



Introducción a los metadatos para las colecciones digitales

José Manuel Barrueco

Índice general

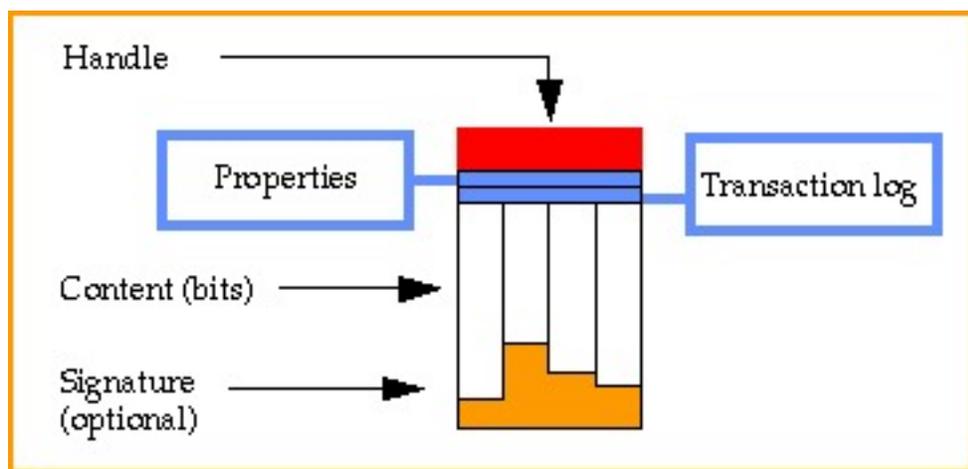
Índice general	1
1 Los objetos digitales	3
1.1. Conceptos fundamentales	3
2 Identificadores	5
2.1. Identificadores para objetos digitales	5
2.2. Sistemas de identificación para objetos digitales	6
3 XML: el estándar por encima de los estándares	11
3.1. Un breve tutorial XML	11
4 Metadatos	19
4.1. ¿Qué son los metadatos?	19
4.2. Aspectos poco conocidos de los metadatos	20
4.3. ¿Para qué sirven los metadatos?	21
4.4. Los diez mandamientos de los metadatos	22
4.5. Principios en la asignación de metadatos	23
4.6. Estructurando los metadatos	24
4.7. Calidad de los metadatos	26
5 Algunos esquemas de metadatos	29
5.1. Esquemas y perfiles de aplicación de metadatos	29
5.2. Esquemas de metadatos descriptivos	30
5.3. Esquemas de metadatos técnicos	33
5.4. Esquemas de metadatos de preservación	34
5.5. Dando unidad, el esquema METS	35
5.6. Perfiles de aplicación de metadatos	37
6 Uso de metadatos en las colecciones del SBD	39

Capítulo 1

Los objetos digitales

1.1. Conceptos fundamentales

1. El marco conceptual en el que se asientan los repositorios o colecciones digitales de nuestras bibliotecas fue definido por Khan y Wilensky en 1995¹. Según estos autores el sistema funciona de la siguiente manera: un usuario con material para ser almacenado en el sistema, construye con él un **objeto digital**. Un objeto digital es una estructura de datos cuyos componentes principales son: (1) el material digital o datos, (2) información sobre los datos o metadatos y (3) un identificador único para ese material. Ese objeto es **depositado** en uno o más **repositorios** desde los cuales puede ser recuperado por los lectores o distribuido a proveedores de servicios a los usuarios finales.

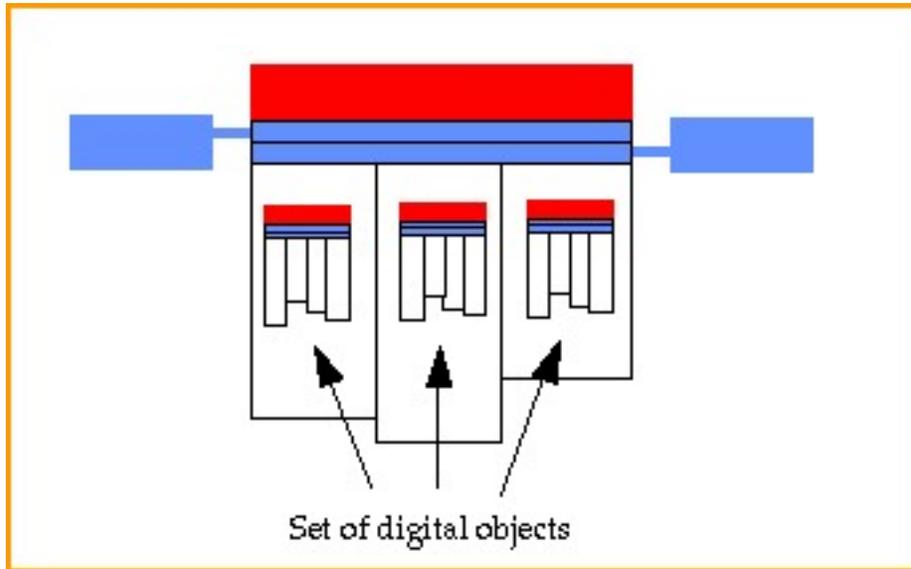


2. Un **repositorio** es un espacio de almacenamiento accesible por internet en el cual se conservan los objetos digitales para ser localizados y recuperados. El repositorio debe tener mecanismos para añadir

¹ <http://www.cnri.reston.va.us/k-w.html>

nuevos objetos a las colecciones (depósito) y para hacerlos disponibles, además de otros potenciales servicios y sistemas de gestión.

3. Existen dos tipos de objetos digitales: elementales o sencillos y compuestos o complejos.



4. Utilizar identificadores únicos es imprescindible en un repositorio. Es fundamental que sean persistentes en el tiempo lo que excluye el uso de identificadores ligados a una localización específica. Deberían persistir incluso si la organización que lo ha asignado desaparece. Finalmente deben existir sistemas informáticos capaces de **resolver** el identificador de forma rápida mediante la asociación de un identificador con su correspondiente localización.
5. En una colección digital hay que diferenciar entre el objeto digital almacenado y el resultado que el usuario obtiene. Por lo tanto, se debe distinguir, al menos, entre los objetos tal como fueron creados, los objetos almacenados en el repositorio y los objetos tal como se ofrecen al usuario. El objeto final será el resultado de ejecutar un programa sobre el objeto almacenado. Este procesamiento del objeto almacenado puede ser una simple transferencia de ficheros o algo mucho más elaborado como dar unidad a un objeto complejo.

Ejercicio 1

Buscar al menos cinco objetos digitales simples y uno compuesto.

Capítulo 2

Identificadores

2.1. Identificadores para objetos digitales

Un principio fundamental de la arquitectura de un repositorio establece que todos los objetos almacenados en él deben tener un identificador persistente. Dicho identificador debe poder ser resuelto para localizar y, dados los permisos adecuados, interactuar con el objeto. El identificador debe ser independiente de cambios en la propiedad, localización, formato de los datos, nivel de seguridad, etc. dado que tales variables no son una parte intrínseca del objeto digital.

