



# ESTANDARES DE METADATOS

Red Mexicana  
de Repositorios  
Institucionales

# DUBLIN CORE

- **Dublin Core** es un modelo de metadatos elaborado por la [DCMI](#) (Dublin Core Metadata Initiative), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos y a promover el desarrollo de los vocabularios especializados de metadatos para describir recursos para permitir sistemas más inteligentes del descubrimiento del recurso.
- Las implementaciones de Dublin Core usan generalmente **XML** y se basan en el Resource **Description Framework**. Dublin Core se define por ISO en su norma ISO 15836 del año 2003, y la norma [NISO Z39.85-2007](#).
- El conjunto de elementos Dublin Core se centró en **13 elementos**, pero concluyó con 15 descriptores como resultado de un consenso y un esfuerzo interdisciplinar e internacional. Ya existen transcripciones a 20 idiomas y ha sido adoptado por el CEN/ISS y posee dos RFCs de Internet ([RFC2413](#)) y ([RFC2731](#)). Es también el estándar oficial del [WWW Consortium](#) y el estándar del Z39.50. Los metadatos Dublin Core han sido aprobados por el organismo nacional de estandarización norteamericano (ANSI/NISO Z39.85) y los utilizan como base, tanto gobiernos, como agencias supranacionales y muchas otras iniciativas de metadatos pertenecientes a comunidades específicas como bibliotecas, archivos, en educación, negocios, etc.

# DUBLIN CORE

- En concreto, la Iniciativa pretende:
- Desarrollar estándares de metadatos para la recuperación de información en Internet a través de distintos dominios.
- Definir el marco para la interoperabilidad entre conjuntos de metadatos.
- Facilitar el desarrollo de conjuntos de metadatos específicos de una disciplina o comunidad que trabaja dentro del marco de la recuperación de información.

# DUBLIN CORE

## Elementos del Contenido

-  **Título (Title).** Se refiere al título que lleva por nombre el documento.
-  **Materia (Subject).** En este campo se hace referencia a los diversos temas que puede contener el material.
-  **Descripción (Description).** En este campo se hace un breve resumen sobre el contenido del objeto digital.
-  **Fuente (Source).** Es como una pequeña ficha bibliográfica que se elabora para asentar los datos sobre la procedencia del documento original.
-  **Lenguaje (Language).** En este campo se establecen las siglas correspondientes al idioma en que se presenta la publicación.
-  **Relación (Relation).** Este campo tiene que ver con el material principal u objetos de su misma referencia, ya sea una colección, una serie, un documento, etc.
-  **Cobertura (Coverage).** Este campo se refiere al proyecto o sitio donde estará resguardada la información. Aquí pueden anotarse fechas, zonas geográficas.

# DUBLIN CORE

## Elementos de Propiedad Intelectual

-  **Autor (Creator).** Aquí se anota el autor intelectual de la obra o documento original.
-  **Editor (Publisher).** Este campo se refiere al sitio o colección responsable, a la que está adscrito el material.
-  **Colaborador (Contributor).** En este campo se anotan, si es que se da el caso, el nombre u organización que contribuyó a la creación del material, que no se especificó en la parte de Autor.
-  **Derechos (Rights).** Se anota en este campo el nombre o la institución a la cual pertenece el material y lo facilitó.
-  **Elementos de Aplicación**
-  **Fecha (Date).** Se anota la fecha de elaboración del registro.
-  **Tipo (Type).** Aquí se menciona la presentación que tiene el objeto digital, ya sea como texto, audio, video, etcétera.
-  **Formato (Format).** En este campo se registra el tipo de extensión con que se presenta el objeto digital, ya sea HTML, JPG, GIFF o PDF.
-  **Identificador (Identifier).** Se refiere a la dirección electrónica de origen a la que está adscrito el material. Para ello se utilizan las siglas URL.

