

2. Capítulo 2. Trabajo relacionado

Si consigo ver más lejos es porque he conseguido auparme a hombros de gigantes.

Isaac Newton (1642-1727)
Matemático y físico británico.

Como el propósito de esta tesis es la implementación de un sistema que genere interfaces de manera dinámica para la administración de colecciones en base a esquemas, a continuación se presenta un análisis de los diferentes estándares usados por ColeXión.

2.1. Editores de esquemas basados en estándares

2.1.1. Reggie

Reggie es un editor que permite la creación de metadatos usando distintos estándares a través de una sola aplicación. Los estándares en los que puede crear los metadatos son HTML (HyperText Markup Language) 3.2 y 4.0, RDF (Resource Description Framework) y RDF abreviado. Mientras que HTML es un lenguaje de marcado para la creación de páginas Web, RDF es un modelo para representación de metadatos avalado por W3C (World Wide Web Consortium). Reggie se basa en un archivo previamente almacenado con el esquema especificado para obtener todos los detalles de sus elementos. Esta herramienta permite que se pueda usar cualquier otro esquema, para esto se le debe indicar el URL del esquema a usar y Reggie obtendrá los elementos de este nuevo esquema para que el usuario haga su modelo¹.

Reggie es un editor que se encuentra a disposición de los usuarios a través de una página Web en forma de *applet*. La Figura 1 presenta la primera interfaz de Reggie, donde se puede especificar el esquema a usar y el idioma, o bien, especificar un esquema propio.

¹ <http://metadata.net/dstc/>

Please choose a schema and language:

Schema: Language:

Or, enter the URL for another schema:

If describing a web page, enter the URL:

Figura 1. Primera interfaz de Reggie (se incluye la figura con permiso de los autores).

Una vez que se indicaron los campos correspondientes con las necesidades del usuario, Reggie presenta una nueva interfaz, donde el usuario puede crear su propio esquema basado en lo que necesite.

The screenshot shows the 'Base De Dublín' interface. At the top, there is a header with the logo, 'Base De Dublín' text, 'About Base De Dublín' button, 'Send feedback to: rdu-info@dstc.edu.au', 'DISTRIBUTED SYSTEMS TECHNOLOGY CENTRE' logo, 'About Reggie' and 'Quit' buttons, and 'Reggie v1.6 © 1998 DSTC Pty Ltd'.

Below the header is a toolbar with buttons: 'Hide / Restore Fields', 'Clear All Values', 'Select a Syntax...', 'Preview', and 'Export...'. The main area contains four fields for customization:

- Título:** Input field with a '+' icon and an 'X' icon. Below it, 'SubElement: None' and 'Language: Spanish'.
- Autor o Creador:** Input field with a '+' icon and an 'X' icon. Below it, 'SubElement: None' and 'Language: Spanish'.
- Subject y Claves:** Input field with a '+' icon and an 'X' icon. Below it, 'Scheme: None' and 'Language: Spanish'.
- Descripción:** Input field with a '+' icon and an 'X' icon. Below it, 'Language: Spanish'.

Figura 2. Interfaz de personalización (se incluye la figura con permiso de los autores).

La Figura 2 presenta la interfaz donde el usuario puede personalizar su modelo. En esta interfaz el usuario puede llenar cada campo del esquema escogido, eliminar los elementos que no necesite, repetir los que desee si necesita más de uno con el signo de más (+), escoger la sintaxis en la que desea su modelo y leer una breve descripción de

qué es lo que debe ir en cada campo. Esta última funcionalidad se hace con el botón de signo de interrogación (?) y es una opción para poder hacer una descripción adecuada.

Por último, la Figura 3 presenta el resultado del modelo generado por Reggie con los atributos del usuario.