En general, un identificador es un medio conciso para referenciar algo. Con el término "identificador" podemos referirnos, al menos, a tres cosas:

- Una cadena de caracteres, normalmente un nombre o número.
- Una especificación, que prescribe cómo construir dichas cadenas. Debe especificar la sintaxis correcta, caracteres permitidos y excluidos, puntuación, elementos obligatorios, uso de mayúsculas y minúsculas, etc. Un ejemplo de especificación de este tipo es la norma ISO que regula el ISBN. Ahora bien, la norma por si misma no permite a alguien construir y registrar un ISBN válido, sino que es necesaria una organización que lo haga posible.
- Un esquema para la implementación de tal norma. Por ejemplo la ISBN International Agency que implementa la norma ISBN mediante la asignación de prefijos a editoriales, registrando los números asignados, dando guías sobre como proporcionar los ISBNs, etc.

Algunos conceptos importantes a tener en cuenta en relación a los identificadores son:

- Un identificador debe ser único. Es un requisito que cada cadena designe a una sólo entidad o referente. La relación inversa no es igual puesto que una entidad puede tener varios identificadores.
- Resolución. Es el proceso por el cual un identificador es la entrada en un servicio para producir como salida una o más piezas de información relativas al objeto identificado. Por ejemplo un código de barras que representa un ISBN puede ser leído y obtener como resultado los metadatos asociados al mismo. Los visualizadores web son un ejemplo de resolución donde la entrada es una url y la salida la página identificada. En un repositorio el procesamiento más habitual será la localización del objeto referenciado. En este sentido, la resolución es el puente entre la identificación de un objeto y el acceso al mismo.
- Interoperabilidad es la capacidad para usar un identificador en servicios fuera del control directo de la autoridad que los ha asignado. Es decir, los identificadores asignados en un contexto pueden ser encontrados en otro lugar o tiempo sin consultar a la institución que lo asignó. Por ejemplo, un cliente puede solicitar un libro a una librería citando su ISBN, sin necesidad de consultar al editor que ha asignado el número.
- Persistencia es el requisito de que un identificador, una vez asignado, denotará el objeto referenciado de forma indefinida, es decir, no será asignado en el futuro a otra entidad aunque el objeto original, la autoridad o institución que lo asignó, hayan desaparecido.

 **En conclusión:**

La gestión de contenidos digitales requiere que los identificadores sean persistentes, únicos, capaces de ser resueltos e interoperables.

2.2. Sistemas de identificación para objetos digitales

Handle

Desarrollado por National Research Initiatives (CNRI), el sistema handle¹ utiliza una arquitectura jerárquica en la cual el CNRI mantiene un Global Handle Registry (GHR) sobre el que operan los Local Handle Service (LHS). Un LHS es responsable de la gestión de handles asociados con múltiples autoridades de nombres, cada una de las cuales tiene definido un espacio local. Estas autoridades supervisan la creación y gestión de identificadores a los recursos locales.

¹<http://handle.net>

Un espacio de nombres se compone de dos partes: un prefijo que identifica a una autoridad de nombres particular y un sufijo o nombre local asignado por esa autoridad a los objetos digitales.

✓ **Ejemplo de handle**

<Handle Naming Authority>/<Handle Local Name>

145.76/jan2005-rk32498833

Las autoridades de nombres deben ser únicas dentro del sistema y los nombres asignados por ellas localmente deben ser igualmente únicos. Las autoridades pueden ser organizadas jerárquicamente, con entidades subordinadas separadas por punto.

☞ **A saber:**

La Universitat de València está registrada como autoridad de nombres con el prefijo 10550.

Los servicios de resolución se proporcionan por un sistema global que mantiene información sobre handles y los repositorios en los cuales están almacenados los recursos. Las aplicaciones cliente interrogan el GHR para identificar y localizar el LHS asociado con la autoridad de nombres correspondiente. El cliente puede entonces interactuar con dicho LHS. En este proceso se utiliza un protocolo específico del sistema handle que permite, entre otras cosas, la autenticación de los clientes como administradores válidos. Existe una pasarela entre el protocolo handle y otros protocolos con el HTTP. Este permite por ejemplo resolver handles utilizando un visualizador web.

✎ **Ejercicio 2**

<http://hdl.handle.net> es una pasarela para la resolución de handles

Utilizar dicha pasarela para resolver el handle: 10251/9317

<http://hdl.handle.net/10251/9317>

El software necesario para implementar un LHS se distribuye gratuitamente por CNRI, quien también supervisa la gestión del GHR y la asignación de autoridades de nombres. Para el registro como autoridad se requiere el pago de una licencia con periodicidad anual.

Digital Object Identifier: DOI

El DOI² es la principal implementación del sistema handle. Se trata de una iniciativa originalmente desarrollada por CNRI y la Association of American Publishers (AAP) y actualmente gestionada por la International DOI Foundation (IDF).

Establece un marco general para la identificación persistente de recursos digitales, tomado del esquema handle, al que se han añadido funcionalidades nuevas para personalizar determinados servicios para acomodarse a determinados requisitos de la industria editorial. De esta forma, el sistema integra la descripción de recursos y la resolución de identificadores con políticas destinadas a dar soporte para un acceso permanente a los recursos.

El sistema DOI ha sido estandarizado en la norma ISO 26324:2010 aprobada en noviembre de 2010.

El sistema se compone de cuatro componentes. Una sintaxis formal y unas reglas para la asignación de identificadores. Servicios de resolución a través del sistema handle. Un modelo de datos y un diccionario para la descripción de recursos.

La sintaxis del DOI sigue la del esquema Handle donde el prefijo se compone de dos partes separadas por punto. La primera se denomina código de directorio, que es siempre constante como 10, y la segunda código de registro. La IDF es la encargada de asignar códigos de registro a las instituciones que deseen registrar DOIs. El sufijo es asignado localmente por las autoridades.

Existe también la figura de las Agencias de Registro. A cada una de ellas se les asignan bloques de prefijos para asignar a las autoridades locales. Son las responsables de mantener la infraestructura que permita a las autoridades asignar y gestionar DOIs. Las autoridades por su parte deben mantener actualizada la información local ya que la persistencia depende del mantenimiento de información sobre valores a los cuales un DOI pueda ser resuelto. El DOI Kernel Metadata Declaration (KMD) asegura la interoperabilidad de los metadatos descriptivos. Este formato está compuesto de los siguientes elementos:

- DOI: DOI name assigned to the resource
- resourceIdentifier(s): Identifier within another system(s) of identification
- resourceName(s): Common name of the resource

²<http://www.doi.org>

- `principalAgent(s) agentRole(s)`: Information about resource creation, publication, etc.
- `StructuralType`: Structure of the resource (i.e., physical fixation, digital fixation, performance, or abstract work)
- `mode(s)`: Mode of perception (i.e., audio, visual, audiovisual, or abstract)
- `ResourceType`: Description of the resource type (e.g., audio file, serial article, PDF, etc.)