# DUBLIN CORE

	Elemento	Etiqueta	Definición	Comentario
1	Título	DC:Title	Nombre del recurso	Nombre del recurso por el que formalmente se conoce.
2	Creador	DC:Creator	Entidad responsable del contenido del recurso	El creador puede ser una persona, una organización o un servicio. El nombre del creador se usará para indicar la entidad.
3	Materia	DC:Subject	Contenido temático del recurso	Materia representada como palabra clave, frase o codificación que represente el contenido del recurso. Se recomienda el uso de un lenguaje documental.
4	Descripción	DC:Description	Resumen del contenido del documento	Normalmente incluye el resumen, sumario, etc.
5	Editor	DC:Publisher	Entidad responsable de la creación del recurso	El editor incluye a personas, organizaciones o servicios.
6	Contribuidor	DC:Contributor	Responsables del desarrollo de los contenidos del recurso	El contribuidor puede ser una persona, entidad o servicio.
7	Fecha	DC:Date	Fecha asociada al ciclo de vida del recurso	Se asocia a la creación o disponibilidad del recurso.
8	Tipo de recurso	DC:Type	Naturaleza del tipo de recurso	Incluye las categorías, funciones, géneros y niveles de agregación al recurso.
9	Formato	DC:Format	Manifestación física o digital del recurso	Incluye los medios según la tipología y dimensiones del recurso. Puede incluir el software, Hardware y medios que se necesitan para poder visualizar y trabajar con el recurso.
10	Identificador	DC:Identifier	Identificación unívoca del recurso	Secuencia de caracteres que identifique de forma unívoca el recurso: DOI, URI...
11	Fuente	DC:Source	Identifica la fuente del que proviene el recurso actual	La fuente de la que ha derivado el recurso de la forma un todo.
12	Lengua	DC:Language	Lengua del contenido intelectual del recurso	Se recomienda el uso de la norma RFC 3066, que en combinación con 639, define la codificación en dos o tres caracteres las diferentes lenguas.
13	Relación	DC:Relation	Referencia a uno a varios recursos relacionados con el actual	Es la mejor práctica para identificar los recursos referenciados.
14	Cobertura	DC:Coverage	Extensión o alcance de los contenidos del recurso	Localización espacial, periodo temporal o jurisdicción.
15	Derechos de autor	DC:Rights	Información sobre los derechos de autor que afectan al recurso	Contiene información sobre los estamentos que gestionan los derechos de autor o referencias a servicios que informan sobre los derechos de autor.

# LOM (Learning Object Metadata)

## RECURSOS EDUCATIVOS DIGITALES REUTILIZABLES

- Learning Object Metadata es un modelo de datos, usualmente codificado en **XML**, usado para describir un objeto de aprendizaje y otros recursos digitales similares usados para el apoyo al aprendizaje. Su propósito es ayudar a la reutilización de objetos de aprendizaje y facilitar su interaccionalidad, usualmente en el contexto de sistemas de aprendizaje on-line: (online learning management systems (LMS)).
- El estándar **IEEE 1484.12.1:2002** sobre metadatos para objetos de aprendizaje es un estándar abierto internacionalmente reconocido (publicado por el Instituto de Ingenieros Eléctricos y Electrónicos) para la descripción de los "objetos de aprendizaje". Las cualidades relevantes de los objetos de aprendizaje que se describen incluyen: título, idioma, tipo de objeto, autor, propietario, términos de distribución, formato, copyright, y cualidades pedagógicas, tales como estilo de la enseñanza o de la interacción.
- Un LOM se define como una **entidad**, digital o no digital que puede ser usada, reutilizada o referenciada durante cualquier actividad de aprendizaje basada en la tecnología.
- Los estándares del LOM se centran en el conjunto mínimo de propiedades que permiten que los objetos educacionales sean gestionados, ubicados y evaluados.

# LOM (Learning Object Metadata)

## Definición de las categorías

- **General:** Identificador, Título, Entrada de catálogo, Lengua, Descripción, Descriptor, Cobertura, Estructura, Nivel de agregación
- **Ciclo de vida:** Versión, Estatus, Otros colaboradores
- **Meta-metainformación:** Identificador, Entrada de catálogo, Otros colaboradores, Esquema de metadatos, Lengua
- **Técnica:** Formato, Tamaño, Ubicación, Requisitos, Comentarios sobre la instalación, Otros requisitos para plataformas, Duración
- **Uso educativo:** Tipo de interactividad, Tipo de recurso de aprendizaje, Nivel de interactividad, Densidad semántica, Usuario principal, Contexto [Nivel educativo], Edad, Dificultad, Tiempo previsto de aprendizaje, Descripción, Lengua
- **Derechos:** Coste, Copyright y otras restricciones, Descripción
- **Relación [con otros recursos]:** Tipo [naturaleza de la relación con el recurso principal], Recurso [recurso principal al que se refiere esta relación]
- **Observaciones:** Persona, Fecha, Descripción
- **Clasificación:** Finalidad, Nivel taxón (taxonómico), Descripción, Descriptor.