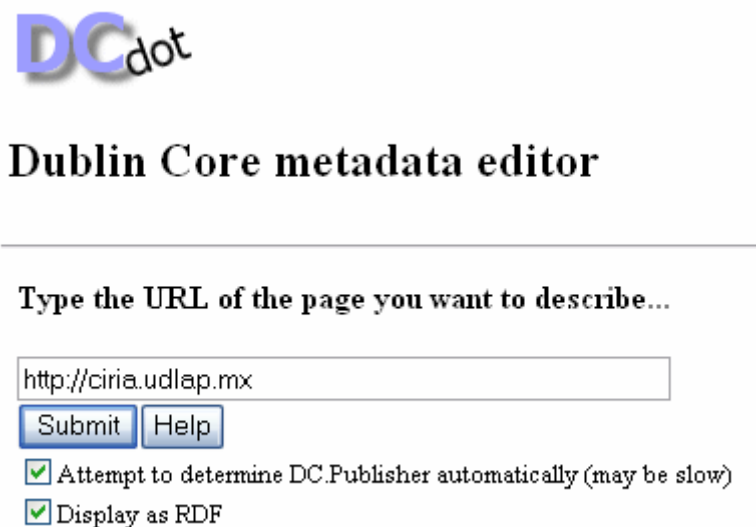
Figura 3. Resultado arrojado por Reggie (se incluye la figura con permiso de los autores).

A pesar de que Reggie genera interfaces de manera dinámica para cualquier esquema que maneje, el resultado final de esto es un documento en XML. Si se quisiera usar esto para administrar colecciones, se debería de copiar este documento en XML y almacenarlo en una base de datos por cada registro que se quisiera guardar. Reggie no realiza ninguna conexión con una base de datos para almacenar el documento generado.

Los autores de Reggie (comunicación personal disponible en el Apéndice 8) señalan que la incapacidad de este sistema para validar la entrada es un problema severo y en el cual están trabajando en DSTC, pero aún no está disponible una versión que lo permita.

2.1.2. DC-dot

DC-dot funciona de una manera similar a Reggie; en el sitio de DC-dot el usuario puede ingresar la página Web que quiere describir, seleccionar si quiere que se incluya el autor y si quiere que se despliegue el resultado en el formato de RDF². La Figura 4 muestra el espacio para hacer esta operación.



The image shows the DC-dot web interface. At the top is the DCdot logo. Below it is the title "Dublin Core metadata editor". A horizontal line separates the title from the input area. The input area contains the text "Type the URL of the page you want to describe...". Below this is a text input field containing "http://ciria.udlap.mx". To the right of the input field are two buttons: "Submit" and "Help". Below the buttons are two checked checkboxes: "Attempt to determine DC.Publisher automatically (may be slow)" and "Display as RDF".

Figura 4. Campos para el uso de DC-dot (se incluye la figura con permiso de los autores).

Una vez que se genera el documento, el usuario puede modificarlo, si considera que el documento no está bien estructurado o si faltan atributos que no fueron extraídos por DC-dot. La Figura 5 muestra el resultado de la descripción del sitio del CIRIA producida por este editor.

² <http://www.ukoln.ac.uk/cgi-bin/dcdot.pl>

If necessary, edit the values in the boxes below, and
Convert metadata to

Title	<input type="text" value="CIRIA. Centro Interactivo de Recursos de Informaci� Aprendizaje"/>
Creator (author)	<input type="text"/>
Subject or keywords	<input "="" type="text" value="Biblioteca; Bibliotecas; CUDI; Mapa; Documentos? Antiguos?en?LOa; Tesis; Especiales; Servicios; Preguntas; Grupo; Biblioteca?Franciscana; Orientaci�?usuarios; Amigos?de?las?Bibliotecas? /SABIUS; Soluciones; Programa?de?Licenciatura:"/>
Description	<input type="text"/>
Publisher	<input type="text"/>

Figura 5. Campos para la modificaci n de un esquema en DC-dot (se incluye la figura con permiso de los autores).