Los DOIs pueden resolverse a través de la tecnología handle que hemos visto. Un DOI puede resolverse a una o varias entidades relacionadas.

El CNRI ha desarrollado un plug-in³ para los principales visualizadores de forma que se puede hacer una resolución directa introduciendo en la barra de direcciones un DOI (o handle).

El sistema DOI ha sido ampliamente implementado en la industria de la publicación electrónica, con CrossRef actuando como la principal Agencia de Registro. El propósito de esta comunidad es proporcionar servicios efectivos de enlace de referencias a través del uso de OpenURL.

 **Ejercicio 3**

Utilizar el resolutor anterior para recuperar los siguientes DOIs

10.1051/0004-6361:20054289
10.1063/1.362637
10.1063/1.2958286
10.1186/1471-2148-6-105
10.1051/0004-6361:20078690

³http://www.handle.net/other_software.html

Capítulo 3

XML: el estándar por encima de los estándares

Todos los estándares de metadatos que veremos a continuación utilizan XML como forma de presentación. Así por ejemplo podemos representar un campo MARC en la forma tradicional:

```
245 10 Frontier orbitals and organic chemical reactions /|cIan Fleming
```

o como MARCXML:

```
<datafield tag="245" ind1="1" ind2="0">  
  <subfield code="a">Frontier orbitals and organic chemical  
    reactions </subfield>  
  <subfield code="c">Ian Fleming</subfield>  
</datafield>
```

Por ello, se hace necesario tener unos conocimientos básicos de XML al hablar de metadatos.

3.1. Un breve tutorial XML

Entre las fortalezas del XML como medio de representación que se citan habitualmente están:

- El ser una norma totalmente abierta, independiente de cualquier software particular.

- La simplicidad de su sintaxis combinada con una gran flexibilidad en la forma de utilización, en particular su capacidad para codificar estructuras jerárquicas simplemente mediante el anidamiento de etiquetas que permite la representación de relaciones complejas entre componentes de metadatos.

Un documento XML sencillo puede ser el siguiente:

 **Ejemplo de XML**

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Diferencias entre HTML y XML:

- XML se utiliza para almacenar datos estructurados, mientras HTML se utiliza para dar formato y visualizar esos datos.
- En estos momentos XML es la herramienta más común para la transmisión de datos entre cualquier clase de aplicaciones.
- En XML se pueden inventar etiquetas.
- Los documentos XML tienen una estructura jerárquica, donde la primera etiqueta se denomina raíz y el resto hijos.
- Todas las etiquetas deben ser cerradas <Para></Para>
- Las etiquetas diferencian mayúsculas de minúsculas. <Para>es diferente de <para>.
- Las etiquetas deben estar correctamente entrelazadas. Sería incorrecto: <i>texto</i>.



Complicando las cosas

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Un elemento se define como todo lo que hay entre la etiqueta de inicio de elemento y la etiqueta de fin de elemento.

Un elemento puede contener: otros elementos, texto, atributos o una mezcla de varios de estos. Los valores de los atributos deben ir siempre entre comillas.

En el caso anterior el elemento raíz es `<bookstore>`. Todos los elementos `<book>` están contenidos dentro del elemento raíz. A su vez cada `<book>` tiene cuatro hijos: `<title>`, `<author>`, `<year>`, `<price>` y un atributo: `category`.

Los atributos aportan información adicional sobre los datos, que a menudo no es parte integrante de los mismos pero puede ser importante para el software que manipule el elemento. En el caso de `lang` anterior, puede servir para delimitar los elementos que pertenezcan a un determinado idioma.

No hay reglas explícitas sobre cuándo usar atributos y cuándo elementos. En general los datos deberían almacenarse en elementos y los metadatos en atributos. Así por ejemplo es habitual asignar identificadores a los elementos. En este caso el identificador no es un dato sino un metadato. Por ejemplo:



Complicando las cosas

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

Un documento XML debe estar bien formado, no se permiten los errores. En caso de errores las aplicaciones que utilicen el documento no funcionarán. Además para ser válido debe ajustarse a una definición de tipo de documento específica DTD o, como alternativa, un esquema XML.



A saber:

Para saber si un documento XML es correcto podemos validarlo utilizando herramientas como: <http://validator.w3.org/>

Dado que los nombres de los elementos son definidos por el desarrollador se pueden producir conflictos al mezclar diferentes elementos procedentes de diferentes aplicaciones. Para solucionar este problema XML proporciona los espacios de nombres.

Un espacio de nombres se define con el atributo **xm:ns** en la etiqueta inicial de un elemento.



Complicando las cosas un poco más

```
<root
xmlns:h=' 'http://www.w3.org/TR/html4/' '
xmlns:f=' 'http://www.w3schools.com/furniture''>

<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>

</root>
```

Basicamente un espacio de nombres es una colección formal de términos gestionados de acuerdo a una determinada política. En el caso concreto de los metadatos la declaración de espacios de nombres permiten al diseñador de un esquema definir el contexto para un término particular, asegurando por lo tanto que el término tiene una única definición dentro de ese espacio. La declaración de varios espacios dentro de un bloque de metadatos permite a los elementos dentro de esos metadatos ser identificados como pertenecientes a uno u otro conjunto de elementos.

Expresado en forma de lenguaje natural, una declaración de este tipo se leería:

- El conjunto de elementos de metadatos Dublin Core está definido en un lugar identificado por un URI. Todos los elementos dentro del alcance de esta declaración pueden ser reconocidos por el prefijo dc:
- El conjunto de elementos de metadatos IEEE-LOM está definido en un lugar identificado por un URI. Todos los elementos dentro del alcance de esta declaración pueden ser reconocidos por el prefijo lom:.

Utilizando esta infraestructura, es posible seleccionar elementos homónimos procedentes de diferentes esquemas. Así por ejemplo el elemento <dc:title>se puede diferenciar del elemento <lom:title>.

Para presentar el contenido de un documento XML de una forma accesible al lector es necesario transformarlo en un formato adecuado para la visualización. A partir de XML podremos obtener documentos en HTML, PDF, etc. Así por ejemplo es posible aplicar hojas de estilo CSS a un documento XML para obtener una versión HTML utilizable por el usuario.

No obstante las hojas de estilo más recomendables son las XSLT (eXtensible Stylesheet Language Transformations) dado que son mucho más sofisticadas al constituir un auténtico lenguaje de programación.