# TEI: Text Encoding Initiative

## Recursos bibliográficos

- Las Normas del Text Encoding Initiative (TEI) están dirigidas a cualquier persona que quiera intercambiar información almacenada en un formato electrónico. En ellas se enfatiza el intercambio de información textual, pero también se hace referencia a otras formas de información como son las imágenes y el sonido. Las Normas son del mismo modo aplicables a la creación de nuevos recursos y al intercambio de los ya existentes.
- 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 en 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.

# TEI: Text Encoding Initiative

## Recursos bibliográficos

- Todos los textos que cumplan el TEI contienen (a) un *encabezado TEI* (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 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* (una única obra) o *compuesto* (una colección 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: Text Encoding Initiative



<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>

<TEI.2>

```
<teiHeader> [ información del encabezado del conjunto de textos ] </teiHeader>
<text>
  <front> [ materia del front del conjunto de textos ] </front>
  <group>
    <text>
      <front> [ materia del front del primer texto ] </front>
      <body> [ materia del body del primer texto ] </body>
      <back> [ materia del back del primer texto ] </back>
    </text>
    <text>
      <front> [ materia del front del segundo texto ] </front>
      <body> [ materia del body del segundo texto ] </body>
      <back> [ materia del back del segundo texto ] </back>
    </text>
    [ aquí van los demás textos o grupos de textos ]
  </group>
  <back> [ materia del back del conjunto de textos ] </back>
</text>
</TEI.2>
```

# Metadata Object Description Schema

## Recursos bibliográficos

- La Library of Congress' Network Development and MARC Standards Office, con expertos interesados, está desarrollando un esquema para un set de elementos bibliográficos que pueden ser usados para una variedad de propósitos, particularmente para aplicaciones bibliotecarias. Como esquema XML pretende ser capaz de llevar datos seleccionados de registros MARC 21, así como permitir la creación de registros de descripción de fuentes originales. Incluye un subconjunto de campos MARC y etiquetas basadas en lenguaje más que en números, en algunos casos reagrupando elementos del formato bibliográfico MARC 21.
- El esquema MODS se expresa usando el esquema XML del World Wide Web Consortium y es mantenido por la Network Development and MARC Standards Office de la Library of Congress.
- Usos MODS podría ser usado potencialmente como:
  - Un formato específico Z39.50 de próxima generación
  - Un esquema de extensión de METS (Metadata Encoding and Transmission Standard)
  - Para representar metadatos para harvesting (cosecha)
  - Para descripción de recursos originales en sintaxis XML
  - Para metadatos en XML que pueden ser empaquetados con un recurso electrónico
  - Para representar un registro MARC simple en XML

# Metadata Object Description Schema

## Ventajas de MODS

- Intenta complementar otros formatos de metadatos. Para algunas aplicaciones, particularmente aquellas que han usado registros MARC, hay ventajas sobre otros esquemas de metadatos. Algunas de ellas son:
  - El set de elementos es más rico que el Dublin Core
  - El set de elementos es más compatible con los datos de bibliotecas que el ONIX
  - Está más orientado al usuario final que el esquema MARCXML
  - El set de elementos es más simple que el formato MARC

## Características de MODS

- Los elementos presentan generalmente la semántica del MARC.
- Algunos datos han sido reempaquetados; en algunos casos, varios elementos de datos MARC han sido convertidos a uno en MODS.
- No asume el uso de ningún código específico de catalogación
- Varios elementos tienen un atributo ID opcional para facilitar el enlace a nivel de elemento

## Limitaciones

- MODS incluye un subconjunto de datos del MARC 21 Format for Bibliographic Data.
- Como un set de elementos que permite la representación de datos que ya existen en 53 sistemas basados en MARC, tiene el propósito de permitir la conversión de campos centrales (core fields) mientras que se excluyen algunos datos específicos.