Todos los esquemas son generados en el est ndar de Dublin Core, pero esto no quiere decir que sea el  nico est ndar en el que se pueda tener la descripci n de una p gina Web. Una vez que el usuario ha determinado que la descripci n de la p gina es la correcta, puede escoger exportarlo a alg n otro esquema, teniendo la posibilidad de importar el modelo al est ndar que m s le convenga.

2.1.3. *MetaEdit*

MetaEdit es un editor de metadatos de la compa a australiana DSTC. Este editor forma parte de la *suite* de soluciones para metadatos de DSTC (MetaSuite), sin embargo, puede ser utilizado como una aplicaci n aparte o junto con toda la suite de herramientas que provee DSTC. La manera en la que funciona MetaEdit es a trav s de su interfaz que permite que se genere un est ndar con ayuda de sus herramientas de edici n como la terminaci n de elementos al momento de escribir, validaci n del modelo actual, capacidad de crear est ndares propios con la herramienta de MetaSchema. Esta herramienta soporta est ndares como Dublin Core, Australian Government Locator Service (AGLS) (AS5044), y Australia and New Zealand Land Information Council (ANZLIC)³.

³ <http://www.dstc.edu.au/1025.html>

2.2. Herramientas para administrar colecciones basadas en esquemas propios

2.2.1. El Pescador

El sitio de la BUAP para la colección de las marcas de fuego presenta una interfaz fácil de manejar que permite hacer búsquedas por palabras clave de algún libro o navegar por toda la colección para encontrar lo que se esté buscando [Green, 2006]. La Figura 6 presenta esta primera pantalla.



Figura 6. Primera pantalla de la colección de marcas de fuego (se incluye la figura con permiso de los autores).

Al hacer una búsqueda en el catálogo o navegar por todos los registros, los elementos son descritos de manera breve con una pequeña ficha. Al momento de querer averiguar más acerca de cierto libro, la ficha crece y presenta una serie de atributos como son la clase, categoría, descripción, periodo, imagen, dimensiones, anotación manuscrita y la forma, que es la marca hecha con aceros al rojo vivo para marcar a qué orden perteneció el libro.

2.2.2. XMLibris

XMLibris tiene como objetivo el almacenamiento, consulta, navegación y visualización de colecciones de documentos antiguos para su publicación en línea [Sánchez, 2006]. En teoría, tanto XMLibris como Pescador permiten administrar cualquier colección, sin embargo estos dos sistemas no son flexibles en el uso de algún otro esquema para representar metadatos que no sea el suyo. No se analizarán otros elementos de XMLibris más que el de la interfaz, al ser el tema principal de esta tesis.

En la interfaz presentada por XMLibris se incluye un campo para búsquedas así como herramientas que permiten la navegación dentro de un elemento de la colección de libros antiguos. La Figura 7 presenta la interfaz de XMLibris.



The screenshot displays the XMLibris web interface. At the top left is the logo "xmlLibris" with a stylized "X" icon. To its right is a search input field and a "Buscar" button. Below this is a breadcrumb trail: "> xmlibris > UDLA > Biblioteca Franciscana > San Gabriel > adlo". The page number is shown as "Página (1 de 12)" with navigation links: "|<primera <<anterior 1 siguiente>> última>|". There is also a "Página" input field and an "Ir" button. A horizontal menu contains links: "•Buscar", "•Anotaciones", "•Efectos", and "•Más opciones". Below the menu, it says "Secciones (1) preliminares,". The main content area shows a preview of a book page with the following text:

Pag. 1.

LIBRO PRIMERO
DE LA VIDA PRODIGIOSA
DEL VENERABLE SIERVO DE DIOS
F. SEBASTIAN DE APARICIO
RELIGIOSO LEGO DE LA REGULAR
Observancia de N. S. P. S. Franciscano.

