

# Tema 4 Tecnologías para la web 3.0

## Primera Parte: RDF , Metadatos y OWL

J. Javier Samper Zapater  
José Fco. García Calderaro  
Departamento de Informática

VNIVERSITAT  ID VALÈNCIA

---

# Tema 4, Primera Parte

## “Resource Description Framework (RDF)”

### Metadatos y OWL

## Objetivos de esta parte: “RDF”

---

- Comprender como los recursos pueden ser descritos mediante RDF

# Índice

---

- Introducción a XML
- ¿Qué es RDF?
- Conceptos básicos
- El modelo RDF: Tripletas y Grafo.
- Utilizando la sintaxis XML.
- Contenedores

# XML

---

- XML: lenguaje para definir lenguajes con etiquetas semánticamente ricas:

**<titulo>El perfume</titulo>**

vs.

**<b>El perfume</b>**

- XML Schema:
  - Especificación para asignar tipos de datos, dominios, rangos de valores y restricciones a las etiquetas XML

# NameSpaces

---

- Namespace: son mecanismos simples para crear nombres globales únicos por los elementos y atributos del lenguaje de etiquetas.
  - Elimina los conflictos entre etiquetas que tengan el mismo nombre.
  - Diferentes lenguajes se pueden mezclar.
- Cada nombre de una **etiqueta** de **XML** debe consistir de dos partes:
  - **Un prefijo:** es una abreviación para el namespace en la declaración del namespace.
  - **Una parte local:** es el identificador para los metadatos.
    - `<xsd:integer>`
- Declaración:

`<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">`

# RDF - Definición

---

- **RDF: *Resource Description Framework***
- W3C: The *Resource Description Framework* (RDF) integrates a variety of applications from library catalogs and world-wide directories to syndication and aggregation of news, software, and content to personal collections of music, photos, and events using XML as an interchange syntax. The RDF specifications provide a lightweight ontology system to support the exchange of knowledge on the Web

# XML --> RDF

**XML**

```
<?xml version="1.0"?>
<River id="Yangtze"
  xmlns="http://www.geodesy.org/river">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

**Yangtze.xml**

“se transforma en”

Mayúsculas (clases)

**RDF**

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

**Yangtze.rdf**

Minúsculas (propiedades)

**Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation**

# Breve análisis del formato

- ① RDF provee un atributo ID para identificar el recurso (**resource**) que está siendo descrito
- ② El atributo ID está en el namespace **RDF**.

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

- ③ Añade el "fragmento identificador simbolo" a el namespace.

Roger L. Costello, David B. Jacobs ©2003 the MITRE Corporation

# Breve análisis del formato (cont.)

① Identifica el **tipo** (clase) del recurso que está siendo descrito.

② Identifica el **recurso** que es descrito. Este recurso es una instancia de River

③ Estas son **propiedades**, o **atributos**, del tipo (clase).

```
<?xml version="1.0"?>  
<River rdf:ID="Yangtze"  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns="http://www.geodesy.org/river#">  
  <length>6300 kilometers</length>  
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>  
  <endingLocation>East China Sea</endingLocation>  
</River>
```

④ **Valores** de las propiedades

Roger L. Costello, David B. Jacobs ©2003 the MITRE Corporation

# RDF Parser

---

- Hay un parser RDF en la web W3C:

<http://www.w3.org/RDF/Validator/>

Indica si el fichero XML está en el formato RDF apropiado

**Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation**

# ¿Qué es RDF?

---

- Finalidad: Proporcionar un modo sencillo de expresar afirmaciones acerca de recursos de la web (por ejemplo páginas Web) o del mundo real.
  - Es un sistema de descripción de recursos con base lógica
  - Relaciona recursos con propiedades y valores
- Trata de aportar interoperabilidad ante la multiplicidad de formatos incompatibles existentes.

# Motivación

---

**El desarrollo de RDF ha venido motivado por distintas razones:**

- La posibilidad de proporcionar información acerca de los recursos existentes en la Web y de los sistemas que los usan ( valorar contenidos de sitios web, descripción de su ámbito y capacidades, reglas de privacidad, etc.).
- La puesta en marcha de aplicaciones que requieren modelos de información abierta y no restringida (catalogación, descripción de procedimientos organizativos, anotación de recursos Web, etc.)
- Hacer procesables aquellos datos que la Web proporciona en forma de hipertexto, para que puedan ser usados fuera del entorno particular, para el que se crearon, de forma que se pueda trabajar con aplicaciones a escala de Internet.
- Facilitar el trabajo cooperativo entre aplicaciones, de forma que estas puedan combinar datos provenientes de distintas fuentes para obtener una nueva información.
- Procesar sin intervención humana, la información del Web mediante agentes de software.

# Conceptos básicos (I)

---

- RDF se basa en **afirmaciones** (statements)
- Ejemplo de afirmación: La página (recurso) <http://www.ejemplo.es> tiene como creador a Javier Samper.
- El fundamento del RDF es un modelo de datos, pensado para representar recursos con propiedades bien identificadas y con valores adecuados para estas propiedades. Para lograr este objetivo aparecen 3 elementos principales, que juntos reciben el nombre de tripleta (S,P,O).
- En principio la tripleta proporciona una base consistente para permitir idealmente, que el entendimiento y significado humano pueda ser interpretado por la máquina.
- Por tanto, los elementos principales de una afirmación son:
  - Sujeto o Recurso: Sobre qué vamos a hacer una afirmación (la página en este caso identificada mediante una URL).
  - Predicado o Propiedad: La propiedad del recurso que estamos describiendo (quién es su creador en este caso).
  - Objeto o Valor: Lo que vamos a asignar como valor a la propiedad anterior (el nombre de su autor).

# Conceptos básicos (II)

---

- Otras afirmaciones podrían ser:
  - La página <http://www.paginacualquiera.es> tiene como fecha-de-creación el 28/03/2002.
  - La página <http://www.paginacualquiera.es> tiene como idioma el Español.
- Para hacer estas afirmaciones comprensibles (procesables) por máquinas necesitaremos:
  - Identificadores para cada uno de los elementos de una afirmación que impidan cualquier posibilidad de confusión con otros similares.
  - Un lenguaje comprensible por máquinas (formal) para representar afirmaciones e intercambiarlas entre máquinas.

# Identificadores y lenguaje: URIs y XML.

---

- Identificadores: URIs

- URI: Uniform Resource Identifier. Identificadores universales para cualquier recurso de la red o de fuera de ella. (Diferenciar identificador de localizador).
- URI engloba URL. De hecho URL es un término informal para referirse a los URIs de los sistemas de identificación más populares: http, ftp, mailto, etc...
- Sintaxis definida.

Ejemplos de esquemas de URI para protocolos ftp, http...