# Resource Description Framework (RDF)

## «Metamodelo» de metadatos

- Resource Description Framework (RDF) [Infraestructura para la Descripción de Recursos] es una base para procesar metadatos; proporciona interoperabilidad entre aplicaciones que intercambian información legible por máquina en la Web. RDF destaca por la facilidad para habilitar el procesamiento automatizado de los recursos Web. RDF puede utilizarse en distintas áreas de aplicación; por ejemplo: en *recuperación de recursos* para proporcionar mejores prestaciones a los motores de búsqueda, en *catalogación* para describir el contenido y las relaciones de contenido disponibles en un sitio Web, una página Web, o una biblioteca digital particular, por los *agentes de software inteligentes* para facilitar el intercambio y para compartir conocimiento; en la *calificación de contenido*, en la descripción de *colecciones de páginas* que representan un "documento" lógico individual, para describir los *derechos de propiedad intelectual* de las páginas web, y para expresar las *preferencias de privacidad* de un usuario, así como las *políticas de privacidad* de un sitio Web. RDF junto con las *firmas digitales* será la clave para construir el "Web de confianza" para el comercio electrónico, la cooperación y otras aplicaciones.
- Es un lenguaje de objetivo general para representar la información en la web (un metadato data model).

# Resource Description Framework (RDF)

La sintaxis serializada RDF básica toma la forma:

- [1] **RDF** ::= ['<rdf:RDF>'] description\* ['</rdf:RDF>']
- [2] **description** ::= '<rdf:Description' idAboutAttr? '>' propertyElt\* '</rdf:Description>'
- [3] **idAboutAttr** ::= idAttr | aboutAttr
- [4] **aboutAttr** ::= 'about="' URI-reference ""
- [5] **idAttr** ::= 'ID="' IDsymbol ""
- [6] **propertyElt** ::= '<' propName '>' value '</' propName '>' | '<' propName resourceAttr '/>'
- [7] **propName** ::= QName
- [8] **value** ::= description | string
- [9] **resourceAttr** ::= 'resource="' URI-reference ""
- [10] **QName** ::= [ NSprefix ':' ] name
- [11] **URI-reference** ::= string, interpreted per [\[URI\]](#)
- [12] **IDsymbol** ::= (any legal [XML name symbol](#))
- [13] **name** ::= (any legal XML name symbol)
- [14] **NSprefix** ::= (any legal [XML namespace prefix](#))
- [15] **string** ::= (any XML text, with "<", ">", and "&" escaped)

Ejemplo

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/">
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

# Resource Description Framework (RDF)

## Gramática formal para RDF

- [1] RDF ::= ['<rdf:RDF>'] obj\* ['</rdf:RDF>']
- [2] obj ::= description | container
- [3] description ::= '<rdf:Description' idAboutAttr? bagIdAttr? propAttr\* '/>' | '<rdf:Description' idAboutAttr? bagIdAttr? propAttr\* '>' propertyElt\* '</rdf:Description>' | typedNode
- [4] container ::= sequence | bag | alternative
- [5] idAboutAttr ::= idAttr | aboutAttr | aboutEachAttr
- [6] idAttr ::= ' ID=" IDsymbol '''
- [7] aboutAttr ::= ' about=" URI-reference '''
- [8] aboutEachAttr ::= ' aboutEach=" URI-reference ''' | ' aboutEachPrefix=" string '''
- [9] bagIdAttr ::= ' bagID=" IDsymbol '''
- [10] propAttr ::= typeAttr | propName '=' string ''' (with embedded quotes escaped)
- [11] typeAttr ::= ' type=" URI-reference '''
- [12] propertyElt ::= '<' propName idAttr? '>' value '</' propName '>' | '<' propName idAttr? parseLiteral '>' literal '</' propName '>' | '<' propName idAttr? parseResource '>' propertyElt\* '</' propName '>' | '<' propName idRefAttr? bagIdAttr? propAttr\* '/>'
- [13] typedNode ::= '<' typeName idAboutAttr? bagIdAttr? propAttr\* '/>' | '<' typeName idAboutAttr? bagIdAttr? propAttr\* '>' propertyElt\* '</' typeName '>'
- [14] propName ::= Qname
- [15] typeName ::= Qname
- [16] idRefAttr ::= idAttr | resourceAttr
- [17] value ::= obj | string
- [18] resourceAttr ::= ' resource=" URI-reference '''
- [19] Qname ::= [ NSprefix ':' ] name
- [20] URI-reference ::= string, interpreted per [URI]

