso que se da es enorme, un documento digital puede estar disponible para varios usuarios simultáneos independientemente de sus capacidades técnicas o físicas, consiguiendo que cualquier persona realice la misma acción, entre más grande sea el grado de accesibilidad mayor será el número de personas que puedan tener acceso a la información.

La exclusiva *transformación dinámica de estructuras de marcado para generar ediciones digitales*, es un proceso totalmente tecnológico explicado detalladamente en el presente trabajo. No podemos dejar de mencionar que a diferencia de hacer una transformación dinámica de productos que no requieran un cuidado editorial, el proceso para generar ediciones digitales si necesitará tomar en cuenta con detalle los elementos editoriales y gráficos para generar una edición digital de buena índole.

Superados los problemas tecnológicos del proceso de digitalización, surge la problemática relacionada con el contenido de la obra en sí, podría existir la duda, de que si en verdad lo que se esta leyendo es en realidad copia fiel de la obra original, por esto el proceso de control de calidad debe de ser altamente riguroso, para respetar por mínimo que parezca cualquier detalle que contenga la obra original, por ejemplo respetar la tipografía, respetar la apariencia de los elementos gráficos, que los errores ortográficos sean nulos, etc. En la medida que se logre este objetivo las ediciones digitales tendrán un grado de aceptación mucho mayor.

Es importante resaltar que gracias a la tecnología se ha logrado que las ediciones digitales logren transmitir las mismas sensaciones al lector que los materiales originales, esto es un gran paso porque podemos decir que no existe actualmente diferencia en cuanto a la presentación del contenido.

En ocasiones uno de los obstáculos que deben enfrentar las ediciones digitales es que se tiene la idea de que se descuida la parte del contenido, poniendo énfasis sólo en el proceso tecnológico, lo cierto es que cada una de las áreas involucradas en el proceso de digitalización son relevantes, si podemos ver a dicho proceso como un rompecabezas y cada una de las áreas involucradas como parte de éste, sabremos entonces que todo es importante.

La parte que analiza con detalle este trabajo, no puede ser explicada sin considerar la interacción en su conjunto. Podemos concluir que un adecuado uso de la tecnología no debe limitar en sus procesos de creación a las disciplinas sólo por no integrarlas adecuadamente, esto es evidente cuando se desarrolla una interfaz con juicios preestablecidos, sin tener elementos suficientemente configurables que permitan escalar o corregir el producto por sí mismo.