```
ftp://ftp.is.co.za/rfc/rfc1808.txt, http://robotica.uv.es/~jsamper,  
mailto:jose.j.samper@uv.es
```

- ¡ Proporcionar un URI es equivalente a asignar a una persona un identificador único (numero de pasaporte) o a un libro un ISBN. !

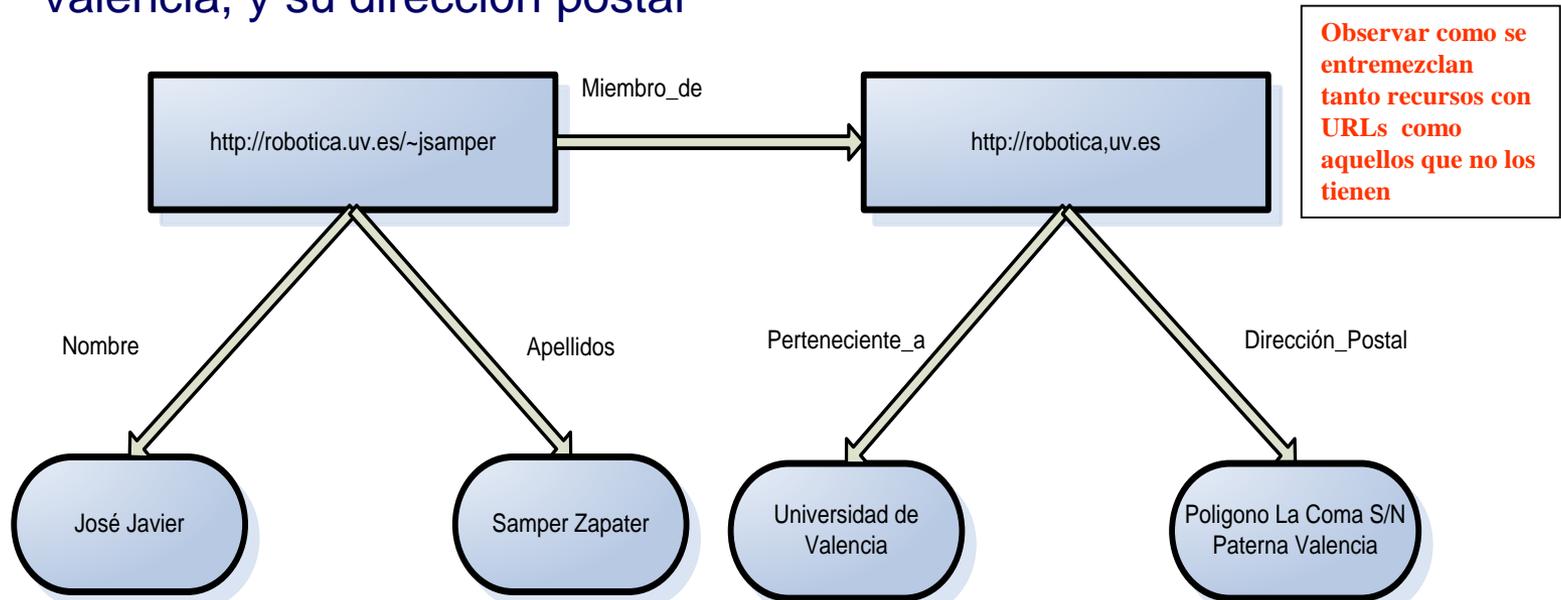
## Definición:

Una referencia URI (o URIref) es una URI, junto a un identificador de fragmento al final (separado por #) .Por ejemplo, en `http://www.example.org/index.html#seccion2`, `seccion2` sería el fragmento.

- Lenguaje formal: XML

# Grafo RDF: Nodos y arcos

- Un modo intuitivo de representar los elementos fundamentales de una afirmación (statement) es utilizando un grafo con nodos para los sujetos y objetos y con arcos para los predicados.
- En el ejemplo de abajo diremos que jsamper es miembro del Instituto de Robótica, y se describirá el nombre completo de la persona, así como también la pertenencia de dicho Instituto a la Universidad de Valencia, y su dirección postal



# Grafo RDF: Nodos y arcos

---

- Extensible indefinidamente. (basta con añadir nuevas conexiones)
- Control descentralizado utilizando URIs

# Clases

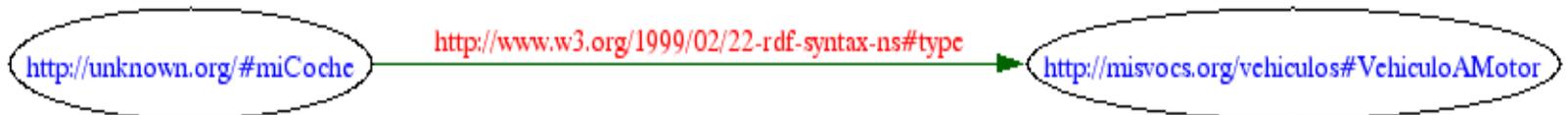
---

- En cualquier proceso de descripción el primer paso será clasificar las diferentes cosas (recursos) a describir.
- Una clase es un concepto genérico, un tipo, una categoría.
- Ejemplos
  - [www.google.com](http://www.google.com) es un buscador
  - [www.cursos.com/cursoXML](http://www.cursos.com/cursoXML) es un curso
  - [www.sia.eui.upm.es/docent/robotica.html](http://www.sia.eui.upm.es/docent/robotica.html) es una asignatura

# Clasificando recursos

- Para afirmar que un recurso pertenece a una clase o categoría (clasificar):

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://misvocs.org/vehiculos">
<rdf:Description rdf:about="http://unknown.org/miCoche">
  <rdf:type rdf:resource="http://misvocs.org/vehiculos#VehiculoAMotor"/>
  <ex:tipo>Monovolumen</ex:tipo>
</rdf:Description>
</rdf:RDF>
```



```
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://misvocs.org/vehiculos">
  <ex:VehiculoAMotor rdf:ID="miCoche">
    <ex:tipo>Monovolumen</ex:tipo>
  </ex:VehiculoAMotor>
</rdf:RDF>
```