Codificando los libros que me interesan

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<root xmlns:dc='http://purl.org/dc/elements/1.1'>
<item>
  <dc:title>Aprendiendo XML en media hora</dc:title>
  <dc:creator>García, Pepe</dc:creator>
  <dc:date>1945</dc:date>
  <dc:identifier>10550/01.0001</dc:identifier>
</item>
<item>
  <dc:title>Imposible aprender XML en media hora</dc:title>
  <dc:creator>García, Julian</dc:creator>
  <dc:date>1986</dc:date>
  <dc:identifier>10550/01.0002</dc:identifier>
</item>
</root>
```



Cosas realmente complicadas

```
<?xml version='1.0' encoding='UTF-8'?>
<html xmlns:xsl='1.0'
      xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
      xmlns='http://www.w3.org/1999/xhtml'>
  <body xmlns:dc='http://purl.org/dc/elements/1.1'>
    <p>Vamos a ver la lista de libros que me interesan:</p>
    <ul>
      <xsl:for-each select='root/item'>
        <li>
          <i><xsl:value-of select='dc:title'></xsl:value-of></i>
          de <b><xsl:value-of select='dc:creator'></xsl:value-of></b>
          es del año: <xsl:value-of select='dc:date'></xsl:value-of></li>
        </xsl:for-each>
      </ul>
      <p>A ver lo listo que soy, voy a sacar los libros posteriores a 1980,
        lo conseguiré?</p>
      <xsl:for-each select='root/item'>
        <xsl:if test='dc:date > 1980'>
          Ummmm, creo que es: <br/>
          <xsl:value-of select='dc:title'></xsl:value-of> <br/>
          He acertado!!!
        </xsl:if>
      </xsl:for-each>
    </body>
  </html>
```



Ejercicio 4

Describir utilizando los elementos Dublin Core title, date, creator e identifier uno de los objetos digitales descargados. El espacio de nombres para Dublin Core está especificado en <http://purl.org/dc/elements/1.1>

Capítulo 4

Metadatos

4.1. ¿Qué son los metadatos?

Los metadatos son información estructurada que describe, explica, localiza o facilita la recuperación, el uso o la gestión de un recurso digital.

Existen tres tipos básicos de metadatos:

- Metadatos descriptivos. Describen el recurso digital con propósitos de localización e identificación.
- Metadatos estructurales. Indican cómo los recursos u objetos complejos están ensamblados, por ejemplo, en qué secuencia están ordenadas las páginas para formar capítulos.
- Metadatos administrativos. Proporcionan información para ayudar en la gestión de un recurso, tal como cuándo y cómo fue creado, tipo de fichero que lo contiene junto con otra información técnica, quién puede acceder a él, etc. Hay diferentes tipos de metadatos administrativos que en ocasiones se mencionan como tipos independientes:
 - Metadatos para la gestión de derechos: describen derechos de propiedad intelectual.
 - Metadatos de preservación: contienen información necesaria para archivar y preservar un recurso digital.

Los metadatos pueden describir recursos a cualquier nivel de agregación. Pueden describir una colección, un recurso individual o una parte de un recurso más amplio (por ejemplo, una fotografía en un artículo). Igual que al catalogar se toman decisiones sobre si se debe generar un registro en el catálogo para un conjunto de volúmenes o uno individual para cada uno

de los mismos, igual se deben tomar decisiones a la hora de asignar metadatos. Los metadatos pueden ser usados también para la descripción a cualquier nivel del modelo de información FRBR (Functional Requirements for Bibliographic Records): obra, expresión, manifestación o ítem.

Los metadatos pueden estar embebidos en un objeto digital o pueden ser almacenados de forma separada. Por ejemplo, pueden estar embebidos dentro de un documento HTML o en los encabezamientos de los ficheros de imágenes.

Al almacenar metadatos dentro de los objetos que describe se asegura que no se perderá, se evitan problemas obvios de enlace entre datos y metadatos y se asegura que tanto el objeto como los metadatos serán actualizados conjuntamente.

Como inconveniente de este sistema podemos citar que es imposible embeber metadatos en todos los objetos. También al almacenarlos de forma separada se puede simplificar la gestión de los propios metadatos y facilitar la búsqueda y localización.



A saber:

Actualmente la forma más frecuente de almacenar los metadatos es separados del objeto, en una base de datos, y enlazados entre sí por diferentes procedimientos.

4.2. Aspectos poco conocidos de los metadatos

- Los metadatos no tienen por qué ser digitales, la catalogación tradicional es una forma de asignar metadatos definidos por MARC.
- Los metadatos se refieren a mucho más que a la mera descripción de un objeto. Aunque estamos habituados a trabajar con metadatos para la descripción, se pueden usar también para indicar el contexto, gestión, procesamiento, preservación y uso de los recursos que se están describiendo.
- Los metadatos pueden proceder de una variedad de fuentes. Puede, por ejemplo, ser generado por un humano (ya sea el creador de un objeto digital, un usuario experto o un bibliotecario), puede ser generado por un sistema informático o puede ser inferido a través de una relación con otro recurso.
- Los metadatos son dinámicos en el sentido de que crecen a lo largo de toda la vida del objeto.

- Los metadatos sobre un objeto digital pueden a su vez convertirse en datos de otros recursos dependiendo de los tipos de agregaciones y de dependencias entre objetos de información y sistemas. Es decir, que la distinción entre lo que constituye datos y lo que constituye metadatos puede ser a veces bastante difusa y puede depender de cómo uno desee utilizar un determinado objeto de información.

4.3. ¿Para qué sirven los metadatos?

- **Localización de información.** La función principal de los metadatos, al igual que la de la catalogación tradicional, es la localización de información relevante para una pregunta determinada, al permitir:
 - la búsqueda de recursos a través de determinados criterios
 - la identificación de recursos
 - la agrupación de recursos similares
 - la distinción de recursos diferentes
- **Organización de recursos electrónicos.** Como el número de recursos basados en la web crece de forma exponencial, se hace necesaria la existencia de agregadores para la organización de recursos especializados. Esta agregación puede realizarse como páginas web estáticas, con los nombres y los enlaces incrustados en el HTML, sin embargo, es mucho más eficiente contruir tales páginas de forma dinámica desde metadatos almacenados en una base.
- **Interoperabilidad.** Al describir un recurso con la ayuda de metadatos se permite que dicha descripción sea entendida tanto por humanos como por sistemas informáticos, de tal forma que se pueda promover la interoperabilidad entre diferentes sistemas. Interoperabilidad es la posibilidad de que múltiples sistemas con diferentes plataformas de software y hardware, estructuras de datos e interfaces, sean capaces de intercambiar datos con una pérdida mínima de contenidos y funcionalidad. Al utilizar esquemas de metadatos normalizados, protocolos comunes de transferencia y tablas de conversión entre diferentes esquemas, se podrán buscar y localizar recursos en la red de forma mucho más efectiva.

Dos formas de interoperabilidad, entre otras, son búsquedas federadas y recolección de metadatos. Para las primeras se puede utilizar el protocolo Z39-50. En este caso no se comparten metadatos sino que se mapean las capacidades de búsqueda de un sistema a un conjunto común de atributos definidos por la norma. Frente a él se sitúa la recolección, empleada, por ejemplo, por la Open Archives Initiative. Con ella todos los proveedores de datos deben convertir sus metadatos a un conjunto común de elementos que serán recogidos por los