ANEXOS

ANEXO – AI

```
1  /* Hoja de estilo:   Nombre edición digital */
2  /* Contenedor */
3  div.principal {
4      font-family:     Verdana;
5      width:           650px;
6      text-align:      left;
7      margin-left:     auto;
8      margin-right:    auto;
9  }
10
11 body {
12     text-align:       center;
13     background-image: url("../imgs/etiquetas/fondo_1.jpg");
14 }
15
16
17 /* Encabezados */
18 h1 {
19     font-family:      "Comic sans MS";
20     font-size:        44pt;
21     font-weight:      bolder;
22     color:             #FFFFFF;
23     text-align:       center;
24     margin-bottom:    20px;
25     margin-left:      30px;
26 }
27
28 h2 {
29     font-size:        16pt;
30     font-style:       italic;
31     color:             #FFFFFF;
32     text-align:       left;
33     margin-left:      40px;
34 }
35
36
```



```
37 /* Párrafos */
38 p {
39     font-size:        11pt;
40     font-weight:      normal;
41     text-indent:      0px;
42     margin-left:      20px;
43     margin-right:     5px;
44 }
45
46 p.centrado {
47     text-align:       center;
48 }
49
50 p.derecha {
51     text-align:       right;
52     padding-right:    10px;
53 }
54
55
56 p.animacion {
57     text-align:       center;
58     margin-left:      auto;
59     margin-right:     auto;
60     margin-top:       10px;
61     margin-bottom:    10px;
62 }
63
64 p.titulo {
65     font-size:        12pt;
66     font-weight:      bold;
67     text-indent:      0px;
68     text-align:       center;
69     margin-left:      0px;
70 }
71
72
73 /* Estilos */
74 span.negritas {
75     font-weight:      bold;
76 }
77
78 span.centrado {
79     text-align:       center;
80 }
81
82
```

```
83 /* Tablas atributo rend */
84 table.transparente {
85     font-size:      11pt;
86     text-align:     center;
87     margin-top:     15px;
88     margin-left:    20px;
89     margin-right:   7px;
90     width:          565px;
91 }
92
93 table.fondo {
94     text-align:     center;
95     width:          590px;
96     height:         137px;
97     margin:         auto;
98 }
99
100 table.verso {
101     margin-bottom:  10px;
102 }
103
104 table.derecha {
105     margin-right:   5px;
106     float:          right;
107 }
108
109 table.actividades {
110     background-color: #0070DF;
111     border:           2px #FFCC33 solid;
112     margin-left:     20px;
113     margin-top:      40px;
114     width:           300px;
115 }
116
117
```

```
118
119 /* Tablas de imágenes */
120 table.completa {
121     margin:          auto;
122 }
123
124 table.margen {
125     margin-top:      10px;
126     margin-right:    5px;
127     float:           left;
128 }
129
130 table.incrustada {
131     margin-left:     5px;
132     margin-top:      5px;
133     float:           right;
134 }
135
136 table.figAnimacion {
137     text-align:      center;
138     padding:         0px;
139     margin-left:     auto;
140     margin-right:    auto;
141 }
142
143
144
145 /* Tablas de navegación */
146 table.principal {
147     background-image: url('../imgs/etiquetas/fondo_1.jpg');
148     width:            650px;
149 }
150
151 table.navegaTemaSup {
152     float:           right;
153     height:          20px;
154 }
155
156 table.navegaSup {
157     width:           625px;
158     margin-top:      0px;
159     margin-bottom:   40px;
160 }
161
162 table.inicioBloque {
163     width:           570px;
164     height:          590px;
165     margin:          auto;
166 }
167
168
169
170
```

```
171 /* td */
172 td.centrado {
173     text-align:    center;
174 }
175
176 td.centro {
177     text-align:    center;
178 }
179
180
181
182
183 /* Listas */
184 ol.numeral {
185     font-size:    11pt;
186 }
187
188 li.simple {
189     font-size:    11pt;
190     list-style-type: none;
191     padding-bottom: 3px;
192     margin-left:  -20px;
193     margin-right: 5px;
194 }
195
196 ul.balazo {
197     font-size:    11pt;
198     list-style-type: disc;
199     margin-left:  0px;
200 }
```

```
201
202
203
204
205
206
207 /* Imágenes */
208 img.borde {
209     border:           0px;
210 }
211
212 img.centro {
213     border:           0px;
214     margin:           auto;
215 }
216
217 img.izquierda {
218     border:           0px;
219     margin-top:       10px;
220     margin-right:     5px;
221     float:            left;
222 }
223
224 img.derecha {
225     border:           0px;
226     margin-left:      5px;
227     margin-top:       5px;
228     float:            right;
229 }
230
231
232
233 /* divs */
234 div.contenedor {
235     font-size:        11pt;
236     font-weight:      normal;
237     text-indent:      0px;
238     margin-left:      20px;
239     margin-right:     5px;
240 }
241
242 div.img {
243     margin-left:      0px;
244     margin-right:     0px;
245     margin-bottom:    0px;
246     margin-top:       0px;
247 }
248
249
250
```