**Abreviado:**

```
</rdf:RDF>
```

# Vocabularios: RDFS

- RDF no proporciona vocabularios específicos para hacer afirmaciones acerca de recursos.
- Un vocabulario sencillo: Dublin Core.



## Element Name: Title

Label: Title  
Definition: A name given to the resource.  
Comment: Typically, Title will be a name by which the resource is formal

## Element Name: Creator

Label: Creator  
Definition: An entity primarily responsible for making the content of the r  
Comment: Examples of Creator include a person, an organization, or a s

## Element Name: Subject

Label: Subject and Keywords  
Definition: A topic of the content of the resource.  
Comment: Typically, Subject will be expressed as keywords, key phrase  
best practice is to select a value from a controlled vocabulary

## Element Name: Description

Label: Description  
Definition: An account of the content of the resource.  
Comment: Examples of Description include, but is not limited to: an abst  
free-text account of the content.

## Element Name: Publisher



```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
- <rdf:RDF xmlns:dcterms="http://purl.org/dc/terms/" xmlns:dc="http://p
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:rdf="ht
- <rdf:Description rdf:about="http://purl.org/dc/elements/1.1/">
  <dc:title xml:lang="en-US">The Dublin Core Element Set v1.1 names;
    Schema</dc:title>
  <dc:publisher xml:lang="en-US">The Dublin Core Metadata Initiative<
  <dc:description xml:lang="en-US">The Dublin Core Element Set v1.1 n
    v1.1. Entries are declared using RDF Schema language to support
  <dc:language xml:lang="en-US">English</dc:language>
  <dcterms:issued>1999-07-02</dcterms:issued>
  <dcterms:modified>2003-03-24</dcterms:modified>
  <dc:source rdf:resource="http://dublincore.org/documents/dces/" /
  <dc:source rdf:resource="http://dublincore.org/usage/decisions/" /
  <dcterms:isReferencedBy rdf:resource="http://www.dublincore.org/d
  <dcterms:isRequiredBy rdf:resource="http://purl.org/dc/terms/" />
  <dcterms:isReferencedBy rdf:resource="http://purl.org/dc/dcmitype,
  </rdf:Description>
- <rdf:Property rdf:about="http://purl.org/dc/elements/1.1/title">
  <rdfs:label xml:lang="en-US">Title</rdfs:label>
  <rdfs:comment xml:lang="en-US">A name given to the resource.</rdfs:
  <dc:description xml:lang="en-US">Typically, a Title will be a name by
  <rdfs:isDefinedBy rdf:resource="http://purl.org/dc/elements/1.1/" /
  <dcterms:issued>1999-07-02</dcterms:issued>
  <dcterms:modified>2002-10-04</dcterms:modified>
  <dc:type rdf:resource="http://dublincore.org/usage/documents/pr
  <dcterms:hasVersion rdf:resource="http://dublincore.org/usage/terr
  </rdf:Property>
```

- RDFS permite definir vocabularios: clases, propiedades..

# Definiendo clases con RDFS (I)

- RDFS es también un vocabulario que utiliza RDF. Los elementos prefijados con rdfs: pertenecen a ese vocabulario.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<rdf:Description rdf:ID="VehiculoAMotor">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Class"/>
</rdf:Description>
</rdf:RDF>
```



# Definiendo clases con RDFS (II)

---

- O escrito de forma abreviada:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:ID="VehiculoAMotor"/>
</rdf:RDF>
```

- Para definir jerarquías de clases uso la relación subclase-de:

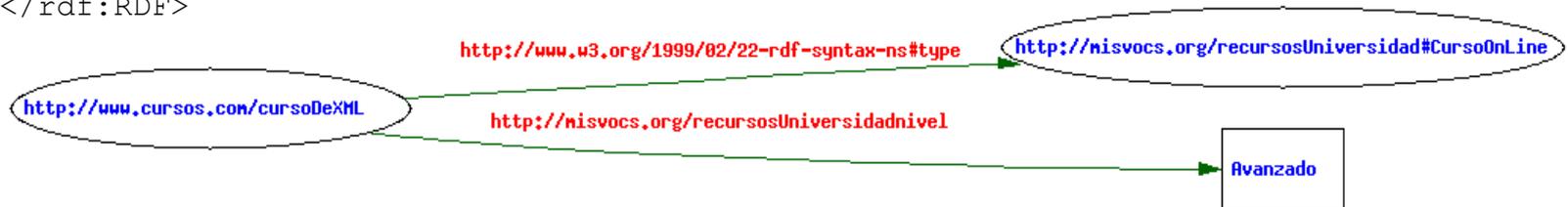
...

```
<rdf:Description rdf:ID="Furgon">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
  schema#Class"/>
  <rdfs:subClassOf rdf:resource="#VehiculoAMotor"/>
</rdf:Description>
```