proveedores de servicios para crear un índice central que ofrecerá la posibilidad de una búsqueda cruzada entre proveedores independientemente de los formatos de metadatos usados por cada uno de los repositorios participantes.

- **Identificación digital.** La mayoría de los esquemas de metadatos incluyen elementos tales como números normalizados para identificar de forma unívoca la obra u objeto que se está describiendo.
- **Archivo y preservación.** Muchos de los proyectos actuales sobre metadatos se centran sobre la recuperación de recursos creados recientemente. Sin embargo, existe una preocupación creciente de que los recursos digitales no sobrevivirán en una forma usable en el futuro. La información digital es frágil, se puede alterar o destruir ya sea de forma intencionada o no. Puede volverse inutilizable a medida que los medios de almacenamiento y las tecnologías software y hardware cambian. La migración de formatos y la emulación del hardware y software actual en plataformas futuras son estrategias que se están empleando para tratar de evitar estos problemas.

Los metadatos son claves para asegurar que los recursos sobrevivirán y continuarán siendo accesibles en el futuro. El archivo y la preservación requieren elementos especiales para trazar la evolución de un objeto digital (de dónde viene y cómo ha cambiado a lo largo del tiempo), detallar sus características físicas, y documentar su comportamiento para emularlo en futuras tecnologías.

4.4. Los diez mandamientos de los metadatos

1. La creación de metadatos de calidad debe ser un aspecto fundamental en la gestión de una colección digital, tan importante como la presentación y difusión de los contenidos. Por ello se le deben dedicar los recursos necesarios.
2. La creación de metadatos es un proceso incremental y debe tener una responsabilidad compartida. Un registro de metadatos debe nacer como un contenedor al que se añadirán los elementos imprescindibles y posteriormente se irá enriqueciendo a medida que pase por los distintos estados de su vida. Incluso se puede pensar en la posibilidad de que sea el propio usuario el que también añada metadatos.
3. Se deben establecer procedimientos detallados en cada unidad responsable de la asignación de metadatos para evitar datos de mala calidad, ineficiencias etc.
4. La implantación de una estrategia de metadatos coherente pasa por la formación de un grupo de expertos en distintas áreas: catalogadores,

técnicos en las áreas que se están describiendo (topógrafos para materiales tipo mapas, o fotografías para digitalización, etc.)

5. Se debe insertar la generación de metadatos en los sistemas informáticos de gestión de la institución de tal forma que se evite la redundancia de los datos, la falta de sincronización y se asegure la interoperabilidad para la transferencia y validación automatizada de los mismos.
6. No hay una solución única para todos los problemas. No existe el esquema de metadatos que pueda responder a todas nuestras necesidades por lo que se hace necesario elegir y combinar elementos en un perfil de aplicación propio.
7. Se hace necesario automatizar la generación de metadatos siempre que sea posible y apropiado.
8. La asignación de metadatos debe hacerse pensando en la interconexión de sistemas. Los metadatos no se van a usar solamente en nuestra aplicación local sino que serán distribuidos a través de protocolos como OAI-PMH a multitud de agregadores.
9. La documentación sobre metadatos debe ser accesible a todos los miembros de la organización.
10. La importancia de disponer de una política de metadatos debe ser entendida por los gestores de la institución.

4.5. Principios en la asignación de metadatos

- **Modularidad.** Es clave en aquellos entornos caracterizados por la diversidad de las fuentes y estilos de gestión de contenidos o por diferentes políticas de descripción de los mismos. Permite crear nuevos esquemas basados en otros ya existentes y beneficiarse de las buenas prácticas existentes más que reinventar elementos desde cero. Es lo que sucede cuando se combina un esquema destinado a la descripción con otro destinado a la conservación para generar uno nuevo expresado en XML. De esta forma se trabajaría con módulos que pueden ser mezclados para abordar los requisitos específicos de una aplicación dada sin sacrificar la interoperabilidad entre diferentes áreas.
- **Extensibilidad.** Los sistemas de metadatos deben permitir la extensibilidad para que se pueda dar cabida a las particularidades de una aplicación concreta. Algunos elementos aparecen en la mayoría de los esquemas de metadatos, por ejemplo, el concepto de creador o de identificador de un determinado objeto. Otros serán específicos de unas determinadas áreas, por ejemplo, la latitud y longitud en un sistema de geolocalización o la profundidad de bits en la descripción de una

fotografía. Una arquitectura de metadatos debe acomodar fácilmente las necesidades locales sin comprometer la interoperabilidad, de tal forma que una aplicación externa que se encuentre con datos locales sea capaz de ignorarlos al tiempo que sigue entendiendo los elementos comunes.

- **Refinamiento.** Los perfiles de aplicación diferirán en función del grado de detalle exigido por cada uno. La asignación de metadatos es costosa por lo que se deben asignar los metadatos necesarios, pero no más. Existen dos formas de refinamiento, la primera es la adición de calificadores para hacer más específico el significado de un elemento. Por ejemplo añadir ilustrador, compositor o autor a un elemento más general que puede ser el creador. Fecha de publicación, fecha de modificación o fecha de acceso son términos más específicos de un elemento general denominado fecha. Una segunda variedad implica la especificación de esquemas particulares o valores que definen los rangos de contenidos aceptados para un elemento dado. Decir que un valor ha sido extraído de un vocabulario controlado o construido de acuerdo a un algoritmo particular puede hacerlo mucho más útil, especialmente de cara al procesamiento automático del mismo.
- **Multilingüismo.** Es esencial adoptar arquitecturas de metadatos que respeten la diversidad cultural y lingüística. Ello implica que los recursos se puedan presentar a los usuarios en su lengua materna, con los caracteres apropiados, etc.

4.6. Estructurando los metadatos

Los esquemas de metadatos son conjuntos de elementos de metadatos diseñados para un propósito específico tal como describir un particular tipo de recurso de información.

La definición o significado de los elementos se conoce como la semántica de un esquema. Los valores dados a los elementos, son los contenidos. Los esquemas, en general, especifican los nombres de los elementos y su significado. Opcionalmente, pueden especificar reglas sobre cómo se debe formular el contenido (por ejemplo, cómo identificar el autor principal) o cuales son los contenidos permitidos (por ejemplo, si los términos deben ser sacados de un vocabulario controlado determinado).

Puede haber también reglas de sintaxis sobre cómo deben ser codificados tanto los elementos como los contenidos. Un esquema de metadatos sin reglas sintácticas definidas se denomina de sintaxis independiente. Los metadatos se pueden condicionar en cualquier sintaxis definible, pero la mayoría de los actuales esquemas utilizan XML.

Es importante tener en cuenta una serie de conceptos prácticos:

- Esquema de metadatos. Plan lógico que muestra las relaciones entre los distintos elementos del conjunto de metadatos, normalmente mediante el establecimiento de reglas para su uso y gestión y específicamente relacionados con la semántica, la sintaxis y la obligatoriedad de los valores. También se refiere a un conjunto de elementos.