ANEXO – A2

Proceso	Producción de Ediciones Digitales
Categoría	OPE
Propósito	La producción de libros digitales que correspondan a las necesidades del sistema.
Descripción	<p>El propósito de la Producción de ediciones digitales para el sistema se compone de uno o más ciclos. Cada ciclo está compuesto de las siguientes fases:</p> <ul style="list-style-type: none"> • Inicio: Revisión del <i>Plan de Desarrollo</i> por los miembros del <i>Equipo de Trabajo</i> para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización. • Requerimientos: Conjunto de actividades cuya finalidad es obtener la documentación de la <i>Especificación de Requerimientos</i> y <i>Plan de desarrollo</i>. Se analiza los recursos disponibles en infraestructura y humanos. • Análisis y Diseño: Conjunto de actividades en las cuales se analizan los requerimientos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado se obtiene la documentación del <i>Análisis y Diseño</i> y <i>Plan de Pruebas de Integración</i>. • Construcción: Conjunto de actividades para producir <i>Componente(s)</i> de software que correspondan al <i>Análisis y Diseño</i>, así como la realización de pruebas unitarias. Como resultado se obtienen el (los) <i>Componente(s)</i> de software probados. <ul style="list-style-type: none"> • Se genera la versión preliminar • Cierre: Integración final de la <i>Configuración de Software</i> generada en las fases para su entrega. <p>Para generar los productos de cada una de estas fases se realizan las siguientes actividades:</p> <ul style="list-style-type: none"> • Distribución de tareas, se asignan las responsabilidades de cada miembro del <i>Equipo de Trabajo</i> de acuerdo al <i>Plan de Desarrollo</i>. • Producción, verificación, validación o prueba de los productos, así como su corrección correspondiente. • Generación del <i>Reporte de Actividades</i>, cada departamento genera uno diferente, el diseño gráfico y desarrollo tecnológico lo realizan en el sistema de control de acceso y editorial mediante un formato.
Objetivos	<p>O1 Lograr que los productos de salida sean adecuados para la tecnología desarrollada en el sistema.</p> <p>O2 Generar ediciones digitales consistentes con la edición impresa, que favorezcan la operatividad y el uso del sistema.</p> <p>O3 Mantener la actualidad tecnológica del modelo y adecuarlo a los nuevos requerimientos</p>
Indicadores	<p>I1 (O1) Apego a estándares de accesibilidad. Apego al estándar de estructura de código.</p> <p>I2 (O1) Estructura de archivos para el sistema.</p> <p>I3 (O2) Integridad de contenido y formación con respecto al original.</p> <p>I4 (O3) Actualización del proceso y los estándares que conforman la base del conocimiento.</p>
Metas cuantitativas	M1 (I1) Verificación (en un rango del noventa al cien por ciento) del nivel dos de las pautas de

	<p>accesibilidad recomendadas por la WAI y de la sintaxis del código HTML 4.01 transicional en el total de las páginas.</p> <p>M2 (I2) Verificación de que la estructura de archivos cumpla al cien por ciento con los requerimientos del sistema.</p> <p>M3 (I3) Verificación de la legibilidad y del cotejo con la edición impresa en el total de las páginas.</p> <p>M4 (I4) Mantener actualizada la configuración de la base de conocimientos del modelo de producción, con respecto a los requerimientos que implica el desarrollo del sistema del sistema.</p>
Responsabilidad	<ul style="list-style-type: none"> ▪ Inicio: RPEDL. • Requerimientos: RPEDL. • Análisis y Diseño: DI, CR, PR. • Construcción: BT • Se genera la versión preliminar PR • Cierre: PR.
Autoridad	<ul style="list-style-type: none"> • Dirección de Acervos Digitales: DAD
Procesos relacionados	Dirección Académica, Dirección Tecnológica y Dirección de Introspección e Innovación.

Entradas

Nombre	Fuente
<i>Edición impresa y archivos digitales del conjunto de libros que deberá digitalizarse.</i>	Subsecretaría de Educación Básica
<i>Fecha de entrega acordada</i>	Subsecretaría de Educación Básica
Requerimientos técnicos	DII, DT

Salidas

Nombre	Descripción	Destino
<i>Reporte(s) de Validación</i>	Documento que comprueba la validez de los productos de acuerdo a los estándares determinados.	DT, DII
Estructuras de codificación en xml	Criterios de uso de la tecnología XML derivados del análisis del libro impreso o digital.	DT, DII
CSS y SCRIPT	Elementos para que los contenidos codificados se transformen al formato de presentación convenido.	DT, DII
Libros digitales	Conjunto de archivos que integran el contenido en HTML, imágenes e interactivos.	DT, DII
		Desarrollo editorial de libro digital
<i>Componente</i>	Conjunto de unidades de código relacionadas.	Desarrollo editorial de libro digital
<i>Software</i>	Sistema de software, destinado a un cliente o usuario, constituido por componentes agrupados en subsistemas, posiblemente anidados.	Desarrollo editorial de libro digital
<i>Configuración de</i>	Conjunto consistente de productos de software, que incluye:	Desarrollo editorial de libro