# Resource Description Framework (RDF)

## Gramática formal para RDF continuación

- [21] IDsymbol ::= (any legal [XML name symbol](#))
- [22] name ::= (any legal [XML name symbol](#))
- [23] NSprefix ::= (any legal XML namespace prefix)
- [24] string ::= (any XML text, with "<", ">", and "&" escaped)
- [25] sequence ::= '<rdf:Seq' [idAttr?](#) '>' [member\\*](#) '</rdf:Seq>' | '<rdf:Seq' [idAttr?](#) [memberAttr\\*](#) '/>'
- [26] bag ::= '<rdf:Bag' [idAttr?](#) '>' [member\\*](#) '</rdf:Bag>' | '<rdf:Bag' [idAttr?](#) [memberAttr\\*](#) '/>'
- [27] alternative ::= '<rdf:Alt' [idAttr?](#) '>' [member+](#) '</rdf:Alt>' | '<rdf:Alt' [idAttr?](#) [memberAttr?](#) '/>'
- [28] member ::= [referencedItem](#) | [inlineltem](#)
- [29] referencedItem ::= '<rdf:li' [resourceAttr](#) '/>'
- [30] inlineltem ::= '<rdf:li' '>' [value](#) '</rdf:li>' | '<rdf:li' [parseLiteral](#) '>' [literal](#) '</rdf:li>' | '<rdf:li' [parseResource](#) '>' [propertyElt\\*](#) '</rdf:li>'
- [31] memberAttr ::= ' rdf:\_n="' [string](#) '"' (where *n* is an integer)
- [32] parseLiteral ::= ' parseType="Literal"'
- [33] parseResource ::= ' parseType="Resource"'
- [34] literal ::= (any well-formed XML)

# Resource Description Framework (RDF)

- Veamos un ejemplo concreto, extraído de la especificación Primer RDF (<http://www.w3.org/TR/rdf-primer/>) donde se muestran una serie de declaraciones o sentencias: "hay una persona identificada por <http://www.w3.org/People/EM/contact#me>, cuyo nombre es *Eric Miller*, cuya dirección de correo electrónico es **em@w3.org**, y cuyo título es "Dr." que podría representarse como el grafo RDF de la siguiente figura:



Grafo RDF que describe a Eric Miller.

# OWL

- El Lenguaje de Ontologías Web (OWL) está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información en lugar de únicamente representar información para los humanos. OWL facilita un mejor mecanismo de interpretabilidad de contenido Web que los mecanismos admitidos por XML, RDF, y esquema RDF (RDF-S) proporcionando vocabulario adicional junto con una semántica formal. OWL tiene tres sublenguajes, con un nivel de expresividad creciente: OWL Lite, OWL DL, y OWL Full.
- El Lenguaje OWL está descrito por una serie de documentos, cada uno con una finalidad diferente, y orientados a una audiencia diferente. A continuación proporcionamos una breve guía para la navegación a través de este conjunto de documentos:
  - Esta [Vista general de OWL](#) ofrece una introducción simple a OWL, proporcionando un listado de las características del lenguaje con breves descripciones.
  - La [Guía de OWL](#) describe el uso del lenguaje OWL mediante un extenso ejemplo. También incluye un [glosario](#) con la terminología utilizada en estos documentos.
  - La [Referencia de OWL](#) ofrece una descripción sistemática y compacta (aún no ha sido formalmente reconocida) de todas las primitivas de modelado de OWL.
  - El documento [Semántica y sintaxis abstracta de OWL](#) es la definición normativa final y formal del lenguaje.
  - El documento [Casos de prueba del Lenguaje de Ontologías Web](#) contiene un amplio conjunto de casos de prueba para el lenguaje.
  - El documento [Casos de uso y requerimientos de OWL](#) contiene un conjunto de casos de uso para un lenguaje de ontologías Web y recopila un conjunto de requerimientos para OWL.