CAPITULO I.
*Patria, Padres, y Nacimiento del V. Aparicio,
y ejmeros de la Divina Providencia en conser-
var su vida.*

A VILLA DE GUDINA,
casi desconocida en otro
tiempo al mismo Reyno de
Galicia por su pequeñez,
se hizo lugar entre las mas
célebres Ciudades del Or-
be desde el dia veinte de
Enero del año de mil qui-
nientos y dos, en que vió
nacido en su suelo à nues-
tro Sebastian. Fueron los Padres de éste, Juan de
Aparicio, y Therea del Prado, humildes Labrado-
res.

At the bottom of the interface, there are links for "pequeño - normal - grande" and the version information "xmlLibris v 0.81 2006-01-31".

Figura 7. Interfaz de navegación de XMLibris (se incluye la figura con permiso del Laboratorio ICT de la UDLA).

En cada página de los libros que son parte de la colección de XMLibris se pueden hacer anotaciones o se puede ampliar la página para poder apreciar mejor algunas de las características del libro. La descripción de los elementos de la colección de XMLibris está hecha bajo el estándar de Dublin Core. La interfaz que presenta XMLibris para registrar los metadatos de los libros es un buen ejemplo de lo que pretende producir ColeXión al ser una interfaz sencilla de usar para registrar metadatos. La Figura 8 presenta esta interfaz.

The image shows the 'Administración' section of the XMLibris website. At the top, there is a navigation bar with links: 'Principal', 'Inicio', 'Datos de contacto', 'Crear colección', 'Aregar páginas', 'Registrar libro', 'Ayuda', and 'Acerca de...'. Below the navigation bar, the page title is 'Colección'. There is a dropdown menu showing 'xmlibris' and a search icon. Below that, the breadcrumb path is '> [xmlibris](#) > [system](#)'. The form contains several fields for metadata registration:

- Fecha de registro (YYYY-MM-DD) [hoy]: 2006-11-5
- Colección: /db/xmlibris/system
- Clave (única): [Empty text box]
- Nombre del registro: [Empty text box]
- Título DC:Title: [Empty text box]
- Autor DC:Creator: [Empty text box]
- Fecha de edición DC:Date (YYYY-MM-DD): [Empty text box]
- Editor DC:Publisher: [Empty text box]
- Formato DC:Format: image/jpg
- Ubicación DC:Identifier (url): http://localhost:8084/xm-dist/visualizar.jsp?uri=xmldb:exist//localf
- Derechos DC:Rights: 2006 Universidad de las Américas - Puebla

At the bottom of the form is a button labeled 'Registrar Metadatos'.

Figura 8. Interfaz para almacenar metadatos en XMLibris (se incluye la figura con permiso del Laboratorio ICT de la UDLA).

Los ejemplos anteriores de XMLibris y Pescador son eficaces al almacenar algún elemento que pueda ser descrito usando el esquema propio de cada sistema. Sin

embargo, si se trata de almacenar algún elemento de distinta naturaleza, el resultado será una pobre descripción de este elemento y ni Pescador ni XMLibris permiten usar otro esquema que no sea el suyo.

2.3. Generación automática de interfaces

El tema de la generación automática de interfaces no es nuevo, es un tema que ha sido investigado para diferentes aplicaciones. Desde la generación de interfaces para aplicaciones de negocios en base a reglas de producción, constantes y limitantes [Vanderdonckt, 1995]; generación de interfaces basadas en modelos declarativos [Schlungbaum, 1996]; generación automática de interfaces tipo control remoto para dispositivos móviles en base a plantillas [Nichols, 2004, 2006]; generación automática de interfaces en base a código que se interpretará en la creación de interfaces pero sin que el usuario la diseñe [Dewan, 1990]; generación automática de interfaces para bases de datos en base al esquema de las bases de datos [Pizano, 1993]; generación automática de interfaces en base al modelo de datos de la aplicación [de Baar, 1992]; creación automática de interfaces en base a los comentarios hechos en el código de la aplicación [Jelinek, 2004]; generación automática de interfaces para aplicaciones programadas en C++ [Beshers, 1989], entre otros. No es la intención ahondar en cada una de estas investigaciones, ya que se presentan como ejemplo de que, aunque se generen interfaces de manera automática, éstas se enfocan sólo a un aspecto de usabilidad, es decir, no hay alguna interfaz universal como los controles universales para los electrodomésticos. Los cuales, aun siendo universales, tienen que programarse a fin de que cumplan su objetivo adecuadamente [Nichols, 2006].