Nombre	Descripción	Destino
Software	<ul style="list-style-type: none"> • <i>Especificación de Requerimientos</i> • <i>Análisis y Diseño</i> • <i>Software</i> (elementos del entregable) 	digital
Reporte de Actividades	<p>Registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como:</p> <ul style="list-style-type: none"> • tiempo de producción, de corrección, de verificación y de validación, • defectos encontrados en verificación, validación o prueba, • tamaño de productos. 	Desarrollo editorial de libro digital

Productos internos

Nombre	Descripción
Plan de Desarrollo	<ul style="list-style-type: none"> • <i>Descripción del Producto</i> Edición digital normalizada de libros de texto con el contenido íntegro de las ediciones impresas, con un diseño editorial que facilite su uso. • <i>Entregables</i> Conjunto de archivos HTML, asociados a una hoja de estilo. • <i>Proceso Específico</i> <ul style="list-style-type: none"> ○ Codificación de contenidos ○ Procesamiento de imágenes ○ Procesos de programación ○ Formación de páginas ○ Control de calidad • <i>Equipo de Trabajo</i> Asignación de personal de los departamentos de Desarrollo Editorial, Diseño Gráfico y Desarrollo Tecnológico, en función de la descripción del producto. • <i>Calendario</i> Fechas de entrega preestablecidas

Nombre	Descripción
<p><i>Especificación de Requerimientos</i></p>	<p>Introducción: Descripción general del software y su uso en el ámbito de negocio del cliente.</p> <p>Descripción de requerimientos:</p> <p>* Funcionales: Necesidades establecidas que debe satisfacer el software cuando es usado en condiciones específicas. Las funcionalidades deben ser adecuadas, exactas y seguras.</p> <p>* Interfaz con usuario: Definición de aquellas características de la interfaz de usuario que permiten que el software sea fácil de entender, aprender, que genere satisfacción y con el cual el usuario pueda desempeñar su tarea eficientemente. Incluyendo la descripción del prototipo de la interfaz.* Interfaces externas: Definición de las interfaces con otro software o con hardware.</p> <p>* Confiabilidad: Especificación del nivel de desempeño del software con respecto a la madurez, tolerancia a fallas y recuperación.</p> <p>* Eficiencia: Especificación del nivel de desempeño del software con respecto al tiempo y a la utilización de recursos.</p> <p>* Mantenimiento: Descripción de los elementos que facilitarán la comprensión y la realización de las modificaciones futuras del software.</p> <p>* Portabilidad: Descripción de las características del software que permitan su transferencia de un ambiente a otro.</p> <p>* Restricciones de diseño y construcción: Necesidades impuestas por el cliente.</p> <p>* Legales y reglamentarios: Necesidades impuestas por leyes, reglamentos, entre otros.</p>

Nombre	Descripción
Análisis y Diseño	<p>Este documento contiene la descripción textual y grafica de la estructura de los componentes de software. El cual consta de las siguientes partes:</p> <p>Arquitectónica: Contiene la estructura interna del sistema, es decir la descomposición del sistema en subsistemas. Así como a identificación de los componentes que integran los subsistemas y las relaciones de interacción entre ellos.</p> <p>Detallada: Contiene el detalle de los componentes que permita de manera evidente su construcción y prueba en el ambiente de programación.</p>
Documentación visual	Documentos que son generados mediante el análisis del libro gráficamente
Transformaciones previas	Documentos previos de integración para pruebas preliminares
Componentes de presentación	Conjunto de unidades de código relacionadas.
Configuración de Software	<p>Conjunto consistente de productos de software, que incluye:</p> <ul style="list-style-type: none"> • Especificación de Requerimientos • Análisis y Diseño • Software (elementos del entregable)
Reporte de Actividades	<p>Registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como:</p> <ul style="list-style-type: none"> • tiempo de producción, de corrección, de verificación y de validación, • defectos encontrados en verificación, validación o prueba, • tamaño de productos.