# OWL

## Características RDF Schema de *OWL Lite*:

- **Class:** Una clase define un grupo de individuos que permanecen juntos porque comparten las mismas propiedades. Por ejemplo, *Juana* y *Pedro* son ambos miembros de la clase *Persona*. Las clases pueden ser organizadas en una jerarquía especial, usando **subClassOf**. Hay una construcción de clase más general llamada **Thing** que es la clase de todos los individuos y una superclase de todas las clases OWL. Hay también una construcción más específica de clase llamada **Nothing** que es la clase que no tiene instancias y una subclases de todas las clases OWL.
- **rdfs:subClassOf:** Las jerarquías de clase deben crearse para hacer una o más declaraciones de que una clase es un subclase de otra clase. Por ejemplo, la clase *Persona* podría ser establecida para ser una subclase de la clase *Mamífero*. Desde aquí un razonador puede deducir que si un individuo es una *Persona*, entonces es también un *Mamífero*.
- **rdf:Property:** Las propiedades pueden usarse para state relaciones entre individuos o desde individuos a valores de datos. Ejemplos de propiedades son **hasChild**, **hasRelative**, **hasSibling**, y **hasAge**. Las tres primeras pueden usarse para relacionar una instancia de una clase *Person* a otra instancia de la clase *Person* (y son entonces ocurrencias de **ObjectProperty**), y el último (**hasAge**) puede usarse para relacionar una instancia de la clase *Persona* a una instancia del tipo de datos *Entero* (y es entonces una ocurrencia de **DatatypeProperty**).

# OWL

Ambos `owl:ObjectProperty` y `owl:DatatypeProperty` son **subclasses** de la clase RDF `rdf:Property`.

- ***rdfs:subPropertyOf***: Las jerarquías de propiedades deben crearse para hacer una o más declaraciones que una propiedad es una subpropiedad de una o más other propiedades. Por ejemplo, **hasSibling** debe ser establecida para ser una subpropiedad de **hasRelative**. Desde aquí un razonador puede deducir que si un individuo está relacionado a otro por la propiedad **hasSibling**, entonces éste está también relacionado a otro por la propiedad **hasRelative**.
- ***rdfs:domain***: Un dominio de una propiedad limita los individuos a los que se aplica la propiedad. Si una propiedad relaciona un individuo a otro individuo, y la propiedad tiene una clase como uno de sus dominios, entonces el individuo debe pertenecer a la clase. Por ejemplo, la propiedad **hasChild** debe ser establecida para tener el dominio de *Mamífero*. Desde aquí un razonador puede deducir que si Pedro **hasChild** Ana, entonces Pedro debe ser un **Mamífero**. Note que ***rdfs:domain*** se denomina una restricción global desde que la restricción está establecida sobre la propiedad y no justamente sobre la propiedad cuando esta está asociada con una clase particular.

# OWL

- **Individual**: Los individuos son instancias de clases, y las propiedades deben usarse para relacionar un individuo con otro. Por ejemplo, un individuo llamado *Juana* debe ser descrito como una instancia de la clase *Persona* y la propiedad **hasEmployer** debe ser usada para relacionar el individuo *Juana* al individuo *Universidad*.
- **rdfs:range**: El rango de una propiedad limita los individuos que la propiedad debe tener como su valor. Si una propiedad relaciona un individuo a otro individuo, y la propiedad tiene una clase como su rango, entonces el otro individuo debe pertenecer a la clase del rango. Por ejemplo, la propiedad **hasChild** debe ser establecida para tener el rango de *Mamífero*. Desde aquí un razonador puede deducir que si *Luisa* está relacionado con *Juana* por la propiedad **hasChild**, (por ejemplo, *Juana* es la hija de *Luisa*), entonces *Juana* es un *Mamífero*. El rango es también una restricción global como lo era el dominio.

# SKOS

## SIMPLE KNOWLEDGE ORGANIZATION SYSTEM

- SKOS (Simple Knowledge Organization System, **Sistema de Organización del Conocimiento Simple**) es un vocabulario RDF para la representación de sistemas de organización del conocimiento (Knowledge Organization Systems, KOSs) semi-formales, tales como tesauros, taxonomías, esquemas de clasificación y listas de encabezamiento de materias. SKOS está basado en RDF (Resource Description Framework) por lo que dichas representaciones pueden ser legibles por máquinas e intercambiarse entre aplicaciones de software, así como publicarse en la World Wide Web.
- SKOS ha sido diseñado para proporcionar un modo de migrar a la **Web Semántica sistemas** de organización del conocimiento ya existentes con un coste bajo. SKOS también proporciona un lenguaje conceptual de modelado muy sencillo e intuitivo para desarrollar y compartir nuevos sistemas de organización. Puede utilizarse por su cuenta, o en combinación con lenguajes más formales, como el Lenguaje de Ontologías Web (**OWL**). SKOS también puede contemplarse como una tecnología de transición que proporciona un nexo de unión entre el formalismo lógico riguroso de los lenguajes de ontologías como OWL y el mundo caótico, informal y débilmente estructurado de las herramientas colaborativas basadas en Web, ejemplificadas por las aplicaciones de etiquetado social.