# Instancias de clases

- Para afirmar que un recurso pertenece a una clase:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ru="http://misvocs.org/recursosUniversidad">
<rdf:Description rdf:about="http://www.cursos.com/cursoDeXML">
  <rdf:type rdf:resource="http://misvocs.org/recursosUniversidad#CursoOnLine"/>
  <ru:nivel>Avanzado</ru:nivel>
</rdf:Description>
</rdf:RDF>
```



```
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns: ru="http://misvocs.org/recursosUniversidad">
  <ru:CursoOnLine rdf:about="http://www.cursos.com/cursoDeXML">
    <ru:nivel>Avanzado</ru:nivel>
  </ru:CursoOnLine>
</rdf:RDF>
```

**Abreviado:**

```
<ru:CursoOnLine rdf:about="http://www.cursos.com/cursoDeXML">
  <ru:nivel>Avanzado</ru:nivel>
</ru:CursoOnLine>
</rdf:RDF>
```

# RDFS Propiedades (I)

---

- Lo lógico después de clasificar es definir propiedades que caracterizan las clases creadas.

```
<rdf:Description rdf:ID="creadaPor">  
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-  
    ns#Property"/>  
</rdf:Description>
```

- O escrito de forma abreviada:

```
<rdfs:Property rdf:ID="creadaPor"/>
```

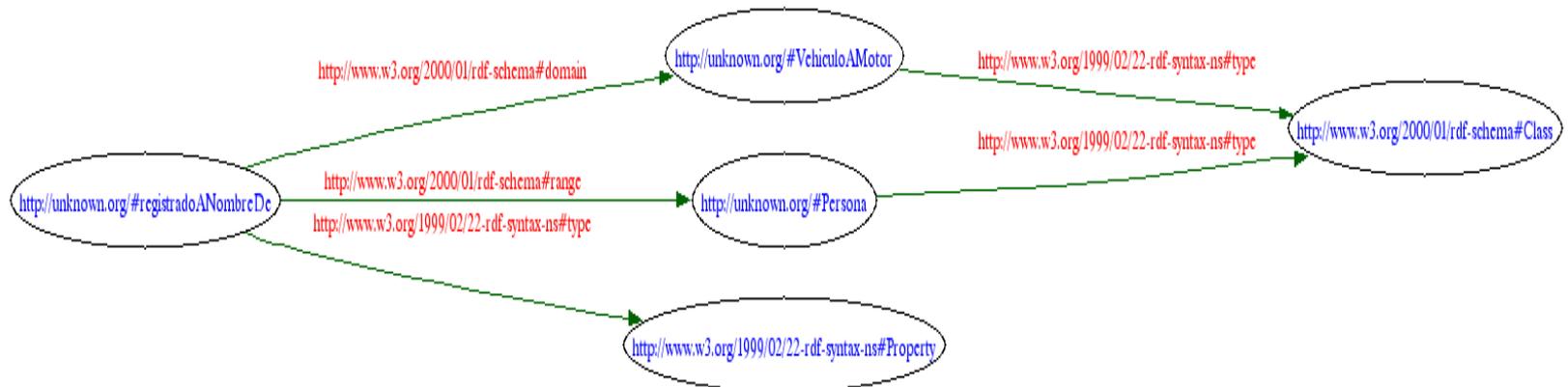


# RDFS Propiedades (II)

- Podemos especificar dominio y rango para las propiedades definidas.

```
<rdf:Description rdf:ID="registradoANombreDe">  
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>  
  <rdfs:domain rdf:resource="#VehiculoAMotor"/>  
  <rdfs:range rdf:resource="#Persona"/>  
</rdf:Description>
```

- El dominio y rango siempre son clases.



# Uso de rdf:datatype (I)

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.uom.org/distance#">

  <simpleType name="kilometer">
    <restriction base="integer">
      </restriction>
    </simpleType>

  <simpleType name="meter">
    <restriction base="integer">
      </restriction>
    </simpleType>
</schema>
```

**uom.xsd**

**Previamente se definen los tipos de datos “kilometer” y “meter” mediante XMLSchema (archivo uom.xsd)**

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length rdf:datatype="http://www.uom.org/distance#kilometer">6300</length>
  <maxWidth rdf:datatype="http://www.uom.org/distance#meter">175</maxWidth>
  <maxDepth rdf:datatype="http://www.uom.org/distance#meter">55</maxDepth>
</River>
```

**Yangtze5.rdf**

# Uso de rdf:datatype (II)

---

- Otra posibilidad es el uso de tipos ya definidos en XMLSchema como NonNegativeInteger, integer, string etc.