Prácticas

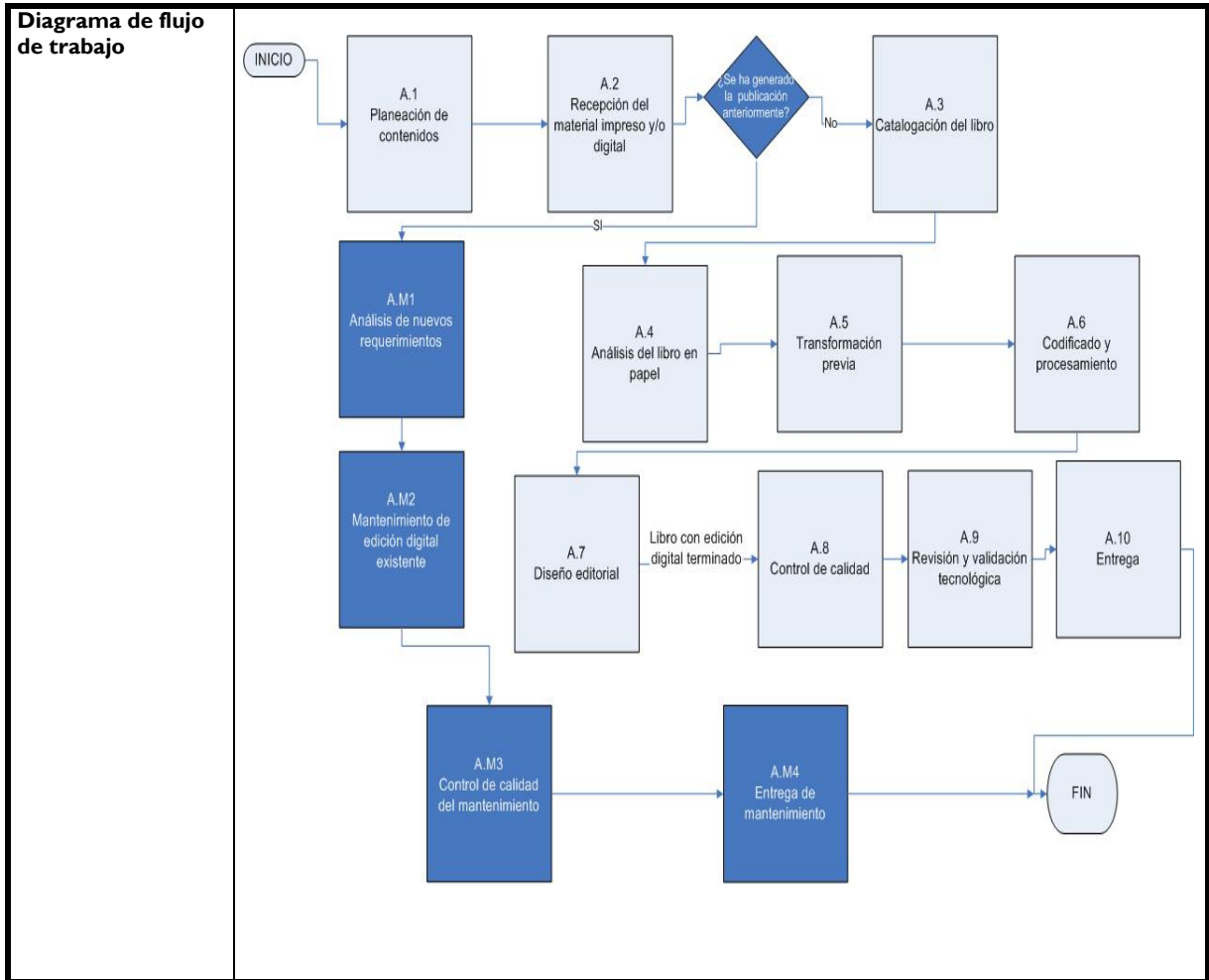
Roles involucrados y capacitación	Rol	Abreviatura	Capacitación
	Dirección de Acervos Digitales	DAD	
	Responsable de la Planeación de Edición Digital de Libros	RPEDL	Capacidad de liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal y desarrollo de software.