El tema de la creación automática de interfaces se consideró por [Schlungbaum, 1996] como el futuro en el desarrollo de aplicaciones, ya que el tiempo que se dedica al diseño y elaboración de las interfaces suele ser demasiado, y en ocasiones puede llegar a ser una labor compleja, frustrante y tediosa [Schlungbaum, 1996] [Dewan, 1990] [Pizano, 1993] [Jelinek, 2004] [Scott, 1988]. La intención de ColeXión es enfocarse al ámbito de generación automática de interfaces para administrar colecciones, facilitando la creación de interfaces parecidas para colecciones distintas. Esto porque la mayoría de las interfaces para administrar colecciones se basan en formularios, siendo común para las personas el llenar aplicaciones en la vida diaria [Pizano, 1993]. Resulta también muy

tedioso que se tenga que implementar una interfaz para cada colección cuando estas son parecidas, ColeXión ahorra la redundancia en la programación de interfaces similares [de Baar, 1992].

2.4. Estándares de ColeXión

Los estándares que usa la herramienta propuesta por esta tesis son los siguientes:

- METS (Metadata Encoding & Transmission Standard)
- MODS (Metadata Object Description Schema)
- MARCXML (MACHINE- Readable Cataloging XML Schema)
- Dublin Core (DC)

Se escogieron estos esquemas ya que son los más comunes para la descripción de colecciones y están respaldados por alguna organización importante a nivel mundial. Para el caso de METS, MODS y MARCXML, es la biblioteca del congreso de los Estados Unidos quien los respalda, mientras que para el caso de DC existe toda una iniciativa que ese encarga de mantener este esquema. La importancia de que alguna institución importante respalde los esquemas tiene que ver con el aspecto de que un buen esquema produce metadatos eficientes que no pongan en riesgo la información que representan [Shankaranarayanan, 2006]. A continuación se presentan los esquemas que usa ColeXión.

2.4.1. Dublin Core

Dublin Core (DC) es un estándar de metadatos que surge como parte de la iniciativa del mismo nombre (Dublin Core Initiative) en 1995 en Dublin, Ohio. El objetivo de esta iniciativa es promover la adopción de estándares de metadatos para describir recursos y permitir la creación de sistemas para “descubrimiento de información” más inteligentes. La misión de esta iniciativa es proveer estándares sencillos para facilitar el compartir, administrar y descubrir información; y lo hace a través del desarrollo y soporte de estándares internacionales para describir recursos [Dublin Core, 2006].

Dublin Core presenta dos variantes, el esquema calificado (*qualified*) y el sencillo. La diferencia entre uno y otro es que el esquema sencillo cuenta con 15 atributos para

describir de manera general al recurso, mientras que el esquema calificado cuenta con 40 atributos para refinar estas descripciones. En la Tabla 1 se hace una descripción del esquema sencillo de Dublin Core. La Tabla 2 presenta la manera en la que se puede refinar el esquema sencillo de Dublin Core usando los elementos del esquema calificado. Ejemplos del refinamiento del esquema sencillo de Dublin Core se presentan en el Apéndice 1.