```
<?xml version="1.0"?>
<Vehicle rdf:ID="Micoche"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.vehicles.org#">
  <speed rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">200</speed>
</Vehicle>
```

# Identificación única rdf:ID

- **Sirve para identificar unívocamente un recurso**
- El valor de rdf:ID es una "URI relativa".
- La "URI completa" es obtenida concatenando la URL del documento XML con "#" y después el valor of rdf:ID, ej.,

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

## Yangtze.rdf

Suponer que este RDF/XML documento esta localizado en la URL: <http://www.china.org/geography/rivers>.  
Por lo que , la URI completa para este recurso es:

<http://www.china.org/geography/rivers#Yangtze>

¡OJO No  
referenciamos un  
fichero ( \*.rdf) !

Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation

# Utilizando la sintaxis de XML xml:base

- En la diapositiva anterior vimos como el URL del documento proporcionó la base URI.
- Depender de la ubicación del documento es frágil: esto fallará si el documento es movido, o es copiado a otra ubicación.
- Una solución más robusta consiste en especificar la base URI en el documento, p.ej.,

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#"
  xml:base="http://www.china.org/geography/rivers">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

Resource URI = concatenation(xml:base, '#', rdf:ID)  
= concatenation(http://www.china.org/geography/rivers, '#', "Yangtze")  
= http://www.china.org/geography/rivers#Yangtze

**Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation**

# Utilizando la sintaxis de XML rdf:about

- En lugar de identificar un recurso con una URI relativa (lo cual requiere una base URI previa), podemos dar la identificación completa de un recurso. Sin embargo se usa más rdf:about, que rdf:ID, ej.,

```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/geography/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

Roger L. Costello, David B. Jacobs ©2003 the MITRE Corporation

# rdf:ID versus rdf:about

---

¿Cuándo se usará rdf:ID? ¿Cuándo se usará rdf:about?

- Cuando queramos **introducir** un recurso, y proveer información inicial sobre él, usaremos rdf:ID
- Cuando queramos **extender** la información dada inicialmente se usará rdf:about
  - La filosofía RDF es semejante a la filosofía de la Web. Es decir cualquiera, en cualquier lugar, en cualquier momento, puede proporcionar información sobre un recurso.

Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation

# Utilizando la sintaxis de XML

## rdf:Description

---

- Un documento RDF es una lista de descripciones de recursos
- Cada descripción corresponde a un recurso y contiene una lista de propiedades que lo definen.
  - Los valores de una propiedad pueden ser literales u otros recursos (definidos por su URI correspondiente).
- Una descripción se representa con un elemento XML `<rdf:Description>` con uno de los siguientes atributos: `rdf:about` o `rdf:ID` (puede no haber ninguno, recurso anónimo)

```
<rdf:Description about="http://robotica.uv.es/~jsamper">
```

```
...
```

```
</rdf:Description>
```

```
<rdf:Description ID="JJSZ">
```

```
...
```

```
</rdf:Description>
```

```
<rdf:Description>
```

```
...
```

```
</rdf:Description>
```

# Utilizando la sintaxis de XML

## rdf:Description / rdf:resource

---

- Un elemento *rdf:Description*
  - Tiene un atributo about (para describir cualquier recurso) o ID (para definirlo) que identifica el recurso descrito.
  - Contiene una secuencia de elementos hijos (propiedades) con sus valores correspondientes.
- Los valores de las propiedades
  - Puede contener texto (Literal).

```
<dc:Title>Sitio web del Departamento de
Informatica.</dc:Title>
```

- Puede tener como valor otro recurso, en cuyo caso se representará mediante un elemento vacío con el atributo *rdf:resource* cuyo valor es el URI del objeto.

```
<dc:Creator rdf:resource="mailto:jose.j.samper@uv.es"/>
```

- Puede contener otro elemento rdf:Description.

```
<dc:Creator>
  <rdf:Description rdf:about="mailto:jose.j.samper@uv.es">
    <os:TrabajaCon rdf:resource="mailto:jfgarcia@robotica.uv.es"/>
  </rdf:Description>
</dc:Creator>
```

# rdf:Description + rdf:type

- Hay todavía otro modo de representar XML. Esta forma clarifica que describimos algo y de qué tipo (la clase) se trata:

```
<?xml version="1.0"?>
<rdf:Description rdf:about="http://www.china.org/geography/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <rdf:type rdf:resource="http://www.geodesy.org/river#River"/>
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</rdf:Description>
```

Esto se lee como: “Esto es una descripción del recurso <http://www.china.org/geography/rivers#Yangtze>. Este recurso es una instancia de tipo (clase) `River`. El recurso <http://www.china.org/geography/rivers#Yangtze> tiene una longitud de 6300 kilómetros, un `startingLocation` de *western China's Qinghai-Tibet Plateau*, Y un `endingLocation` de *East China Sea*.”

*Nota: La forma que hemos visto antes es una abreviatura de esta forma larga. Un Analizador gramatical de RDF interpreta la forma abreviada como si se tratara de esta larga.*

**Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation**

# Representaciones Equivalentes

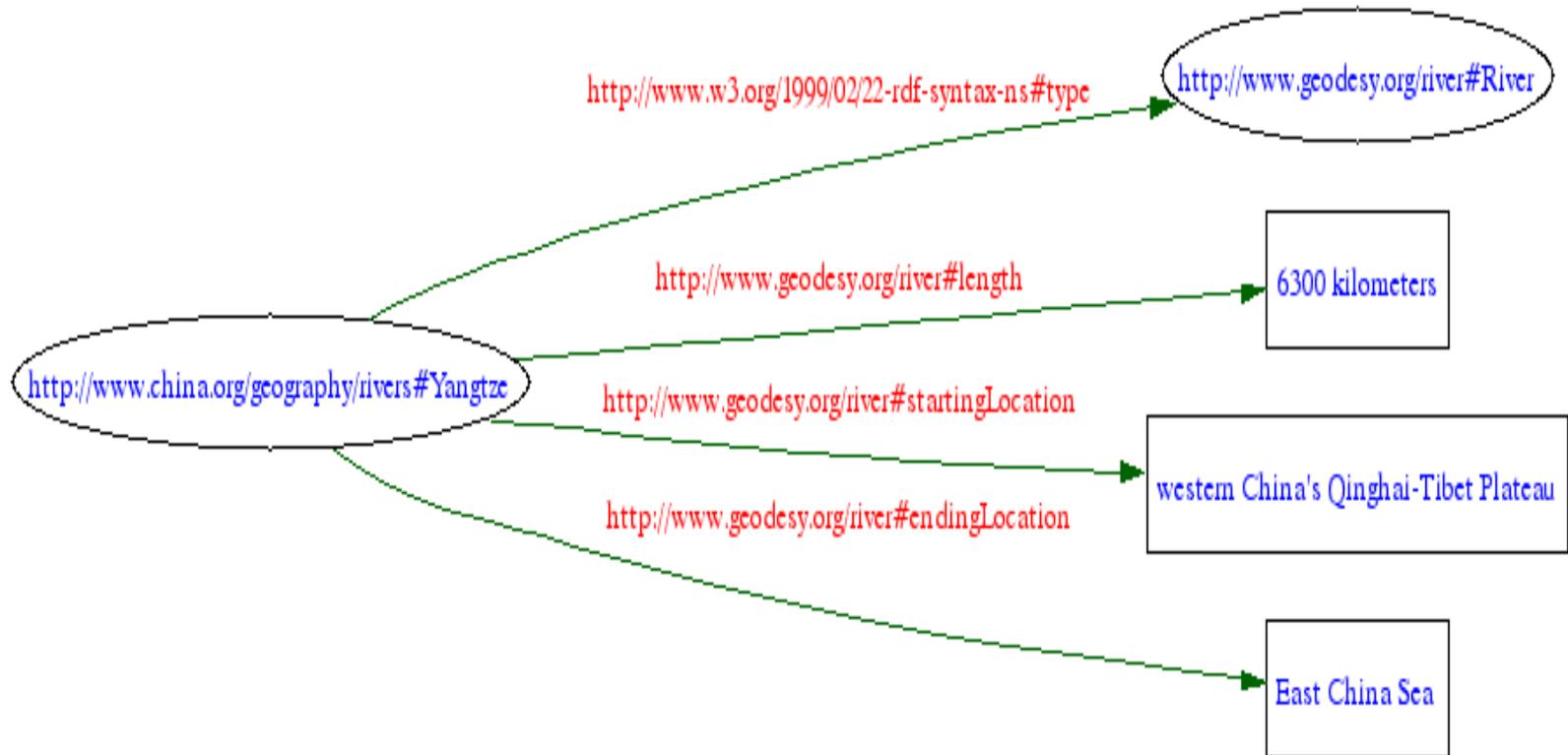
```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#"
  xml:base="http://www.china.org/geography/rivers">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/geography/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

```
<?xml version="1.0"?>
<rdf:Description rdf:about="http://www.china.org/geography/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <rdf:type rdf:resource="http://www.geodesy.org/river#River"/>
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</rdf:Description>
```

← Nota: en la literatura RDF los ejemplos son típicamente mostrados de esta forma.

# Equivalente en grafo



Legenda:

Elipse indica "Recurso"

Rectangulo indica "valor"

# rdf:value

XML

```
<?xml version="1.0"?>
<River id="Yangtze"
  xmlns="http://www.geodesy.org/river"
  xmlns:uom="http://www.measurements.org/units-of-measure">
  <length uom:units="kilometers">6300</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

Yangtze4.xml



RDF

Yangtze4.rdf

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#"
  xmlns:uom="http://www.measurements.org/units-of-measure#">
  <length>
  <rdf:Description>
    <rdf:value>6300</rdf:value>
    <uom:units>kilometers</uom:units>
  </rdf:Description>
  </length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

**RDF no permite atributos sobre las propiedades, ( excepto rdf:resource)**

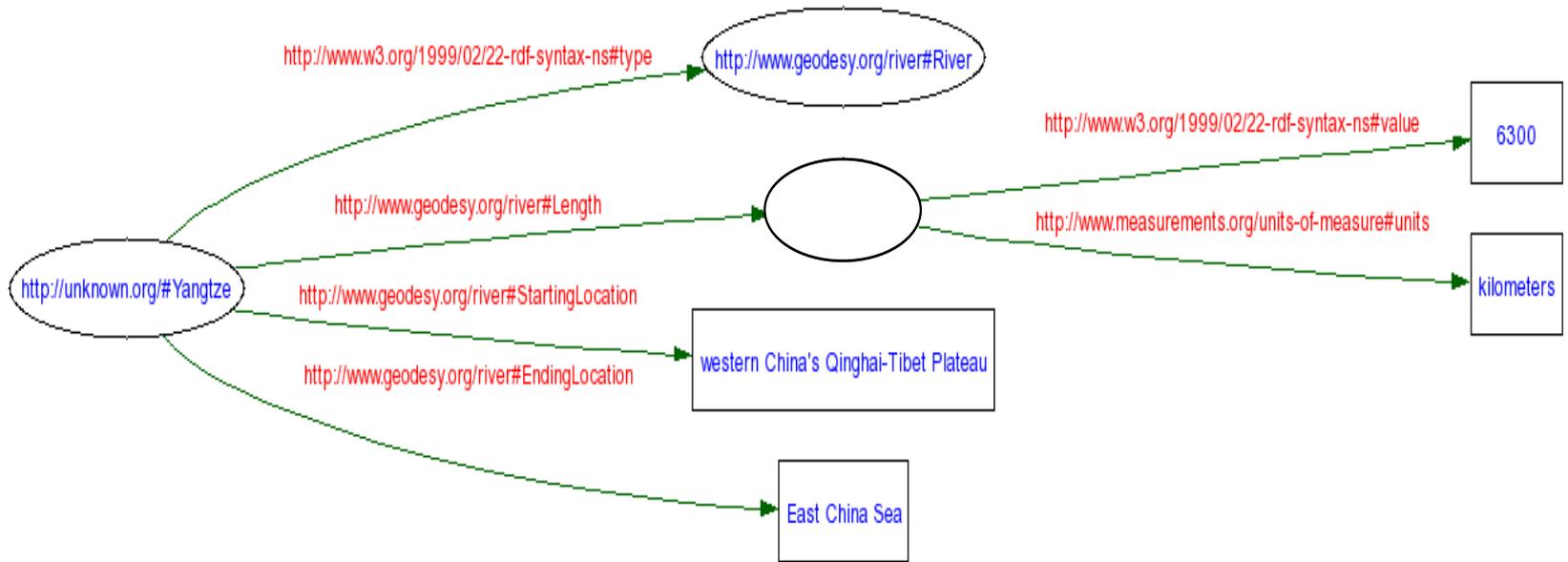
**SOLUCIÓN:**  
6300 es un primer valor al que se le añade información adicional (kilometers)