Responsable de Desarrollo de la Edición Digital de Libros	RDEDL	Conocimiento y experiencia en el desarrollo y mantenimiento de software.
Responsable de la recepción del material impreso y/o digital.	RRM	Conocimiento editorial.
Responsable de Desarrollo Editorial	RDE	Conocimiento editorial, de la lengua, manejo de herramientas para marcado y manejo de grupos de trabajo
Responsable de Diseño Gráfico	RDG	Conocimiento en diseño gráfico, conocimiento de herramientas para procesamiento de elementos visuales y manejo de grupos de trabajo
Responsable de Desarrollo Tecnológico	RDT	Conocimiento en desarrollo de software y procesamiento de información, herramientas de programación y manejo de grupos de trabajo
Responsable de Proceso	RP	Conocimiento y experiencia suficientes para el control del proceso involucrado, puede ser Diseño gráfico, programación, editorial.
Diseñador	DI	Conocimiento y experiencia en el diseño de la estructura de los componentes de software.
Programador	PR	Conocimiento y/o experiencia en la programación, integración y pruebas unitarias.
Corrector	CR	Conocimiento en las técnicas de revisión y experiencia en el desarrollo y mantenimiento de software.
Bibliotecólogo	BT	Conocimiento en bibliotecología.

Actividades

Rol	Descripción
PLANEACIÓN	
A1. (O1)	1 Planeación de contenidos
DAD	Selección del material impreso que se desarrollará como libros digitales.
A2. (O1)	2 Recepción de materia impreso y/o digital
RPEDL	Verificar mediante <i>Lista de verificación</i> la integridad de la información impresa y la operatividad de los archivos digitales, y valida si es una publicación que ya se haya trabajado.
A3. (O1)	3 Catalogación del libro
BT	Descripción del contenido del libro conforme a estándares de metadatos
A4. (O2)	4 Análisis del libro en papel
RDE, RDG,	<i>Análisis</i> estructural de los elementos gráficos y de contenido concernientes a la <i>Especificación de requerimientos</i> , en función del <i>Plan de desarrollo</i> . De éste <i>análisis</i> se derivan el <i>diseño</i> y las siguientes actividades paralelas o consecutivas