- Perfiles de aplicación. Ya hemos mencionado que no existe el esquema de metadatos perfecto por lo que se hace necesario tomar lo mejor de varios de ellos. Un perfil de aplicación define el uso de los elementos de metadatos incluidos en un conjunto de elementos. Mientras que un conjunto de elementos establece conceptos, expresados por los propios elementos de metadatos, y se enfoca sobre la semántica o los significados de aquellos elementos, un perfil de aplicación va más lejos y añade las reglas de la organización y las directrices en el uso de los elementos. Identifica las obligaciones y limitaciones de los elementos, y proporciona comentarios y ejemplos para ayudar a la comprensión de los elementos. Los perfiles de aplicación pueden incluir elementos integrados procedentes de uno o más conjuntos de elementos permitiendo de este modo a una aplicación determinada cumplir sus requisitos funcionales.
- Esquema de codificación. Lista controlada de todos los valores aceptados en lenguaje natural o cadena de caracteres formateada con una sintaxis concreta, diseñados para su procesamiento automatizado. Incluye reglas/formatos para la entrada de datos tales como fechas, nombres de personas, etc.
- Tabla de equivalencias. Especificación para mapear una norma de metadatos a otra. Las tablas de equivalencias también pueden establecerse entre esquemas y perfiles de aplicación.
- Sintaxis y semántica. Semántica se refiere al significado, sintaxis se refiere a la forma. Para intercambiar metadatos entre dos comunidades es necesario que exista un acuerdo sobre ambos. Por ejemplo deben ponerse de acuerdo sobre el significado del elemento Título o Creador y cómo se van a codificar y expresar dichos elementos. Junto con estos dos aspectos es importante también el uso de vocabularios controlados o espacios de nombres para acotar el alcance e incrementar la precisión de una descripción.
- Identificar y nombrar elementos. Es necesario utilizar estándares tales como el URI (Uniform Resource Identifier) para que cada elemento pueda ser identificado con un código entendible a nivel internacional de tal forma que se facilite el proceso de metadatos en diferentes lenguas y aplicaciones.

- Elementos opcionales frente a obligatorios. Los esquemas de metadatos deben ser flexibles puesto que los elementos que son esenciales en un área pueden ser innecesarios en otra. Por ello, pocos o ningún elemento deben de hacerse obligatorios en un esquema de carácter general. Esta situación cambia cuando lo que se está definiendo es un perfil de aplicación.

4.7. Calidad de los metadatos

Frente al mundo del MARC y el catálogo tradicional donde existen múltiples herramientas para controlar la calidad de los datos, en cuanto a valores predefinidos, sintaxis correctas, etc. el emergente mundo de los metadatos para documentos digitales se asemeja al lejano oeste donde la normalización brilla por su ausencia. En lugar de personal bibliotecario que aplica unos estándares bien documentados a un número relativamente pequeño de recursos, lo que nos encontramos es con personas con poca formación, trabajando de forma aislada (sin la adecuada documentación) para describir un creciente y complejo número de recursos. Para complicar la cosa tenemos registros creados de forma automática o creados a partir de la integración de registros diversos.

Con esta situación no es sorprendente que los registros puestos a disposición de agregadores como Recolecta¹ tengan una variedad tal que hagan muy difícil la integración más allá de la mera búsqueda y casi imposible la prestación de servicios de valor añadido sobre los mismos.

Por lo tanto, sin un mínimo nivel de calidad en los metadatos, la creación de servicios sobre esos datos es, como poco, compleja.

Los problemas que habitualmente nos encontramos al trabajar con metadatos son:

- Datos incompletos. En algunos casos se puede decidir dejar sin especificar aspectos que se consideran obvios o sin utilidad para la institución que crea los metadatos (por ejemplo porque el contenido de un elemento sería el mismo en todos los registros); o porque los metadatos proceden de aplicaciones ya existentes en las que simplemente dichos elementos no estaban definidos. Posiblemente la institución podrá trabajar sin problemas, pero cuando esos metadatos salen e intentan ser integrados en agregadores temáticos es cuando surgen los problemas al perderse el contexto original. Un ejemplo claro es el uso de los elementos "type" y "format" del Dublin Core.

¹<http://recolecta.net>

- Datos incorrectos, por ejemplo por la utilización errónea de elementos tales como la inclusión en un solo elemento contenidos que deberían repetirse en varios.
- Datos confusos. Por ejemplo el uso inconsistente de cadenas de caracteres (palabras clave separadas unas veces por ";" y otras por ","), inclusión de caracteres erróneos o mal codificados (utf8 incorrecto en XML), o entidades codificadas doblemente en XML, etc.
- Datos insuficientes.

Capítulo 5

Algunos esquemas de metadatos

5.1. Esquemas y perfiles de aplicación de metadatos

Entendemos por esquema de metadatos aquel conjunto de elementos de metadatos diseñado para un propósito específico, tal como describir un tipo particular de recurso de información. Generalmente especifican un diccionario de nombres de elementos y su semántica. Opcionalmente pueden especificar también reglas aplicables a cómo debe ser formulado el contenido, su representación (uso de mayúsculas/minúsculas), valores permitidos o no, etc.

Se están desarrollando múltiples esquemas de metadatos dirigidos a una gran variedad de entornos y disciplinas: Dublin Core, MODS, MIX, MPEG-7, etc.

Es importante diferenciar entre esquemas de metadatos y perfiles de aplicación de metadatos. Es muy difícil que un esquema cumpla con todos los requisitos necesarios en un determinado ámbito de aplicación. Esto lleva a la necesidad de crear ampliaciones locales a esquemas generales o bien a definir perfiles de aplicación.

Un perfil combina elementos de dos o más esquemas para responder a las necesidades de una aplicación local. Un perfil no puede crear elementos nuevos sino simplemente unir elementos ya existentes. Puede también refinar las definiciones de los elementos pero solamente haciéndola más específica o concreta.

5.2. Esquemas de metadatos descriptivos

Dublin Core

Dublin Core es el esquema más ampliamente utilizado por muchas razones entre las cuales la simplicidad es quizás la principal. Se trata de un conjunto de 15 elementos básicos diseñados para la descripción y descubrimiento de cualquier tipo de objeto digital. Sobre él se han construido muchas bibliotecas digitales y además está presente como parte integrante de otros estándares como el Open Archives Initiative - Protocol for Metadata Harvesting (OAI-PMH).

Los principios que guiaron la selección de estos elementos fueron: la simplicidad, la interoperabilidad semántica (Title = title = título), el consenso internacional, su aplicación a cualquier dominio y formato, la posibilidad de ser ampliado y el carácter modular.

Title	Format
Creator	Identifier
Subject	Source
Description	Language
Publisher	Relation
Contributor	Coverage
Date	Rights
Type	

Más información en:

<http://dublincore.org/documents/dces/>

Si estos quince elementos se consideran demasiado amplios para una aplicación determinada, pueden ser calificados para precisar el alcance de cada uno de ellos. Ahora bien, mientras esta técnica aumenta la precisión, dificulta la interoperabilidad. Esta ampliación es lo que se denomina Dublin Core Calificado o DCMI Metadata Terms.

Más información en:

<http://dublincore.org/documents/dcmi-terms/>

✓ Ejemplo registro DC