# Ventaja de los recursos anónimos

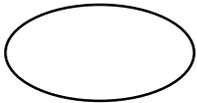
```
<rdf:Description>  
  <rdf:value>6300</rdf:value>  
  <uom:units>kilometers</uom:units>  
</rdf:Description>
```

Esto es un recurso anónimo. Su propósito es solamente proporcionar un contexto para las dos propiedades. Otros documentos de RDF no tendrán ninguna necesidad de ampliar este recurso. En este caso, no hay así pues, razón de dar al recurso un identificador. En este caso tiene sentido utilizar un recurso anónimo.

# Modelo RDF (grafo)



## Legend:



An anonymous resource (also called a "*blank node*"). That is, a resource with no identifier. (Note: RDF Parsers will typically generate a unique identifier for anonymous resources, to distinguish one anonymous resource from another.)

# rdf:parseType="Resource"

Si el valor de una propiedad está compuesto por varios valores entonces una opción es crear un **Recurso anónimo**.

RDF provee una manera abreviada, con la cual no se necesita crear un elemento `rdf:Description`, mediante **`rdf:parseType="Resource"`**:

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#"
  xmlns:uom="http://www.measurements.org/units-of-measure#">
  <length rdf:parseType="Resource">
    <rdf:value>6300</rdf:value>
    <uom:units>kilometers</uom:units>
  </length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

**Yangtze4,v2.rdf**

**Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation**

# Equivalentes!

```
<length>
  <rdf:Description>
    <rdf:value>6300</rdf:value>
    <uom:units>kilometers</uom:units>
  </rdf:Description>
</length>
```

```
<length rdf:parseType="Resource">
  <rdf:value>6300</rdf:value>
  <uom:units>kilometers</uom:units>
</length>
```

Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation

# Interpretando RDF

En este documento el recurso es anónimo – no posee identificador.

```
<?xml version="1.0"?>
<River xmlns="http://www.geodesy.org/river#">
  <name>Yangtze</name>
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

**Yangtze3.rdf**

Esta especificación se lee como: “Esto es una instancia del tipo River (clase). El River recibe el nombre de Yangtze, tiene una longitud de 6300 kilometers, su startingLocation es western China's Qinghai-Tibet Plateau, y su endingLocation es el East China Sea.”

**Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation**

# Desventaja de usar recursos anónimos

```
<?xml version="1.0"?>
<River xmlns="http://www.geodesy.org/river#">
  <name>Yangtze</name>
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

<http://www.china.org/geography/rivers/yangtze.rdf>

```
<?xml version="1.0"?>
<River xmlns="http://www.geodesy.org/river#">
  <name>Yangtze</name>
  <name>Dri Chu - Female Yak River</name>
  <name>Tongtian He, Travelling-Through-the-Heavens River</name>
  <name>Jinsha Jiang, River of Golden Sand</name>
</River>
```

<http://www.encyclopedia.org/yangtze-alternate-names.rdf>

Una herramienta de agregación no podría determinar si estos documentos están hablando del mismo recurso.

**Agregar**

Roger L. Costello, David B. Jacobs ©2003 the MITRE Corporation

# Solución: Identificar Unívocamente

<http://www.china.org/geography/rivers/yangtze.rdf>

```
<?xml version="1.0"?>
<River
rdf:about="http://www.china.org/geography/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet
Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

<http://www.encyclopedia.org/yangtze-alternate-names.rdf>

```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/geography/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <name>Dri Chu - Female Yak River</name>
  <name>Tongtian He, Travelling-Through-the-Heavens River</name>
  <name>Jinsha Jiang, River of Golden Sand</name>
</River>
```

**La herramienta agrega datos sobre el recurso Yangtze.**

**Datos agregados!**

**Cualquier documento RDF que identifique el recurso, añadirá datos sobre el recurso. ( recordar: cualquiera, en cualquier lugar y en cualquier instante puede decir algo sobre él)**

**Datos distribuidos en red!**

```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/geography/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
  <name>Dri Chu - Female Yak River</name>
  <name>Tongtian He, Travelling-Through-the-Heavens River</name>
  <name>Jinsha Jiang, River of Golden Sand</name>
</River>
```

Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation

# Otro ejemplo de agregación

```
<?xml version="1.0"?>
<Dam rdf:ID="ThreeGorges"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/dam#"
  xml:base="http://www.china.org/geography/rivers">
  <name>The Three Gorges Dam</name>
  <width>1.5 miles</width>
  <height>610 feet</height>
  <cost>$30 billion</cost>
</Dam>
```

<http://www.encyclopedia.org/three-gorges-dam.rdf>

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#"
  xml:base="http://www.china.org/geography/rivers">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
  <obstacle rdf:resource="http://www.china.org/geography/rivers #ThreeGorges"/>
</River>
```

<http://www.china.org/geography/rivers/yangtze.rdf>

Agregar!