Elemento	Descripción
Contribuidor (contributor)	La entidad responsable de hacer contribuciones al contenido del recurso.
Cobertura (coverage)	El objetivo del contenido del recurso
Creador (creator)	La entidad primaria responsable de la creación del contenido del recurso
Fecha (date)	Una fecha asociada con algún evento en el ciclo de vida del recurso
Descripción (description)	Un resumen del contenido del recurso
Formato (format)	La manifestación física o digital del recurso
Identificador (identifier)	Referencia única del recurso en un contexto dado
Idioma (language)	Idioma del contexto intelectual del recurso
Editor (publisher)	La entidad responsable de hacer disponible el recurso
Relación (relation)	Referencia a un recurso relacionado
Derechos (rights)	Información acerca de los derechos sobre el recurso
Fuente (source)	Referencia al recurso del que está derivado el presente recurso
Tema (subject)	El tema del contexto del recurso
Título (title)	El nombre dado al recurso
Tipo (type)	La naturaleza o género del contenido del recurso

Tabla 1. Elementos del esquema sencillo de Dublin Core.

Elemento esquema sencillo	Elemento(s) con que se puede refinar
Contribuidor (contributor)	-
Cobertura (coverage)	Espacial (spatial) Temporal (teporal)
Creador (creator)	-
Fecha (date)	Creado (created) Válido (valid) Disponible (available) Publicado (issued) Modificado (modified) Fecha en la que se aceptó (date accepted) Fecha en que se patentó (date copyrighted) Fecha en que se sometió (date submitted)
Descripción (description)	Tabla de contenido (table of contents) Sinopsis (abstract)
Formato (format)	Tamaño o duración (extent) Medio (medium)
Identificador (identifier)	Cita bibliográfica (bibliographic citation)
Idioma (language)	-
Editor (publisher)	-
Relación (relation)	Es versión de (is version of) Tiene versión (has version) Es reemplazado por (is replaced by) Reemplaza a (replaces) Es requerido por (is required by) Requiere (requires) Es parte de (is part of) Tiene parte de (has part) Es referenciado por (is referenced by) Hace referencia a (references) Es formato de (is format of)

	Tiene formato de (has format) Cumple con (conforms to)
Derechos (rights)	Derechos de acceso (access rights)
Fuente (source)	-
Tema (subject)	-
Título (title)	Alternativo (alternative)
Tipo (type)	-

Tabla 2. Refinamiento del esquema sencillo de Dublin Core.

2.4.2. MARCXML

MARCXML es un marco de trabajo desarrollado por la biblioteca del congreso y por la oficina MARC de estándares. El objetivo de MARCXML es proporcionar una manera flexible y extensible a los usuarios del estándar MARC (MACHINE- Readable Cataloging) para trabajar con datos en este formato. MARC es un estándar para la representación y comunicación de información bibliográfica en forma que pueda ser leída por una máquina. Este marco de trabajo pretende incluir esquemas, hojas de estilo y herramientas de software. Todo esto desarrollado y mantenido por la biblioteca del congreso de los Estados Unidos.

El esquema de MARCXML puede ser usado para almacenar un recurso en el formato completo de MARC pero en XML, como una extensión del esquema de METS, o actuar como un canal para permitir diferentes transformaciones de MARC a otros esquemas como Dublin Core. MARCXML, a diferencia de MODS, asegura que las transformaciones de un esquema a otro (MARC a MARCXML) serán sin perder información específica y que además se podrá hacer en dos sentidos [MARCXML, 2006].

MARCXML cuenta con seis tipos complejos y dos elementos que se mencionan en el Apéndice 2.

2.4.3. MODS

MODS es un esquema para elementos bibliográficos que se puede usar para una gran variedad de propósitos, pero que está orientado principalmente hacia aplicaciones bibliotecarias. Este estándar está desarrollado por la biblioteca del congreso y por la oficina de estándares MARC y todavía se encuentra en desarrollo. MODS está basado en los atributos de MARC; la diferencia es que MODS usa descriptores más accesibles para los usuarios ya que sus nombres hacen referencia a los metadatos que contienen, no como MARC que usa un código numérico para el cual el usuario debe haber tenido alguna capacitación previa para usarlo [MODS, 2006].