```
<dcterms:title>Godiva Chocolatier</dcterms:title>
<dcterms:alternative>Godiva store</dcterms:alternative>
<dcterms:creator>Confection, Nancy</dcterms:creator>
<dcterms:subject>Chocolate</dcterms:subject>
<dcterms:subject xsi:type=''dcterms:lchsh''>Truffles (Confectionery)
</dcterms:subject>
```

✓ **Ejemplo registro DC (cont.)**

```
<dcterms:created>2008--6--28</dcterms:created>
<dcterms:identifier>http://www.godiva.com</dcterms:identifier>
<dcterms:abstract>Provides access to chocolate collections, gifts, ...
</dcterms:abstract>
```

MODS

MODS (Metadata Object Description Schema) ofrece una alternativa a las deficiencias de Dublin Core. Ofrece un diccionario mucho más rico, de aproximadamente ochenta elementos, que permite un mayor grado de precisión pero retiene la interoperabilidad dado que dichos elementos son fijos y se utilizan sin calificadores. Aunque está diseñado específicamente para objetos digitales, está basado en un subconjunto de la norma MARC y por lo tanto se integra bien con los metadatos almacenados en los catálogos tradicionales de las bibliotecas.

A diferencia del Dublin Core que presenta una estructura plana, MODS aprovecha el carácter jerárquico de XML para definir elementos, elementos hijos o subelementos y atributos de elementos. El contenido se incluye siempre en los elementos de nivel más bajo, introduciendo el concepto de contenedor. Así si el elemento <titleInfo>contiene subelementos para <title>, <partName>y <partNumber>, entonces se dice que es un contenedor de elementos más específicos. Un contenedor no puede tener datos sino solamente subelementos.

Los atributos, como en cualquier documento XML, pueden estar asociados a cualquier elemento.

Un documento MODS contiene una declaración de espacio de nombres: <mods xmlns:mods="http://www.loc.gov/mods/v3">

✓ **Ejemplo elemento MODS y su presentación ISBD**

```
<originInfo>
  <place>
    <placeTerm type="code" authority="marccountry">nyu</placeTerm>
    <placeTerm type="text">Ithaca, NY</placeTerm>
  </place>
  <publisher>Cornell University Press</publisher>
  <copyrightDate>1999</copyrightDate>
</originInfo>
```

✓ **Ejemplo elemento MODS y su presentación ISBD (cont.)**

Ithaca, NY : Cornell University Press, c1999.

Más información en:

<http://www.loc.gov/standards/mods/userguide/index.html>

Learning Object Metadata: LOM

Se trata de un estándar abierto de la IEEE (IEEE 1484.12.1 2002 Standard for Learning Object Metadata) para la descripción de objetos de aprendizaje. Se define como tal cualquier entidad, digital o no, que pueda ser usado con propósitos educativos.

Igual que MODS, LOM está compuesto de una jerarquía de elementos. En un primer nivel existen nueve categorías, cada una de las cuales contiene subelementos. Estos pueden ser simples, es decir, almacenan datos directamente, o pueden ser contenedores si almacenan otros elementos. La semántica de un elemento viene determinada por su contexto, por el elemento padre o contenedor en la jerarquía y por otros elementos en el mismo contenedor. Por ejemplo, existen hasta seis elementos denominados Description, el significado de cada uno de ellos vendrá marcado por el elemento padre en el que se incluye cada uno de ellos: General.Description, Rights.Description, Classification.Description, etc.

Las nueve categorías principales son:

General	Educational
Life Cycle	Rights
Meta-Metadata	Relation
Technical	Annotation
Classification	

Más información en:

http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

✓ **Ejemplo descripción LOM**

```
<lom xsi:schemaLocation=''http://www.imsglobal.org/xsd/imsmd.xsd''>
<general>
  <title>
    <langstring xml:lang=''en''>Draft Standard for Learning
    Object Metadata</langstring>
```

✓ **Ejemplo descripción LOM (cont.)**

```
<langstring xml:lang='nl'>Voorstel van Standaard voor Metadata
van Leerobjecten</langstring>
</title>
<educational>
<interactivitytype>
<source>
<langstring xml:lang='x-none'>LOMv1.0</langstring>
</source>
<value>
<langstring xml:lang='x-none'>Expositive</langstring>
</value>
</interactivitytype>
</lom>
```

Ejemplo completo en:

http://www.imsglobal.org/metadata/imsmdv1p2p1/test_schema.xml

5.3. Esquemas de metadatos técnicos

NISO Metadata for Images in XML: MIX

Este esquema permite la codificación en XML de los elementos definidos en la norma Data Dictionary - Technical Metadata for Digital Still Images (ANSI/NISO Z39.87-2006).

El rango de información que puede ser codificado en MIX es muy grande, desde datos básicos sobre tipos de ficheros y tamaños, hasta detalles de la captura de la imagen o detalles sobre cómo se ha tratado la imagen después de la captura.

Aunque un registro MIX puede ser muy grande y complejo, casi todos sus elementos son opcionales con lo cual un registro básico puede ser muy simple.

Utiliza la misma estructura jerárquica que hemos visto en MODS y LOM, con datos en el nivel más bajo de la estructura e incluyendo el concepto de contenedores.

Más información en:

<http://www.loc.gov/standards/mix/>

✓ **Ejemplo descripción LOM**

```
<mix xsi:schemaLocation=''http://www.loc.gov/standards/mix/mix10/mix10.xsd''>
<BasicDigitalObjectInformation>
  <fileSize>33</fileSize>
</BasicDigitalObjectInformation>
<BasicImageInformation>
  <BasicImageCharacteristics>
    <imageWidth>400</imageWidth>
    <imageHeight>200</imageHeight>
  </BasicImageCharacteristics>
</BasicImageInformation>

</mix>
```

5.4. Esquemas de metadatos de preservación

PREservation Metadata Implementation Strategies: PREMIS

La preservación efectiva a largo plazo de contenidos digitales requiere metadatos específicos además de los ya establecidos en los puntos anteriores. El tipo de datos que es necesario registrar incluyen detalles sobre la procedencia y propiedad, integridad de la información, registro de todos los cambios y acciones llevadas a cabo sobre el objeto así como cualquier información técnica o sobre derechos que sea necesaria para suministrarla al usuario. Claramente habrá un cierto grado de solapamiento, en especial con los metadatos técnicos con otros esquemas.

PREMIS es esencialmente un diccionario, un conjunto de elementos, a partir del cual se han derivado diferentes esquemas de representación en XML. Estos elementos cubren:

- El objeto mismo (object), incluyendo identificadores, integridad, información sobre su creación y su relación con otros objetos.
- Eventos (events) asociados con él, tales como su creación y cómo y cuándo ha sido modificado a posteriori.
- Agentes (agents) asociados con su preservación (personas, organizaciones, software, etc)
- Derechos (rights) asociados con el objeto y los agentes.

✓ **Ejemplo descripción PREMIS**