RDT.	(ver diagrama). (DOCUMENTACIÓN DE DOS DEPARTAMENTOS)
AMI (O3) M1 Análisis de nuevos requerimientos	
RDE, RDG, RDT.	Análisis estructural de los elementos gráficos y de contenido concernientes a la actualización de la publicación digital.
EJECUCIÓN	
A5. (O2) 5 Transformación previa	
RDE, RDG, RDT.	Generación progresiva de plantillas preliminares para visualizar el resultado de las actividades intermedias. Se realiza o se asigna la elaboración de los instrumentos base del trabajo que desarrolla cada departamento: Estructura prototipo en XML (Desarrollo Editorial) Documentación de elementos gráficos prototipo o <i>documentación visual</i> (Diseño Gráfico) Plantillas de transformación prototipo o previas (Desarrollo Tecnológico) <i>Componentes de presentación</i> de uso transitorio en la codificación de contenidos
	Incorporar a la base de conocimiento intermedia
A6. (O2) 6 Codificado y procesamiento	
DI, PR, CR	Se integra la <i>configuración de software</i> que determinará las actividades y los <i>reportes de actividades</i> subsecuentes Codificación de textos en lenguaje XML. (VERIFICACIÓN - VER1) Procesamiento de imágenes y desarrollo de animaciones e interactivos. (VERIFICACIÓN - VER2) Programación de elementos de transformación (VERIFICACION - VER3).
A7. (O2) 7 Diseño Editorial	
DI, PR, CR	Integración de los elementos anteriores para asegurar la consistencia en el soporte digital. (VERIFICACIÓN - VER4)
AM2 (O3) M2 Mantenimiento de edición digital	
RDE, RDG, RDT.	<i>Mantenimiento</i> estructural de los elementos gráficos y de contenido concernientes a la actualización de la publicación digital.
MONITOREO Y CONTROL	
A8. (O3) 8 Control de calidad	
RDE PR	Revisa los aspectos editoriales de la publicación digital en cumplimiento con la <i>Especificación de requerimientos</i> . (VERIFICACIÓN - VER5)
	Incorporar a la base de conocimiento final
	El programador realiza un monitoreo de los elementos que integra en la transformación (VERIFICACIÓN - VER6)
AM3 (O3) M3 Control de calidad del mantenimiento	
RDE, RDG, RDT.	Control de cambios en el mantenimiento de la publicación digital
CIERRE	
A9. (O3) 9 Revisión y validación tecnológica	
RDT, PR	El PR verifica los aspectos de funcionalidad de la publicación digital para cumplir con la <i>Especificación de requerimientos</i> en el servidor de transformación preliminar. (VERIFICACIÓN - VER7), el RDT valida (VALIDACION - VAL1), si la validación fue satisfactoria el programador copia la publicación digital al servidor de transformación final. El RDT valida (VERIFICACIÓN - VER8) la correcta integración al servidor de transformación final.
A10. (O3) 10 Entrega	
RDT	Define y establece la logística de entrega. Crea una copia de seguridad y hace la entrega en un CD-ROM.
	Se incorporan a la Base de conocimiento final las modificaciones que se hayan incorporado al proceso
AM4 (O3) M4 Entrega de publicación actualizada	
RDE, RDG, RDT.	Cierre del mantenimiento de la publicación digital.



Verificaciones y validaciones

Verificación o validación	Actividad	Producto	Rol	Descripción
VER1	A6	Texto codificado	CR	
VER2	A6	Imágenes procesadas	DI	
VER3	A6	Elementos de transformación	PR	
VER4	A7	Elementos integrados	CR	
VER5	A8	Aspectos editoriales	CR	
VER6	A8	Elementos de programación	PR	
VER7	A9	Funcionalidad	PR	
VER8	A9	Elementos integrados	RDT	
VAL1	A9	Elementos finales	RDT	

Incorporación a la	Producto	Forma de aprobación
--------------------	----------	---------------------

base de conocimiento (Repositorio documental)	Carpeta de Entregable	Validación en HTML 4.01 Transitional, Niveles de Accesibilidad de Prioridad 1 y 2, con herramienta AcceVerify - HiSoftware
	Carpeta de Entregable	Inspección e informe de anomalías de elementos en el contenido final del libro

Recursos de infraestructura	Actividad	Recurso
	A1	Infraestructura de Hardware y Software, responsables del Desarrollo de la Edición Digital de Libros.
	A2	Infraestructura de Hardware y Software, responsable de Bibliotecología.
	A3	Infraestructura de Hardware y Software, RDE – RDT, RDE. RDG
	A4	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	A5	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	A6	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	A7	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	A8	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	A9	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	A10	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	AM1	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	AM2	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	AM3	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG
	AM4	Infraestructura de Hardware y Software, RDP – RDT, RDE. RDG

Situaciones excepcionales	Los requerimientos del sistema pueden impedir que la validación de estándares se cumpla al 100%.
----------------------------------	--

ANEXO – A3

Es importante mencionar que no existe una metodología de software universal, las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que la metodología sea configurable. RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura parte de la que tomaremos en nuestra metodología adaptada.

La Arquitectura de un sistema es la organización o estructura de sus partes más relevantes.

Un arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades.

RUP establece refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo evolutivo

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Concepción:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.
- **Mantenimiento:** una vez instalado el producto, el usuario realiza requerimientos de ajuste, esto se hace de acuerdo a solicitudes generadas como consecuencia del interactuar con el producto.