# SKOS

- El objetivo de **SKOS** no es sustituir vocabularios conceptuales originales en su contexto inicial de uso, sino que puedan implementarse en un espacio compartido, basado en un modelo simplificado, que haga posible su reutilización y una mejor interoperabilidad.

## Modelo

- El modelo de datos SKOS es en realidad una **ontología** definida con OWL Full. Obviamente, al estar basado en RDF, SKOS estructura los datos en forma de tripletas que pueden ser codificadas en cualquier sintaxis válida para **RDF**. SKOS puede ser utilizado conjuntamente con OWL para expresar formalmente estructuras de conocimiento sobre un dominio concreto ya que SKOS no puede realizar esta función al no tratarse de un lenguaje para la representación de conocimiento formal.

# SKOS

- Es posible asignar a los conceptos códigos de clasificación o de identificación dentro de un esquema conceptual determinado. Estas notaciones no están expresadas en lenguaje natural sino en forma de códigos nemotécnicos o similares. Los conceptos también pueden ser documentados con notas de diferente naturaleza como definiciones, notas de alcance o notas de edición entre otras.
- El modelo SKOS contempla el establecimiento de enlaces entre conceptos denominados relaciones semánticas. Estas relaciones pueden ser jerárquicas o asociativas, contemplándose la posibilidad de ampliar la tipología de relaciones. Los conceptos también pueden agruparse en colecciones que a su vez pueden etiquetarse y ordenarse. SKOS se complementa con la posibilidad de que conceptos de diferentes esquemas se pueden mapear entre sí empleando relaciones jerárquicas, asociativas o de equivalencia exacta.

# SKOS

## Vocabulario

### ■ Conceptos y esquemas de conceptos

- skos:Concept
- skos:ConceptScheme
- skos:inScheme
- skos:hasTopConcept

### ■ Etiquetas léxicas

- skos:prefLabel
- skos:altLabel
- skos:hiddenLabel

### ■ Relaciones semánticas

- skos:semanticRelation
- skos:broaderTransitive
- skos:narrowerTransitive
- skos:broader
- skos:narrower
- skos:related

### ■ Notaciones

- skos:notation

## ■ Documentación

- skos:note
- skos:scopeNote
- skos:historyNote
- skos:changeNote
- skos:definition
- skos:editorialNote
- skos:example

## ■ Colecciones de conceptos

- skos:Collection
- skos:OrderedCollection
- skos:member
- skos:memberList

## ■ Propiedades de mapeado

- skos:mappingRelation
- skos:exactMatch
- skos:broadMatch
- skos:narrowerMatch
- skos:relatedMatch



# ONIX (ONline Information Exchange)

## Publicación

- **ONIX for Books** is the international standard for representing and communicating book industry product information in electronic form.
- ONIX for Books was developed and is maintained by EDItEUR, jointly with Book Industry Communication (UK) and the Book Industry Study Group (U.S.), and has user groups in Australia, Belgium, Canada, Finland, France, Germany, Italy, the Netherlands, Norway, Russia, Spain, Sweden and the Republic of Korea.
- In April 2009, EDItEUR [announced the release](#) of a major new version of the ONIX for Books standard: ONIX 3.0. This release of ONIX is the first since 2001 that is not backwards-compatible with its predecessors and, more importantly, provides a means for improved handling of digital products. A revised version (3.0.1) was subsequently released in January 2012.
- ONIX is an acronym for **ON**line **I**nformation **EX**change. ONIX for Books refers to a standard format that publishers can use to distribute electronic information about their books to wholesale, e-tail and retail booksellers, other publishers, and anyone else involved in the sale of books. ONIX enables book information to be communicated between different organizations even if they have different technical infrastructures and business needs. It isn't a database, but provides a standard **XML template** for organizing data storage.