```
<premis xsi:schemaLocation='' http://www.loc.gov/standards/premis/v2/premis-v2-0.xsd'' >
<premis:preservationLevel>
  <premis:preservationLevelValue>full</premis:preservationLevelValue>
  <premis:preservationLevelDateAssigned>20070529
    </premis:preservationLevelDateAssigned>
</premis:preservationLevel>

<premis:creatingApplication>
  <premis:creatingApplicationName>Adobe Photoshop
    </premis:creatingApplicationName>
  <premis:creatingApplicationVersion>CS2
    </premis:creatingApplicationVersion>
  <premis:dateCreatedByApplication>1998-10-30T08:29:02
    </premis:dateCreatedByApplication>
</premis:creatingApplication>
```

Más información en:

<http://www.loc.gov/standards/premis/v2/premis-2-0.pdf>

5.5. Dando unidad, el esquema METS

Metadata Encoding and Transmission Standard (METS) proporciona una estructura jerárquica para codificar metadatos sobre la estructura interna de un ítem. Esto es simplemente una serie de elementos enlazados denominados div (por división) cuyo resultado emula la estructura de un objeto digital. Por ejemplo, un libro digitalizado tendrá su estructura de divs organizada para emular su división original en capítulos, secciones, etc.

Un documento METS consiste de cinco secciones principales:

- Metadatos descriptivos. Puede simplemente apuntar a una descripción externa al propio documento (por ejemplo a un registro MARC en un OPAC) o contener embebido los datos o una mezcla de ambos.
- Metadatos administrativos. Proporciona información sobre cómo se crearon los ficheros, propiedad intelectual, metadatos técnicos, etc. Se puede incluir una o varias entradas de metadatos administrativos por cada documento.
- Grupos de ficheros. Lista todos los ficheros que componen el objeto digital.

- Mapa estructural. Es el corazón de un documento METS. Se encarga de diseñar una estructura jerárquica para el objeto y enlazar los elementos de esta estructura con ficheros y metadatos que pertenecen a dichos ficheros.
- Comportamiento. Se puede utilizar para asociar determinados comportamientos con los contenidos del fichero mets. Por ejemplo ejecutar alguna aplicación sobre el contenido de un elemento determinado.

✓ Metadatos descriptivos

```
<dmdSec ID="dmd002">
  <mdWrap MIMETYPE="text/xml" MDTYPE="DC" LABEL="Dublin Core">
    <dc:title>Tityre tu patulae recubans sub tegmine fagi </dc:title>
    <dc:creator>Virgili Maro', Publi, 70-19 aC.</dc:creator>
    <dc:title.alternative>Opera</dc:title.alternative>
    <dc.date.created>[ca. 1450] ;</dc.date.created>
  </mdWrap>
</dmdSec>
```

✓ Metadatos técnicos

```
<amdSec ID="AMD001">
  <mdWrap MIMETYPE="text/xml" MDTYPE="MIX" LABEL="MIX">
    <mix:CameraCaptureSettings>
      <mix:ImageData>
        <mix:fNumber>14</mix:fNumber>
        <mix:exposureTime>1/3.3 seconds</mix:exposureTime>
        <mix:exposureProgram>3</mix:exposureProgram>
        <mix:isoSpeedRatings>100</mix:isoSpeedRatings>
      </mix:ImageData>
    </mix:CameraCaptureSettings>
  </mdWrap>
</amdSec>
```

✓ Ficheros

```
<fileSec>
  <fileGrp ID="filegrp1" USE="Large images">
    <file ID="file11" MIMETYPE="image/jpeg" SEQ="1">
      <FLocat LOCTYPE="URL">
        http://weblioteca.uv.es/europeana/ms/0768/uv_ms_0768_11.jpg
      </FLocat>
    </file>
  </fileGrp>
</fileSec>
```

✓ **Ficheros (cont.)**

```
</file>
<file ID="file12" MIMETYPE="image/jpg" SEQ="2">
  <FLocat LOCTYPE="URL">
    http://weblioteca.uv.es/europeana/ms/0768/uv_ms_0768_2l.jpg
  </file>
</fileGrp>
<fileGrp ID="filegrp2" USE="Small images">
  <file ID="files1" MIMETYPE="image/jpg" SEQ="1">
    <FLocat LOCTYPE="URL">
      http://weblioteca.uv.es/europeana/ms/0768/uv_ms_0768_1s.jpg
    </file>
  <file ID="files2" MIMETYPE="image/jpg" SEQ="2">
    <FLocat LOCTYPE="URL">
      http://weblioteca.uv.es/europeana/ms/0768/uv_ms_0768_2s.jpg
    </file>
  </fileGrp>
</fileSec>
```

✓ **Mapa estructural**

```
<structMap TYPE="physical">
  <div ORDER="0" TYPE="book" LABEL="Tityre tu patulae ..."
    DMDID="dmd002">
    <div ORDER="1" TYPE="page" LABEL="Enq.">
      <fptr FILEID="file11"/>
      <fptr FILEID="files1"/>
    </div>
    <div ORDER="1" TYPE="page" LABEL="FG 1">
      <fptr FILEID="file12"/>
      <fptr FILEID="files2"/>
    </div>
  </div>
</structMap>
```



Ejercicio 5

Elaborar un fichero METS para dar unidad al objeto digital complejo que se ha descargado.

5.6. Perfiles de aplicación de metadatos

Los perfiles de aplicación de metadatos se caracterizan por:

- Pueden basarse en uno o varios esquemas existentes y tomar elementos de varios de ellos, pero no puede crear elementos nuevos.
- Si fuera necesario crear elementos nuevos sería necesario crear un esquema nuevo y asegurarse el mantenimiento del mismo.
- Se pueden especificar valores específicos para un elemento particular. Por ejemplo se puede especificar un vocabulario controlado para algún elemento o formatos concretos para fechas, etc.

Un perfil de aplicación debe indicar los esquemas en los que se basa y una relación de los elementos que utiliza con la descripción de los mismos.

La descripción puede ser más o menos detallada pero es recomendable utilizar al menos los siguientes elementos:

- Identificador del elemento. El nombre dado al elemento tal y como va a figurar en la base de datos
- Etiqueta. Un nombre asociado al identificador destinado a ser interpretado por una persona.
- Definición. Especifica el tipo de información requerida por el elemento que se está describiendo. En la mayor parte de los casos se tratará de la misma definición que se aporta en el esquema del que se toma el elemento.
- Obligación. Indica si el elemento debe ser incluido o puede ser omitido. En general una obligación se puede especificar con alguno de los siguientes tres valores: Obligatorio (siempre debe ser incluido aunque haya que crear valores específicos en caso de no existir la propiedad), Requerido si existe (se debe incluir si la propiedad existe) y Opcional (no es obligatorio).
- Repetible. Indica si se debe usar un valor único o múltiple para el elemento.
- Tipo de datos. Establece listas de términos o códigos de clasificación de la que se van a extraer los valores del elemento
- Mapeo al esquema correspondiente.

Ejemplos de perfiles de aplicación

Capítulo 6

Uso de metadatos en las colecciones del SBD