MODS puede ser usado en sustitución del formato Z39.50 que permite la recuperación en diferentes catálogos bibliográficos, por separado o simultáneamente, desde una única interfaz de búsqueda; como una extensión del esquema de METS (este estándar se define más adelante) ya que METS permite la integración de otro esquema para refinar la descripción de metadatos; como representación de metadatos para la minería de datos en algún sistema de minería de datos; para crear descripciones propias o para representar un registro simplificado de MARC en XML. Las ventajas de MODS son que sus elementos son más ricos que los de Dublin Core, son más compatibles con las bibliotecas que los de ONIX (Online Interactive eXchange), que es un estándar para representar libros, publicaciones periódicas y video de manera electrónica; el esquema de MODS está más orientado al usuario final que el esquema completo de MARCXML y sus elementos son más simples que los del formato completo de MARC. Las limitaciones principales que presenta MODS son que, a pesar de que está basado en atributos de MARC, no se puede hacer una conversión de ida y vuelta entre estos dos estándares, sólo es posible de MARC a MODS y en ocasiones la información específica de un registro puede no ser interpretada de la manera correcta. MODS cuenta con veinte elementos descritos en el Apéndice 3.

2.4.4. METS

METS surge como una iniciativa para aprovechar el trabajo del proyecto *Making of America II*, cuyo objetivo es proponer un estándar para los objetos de una biblioteca digital y poder codificar metadatos descriptivos, administrativos y estructurales en cada uno de estos objetos [The Making of America II, 1998]. METS pretende proporcionar

un documento en XML para codificar metadatos necesarios para la administración de bibliotecas digitales, pero también pretende que este documento sirva para hacer intercambios entre repositorios de bibliotecas digitales [METS, 2006]. Las bondades de este estándar para poder hacer intercambios entre repositorios y poder incluir elementos de otros estándares como MODS o DC hacen de METS la mejor opción para hacerlo el protocolo de intercambio en *ColeXión*.

Existen siete secciones principales en un documento de METS, las cuales se describen a continuación.

Encabezado de METS (METS Header) – Esta sección incluye información que describe al documento de METS en sí. Incluye información como quién es el creador, editor, entre otros.

Metadatos descriptivos (Descriptive Metadata) – En esta sección es donde se puede incluir atributos de otros estándares o metadatos descriptivos que no se incluyan como parte del documento de METS, como por ejemplo registros en algún sitio Web con el formato MARC.

Metadatos administrativos (Administrative Metadata) – Esta sección proporciona información de cómo es que los archivos fueron creados, quién es el autor, así como metadatos del objeto original del que proviene el objeto digital.

Sección de archivos (File Section) – En esta sección se enlistan todos los documentos que contengan información que tenga que ver con las versiones electrónicas del objeto digital.

Mapa estructural (Structural Map) – Este es el corazón del documento de METS. Describe una estructura jerárquica del objeto digital y liga los elementos de esa estructura a los archivos y metadatos pertenecientes al objeto.

Ligas estructurales (Structural Links) – Esta sección permite que se creen hipervínculos entre los nodos de la estructura de un objeto digital hecha en la sección de *Structural Map*. Esto es importante al momento de describir un sitio Web.

Comportamiento (Behavior) – Esta sección permite asociar alguna acción a algún elemento del objeto digital. Esta acción se logra apuntando a una sección de código que la ejecutará.

Actualmente existen dos tipos importantes de herramientas que tienen que ver con metadatos y esta tesis: los editores de esquemas de metadatos basados en estándares y las herramientas para administrar colecciones basadas en esquemas propios. Tal como

se puede apreciar, las dos herramientas son excluyentes, es decir, una herramienta que facilite la administración de colecciones no permite que se modifique el esquema bajo el que administra sus elementos con la facilidad con la que lo permite un editor, y un editor de esquemas es sólo una herramienta que facilita la creación de un diseño funcional de metadatos.

En el siguiente capítulo se abordará el tema del diseño de *ColeXión*.