# ONIX (ONline Information Exchange)

- Este estándar ha sido desarrollado y es mantenido por EDItEUR junto a Book Industry Communication (UK) y Book Industry Study Group (US) y con grupos de usuarios de Australia, Canadá, Francia, Alemania y la República de Corea.
- ONIX Nace de la premisa que mientras más información tengan los clientes de un libro, habrá mayor opción de que lo compren. En el formato impreso, la cubierta sobrepuesta del libro contiene mucha información promocional: diseño de la cubierta, reseña, biografía del autor, etc. Toda esta información empuja al lector a comprarlo.
- Ahora, Internet se ha convertido en un lugar popular para comprar libros; sin embargo, en línea no se cuenta con un libro físico para hojear. En su lugar, existe una página web dedicada al libro, la que puede ser diseñada para tener toda la rica información de la cubierta y más, tal como archivos de audio y video relacionados con el libro.
- Sin embargo, obtener todos esos datos desde las editoriales para ser entregados a los vendedores, ha sido un desafío, complicado por el hecho que las más importantes compañías (tal como Ingram, Bowker y Amazon) tienen diferentes formatos para recibir los datos. Esta falta de estandarización ha provocado un consumo de tiempo y ha dificultado a las editoriales, las actividades de exportación de la información de sus libros.
- Por esto, el propósito de ONIX es estandarizar la transmisión de la información de sus productos, de forma que los mayoristas, minoristas y otros involucrados en la cadena de suministro, sean capaces de aceptar la información transferida electrónicamente en el formato internacional ONIX.

# ONIX (ONline Information Exchange)

ONIX usa XML, en particular por las siguientes razones:

- XML está optimizado para crear documentos complejos y transmitir e intercambiar datos entre computadores.
- XML es lector de texto (text-readable) lo que significa que tanto los humanos como los computadores pueden reconocer y leer los datos. La mayoría de las etiquetas que definen cada elemento de dato, consisten de palabras o abreviaturas en inglés, por ejemplo, un mensaje ONIX puede poner el nombre de la editorial de la siguiente manera: "<PublisherName>Scribner's</PublisherName>". Estos factores facilitan a las pequeñas organizaciones diseñar e implementar sistemas compatibles con ONIX.
- El software XML es barato, lo que significa que aún las pequeñas editoriales pueden usarlo, cumpliendo así con el mayor objetivo del comité ONIX.

# ONIX (ONline Information Exchange)

ONIX puede definir muchas cosas en un libro, por ejemplo:

- título
- autor
- ISBN
- precio
- disponibilidad
- Opiniones y extractos
- Códigos de tema BISAC
- Derechos territoriales
- Ligas a sitios e imágenes de cubiertas de libros
- ...y mas!



Red Mexicana  
de Repositorios  
Institucionales

# SGML Standard Generalized Markup Language

## Lenguajes de marcas y estructuración de la información:

- SGML "Estándar de Lenguaje de Marcado Generalizado". Consiste en un sistema para la organización y etiquetado de documentos. La Organización Internacional de Estándares (ISO) normalizó este lenguaje ISO 8879:1986, Information processing — Text and office systems — Standard Generalized Markup Language (SGML).
- El lenguaje SGML sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.
- El lenguaje HTML está definido en términos del SGML. XML es un estándar de creación posterior, que incorpora un subconjunto de la funcionalidad del SGML (suficiente para las necesidades comunes), y resulta más sencillo de implementar pues evita algunas características avanzadas de SGML.
- 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.

# Referencia

## Lenguajes de marcas y estructuración de la información:

-  <http://www.dublincore.org/>
-  <http://www.w3.org/wiki/DublinCore>
-  <http://www.dublincore.org/documents/1999/07/02/dces/>
-  <http://www.loc.gov/standards/mods/>
-  <http://es.wikipedia.org/wiki/LOM>
-  [http://www.tei-c.org/Guidelines/Customization/Lite/teiu5\\_sp.html](http://www.tei-c.org/Guidelines/Customization/Lite/teiu5_sp.html)
-  <http://ltsc.ieee.org/wg12/>
-  <http://www.w3.org/RDF/>
-  <http://www.w3.org/TR/owl-features/>
-  <http://www.w3.org/2004/02/skos/intro>
-  <http://www.bisg.org/what-we-do-21-15-onix-for-books.php>
-  <http://www.w3.org/TR/NOTE-sgml-xml-971215>
-  <http://www.w3.org/MarkUp/SGML/>
-  <http://www.w3.org/2007/09/OWL-Overview-es.html>