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#"
  xml:base="http://www.china.org/geography/rivers">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
  <obstacle>
    <Dam rdf:ID="ThreeGorges"
      xmlns="http://www.geodesy.org/dam#">
      <name>The Three Gorges Dam</name>
      <width>1.5 miles</width>
      <height>610 feet</height>
      <cost>$30 billion</cost>
    </Dam>
  </obstacle>
</River>
```

Fijarse que la referencia al recurso ThreeGorges Dam ha sido reemplazada por la información extra que encontró la herramienta de agregación

# Contenedores (I)

---

- La propiedad de pertenencia se describe con el elemento *rdf:li* (por list item)
  - Un Parser RDF reemplazará cada *rdf:li* con *rdf:\_1*, *rdf:\_2* y *rdf:\_3*, etc
- Un contenedor se describe con uno de los elementos siguientes *rdf:Bag*, *rdf:Seq* o *rdf:Alt*.
  - *rdf:Bag* type es usada para representar una colección sin orden
  - *rdf:Seq* type es usada para representar una secuencia de propiedades
  - *rdf:Alt* type es usada para representar un conjunto de propiedades alternativas
- Se usan como un Description pero no pueden tener un *rdf:about*, solamente *rdf:ID*

# Contenedores (II)

Aseveración: "Los estudiantes del curso 6.001 son Amy, Tim, John, Mary, y Sue",.

Se podría describir el curso asignando a la propiedad `s:students` un valor que sea un contenedor del tipo `rdf:Bag` (el grupo de estudiantes) para a continuación, usando las propiedades de pertenencia, describir los estudiantes individuales, como miembros de este contenedor, como muestra el grafo RDF.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

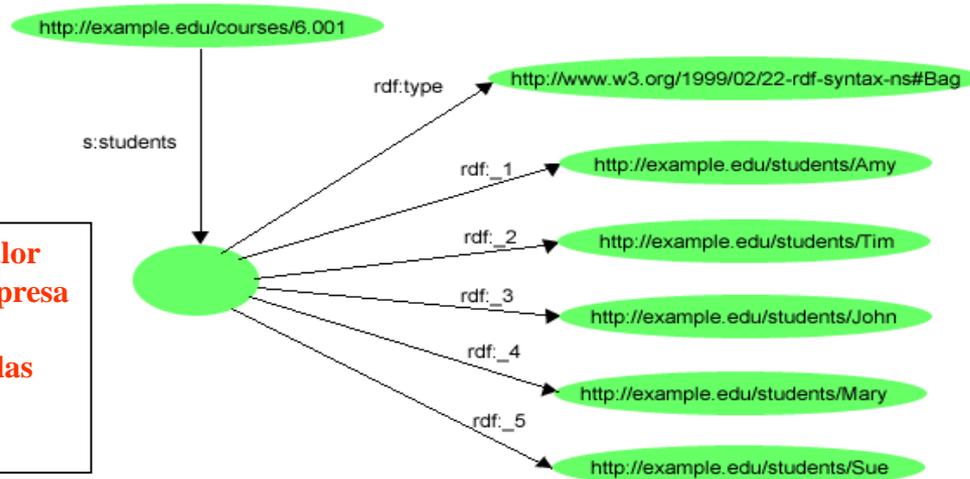
...

```
<rdf:Bag rdf:ID="students_from_6.001">
  <rdf:li rdf:resource="http://example.edu/students/Amy"/>
  <rdf:li rdf:resource="http://example.edu/students/Tim"/>
  <rdf:li>
```

...

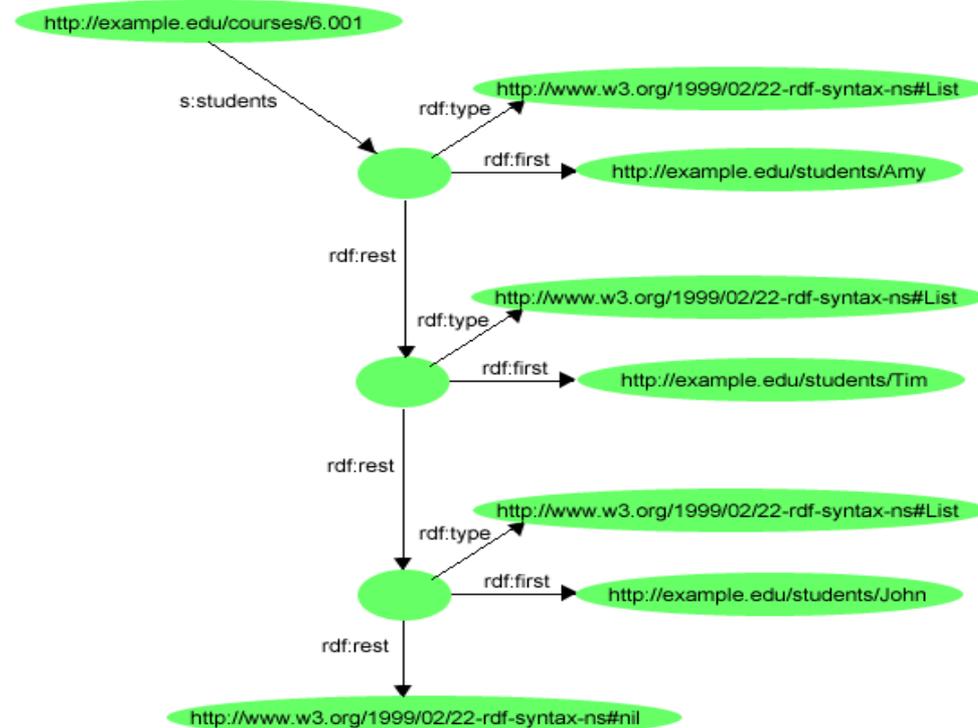
```
</rdf:li>
</rdf:Bag>
</rdf:RDF>
```

**Puesto que en este ejemplo el valor de la propiedad `s:students` se expresa por un Bag, no se quiere dar significación al orden dado por las URIs de cada estudiante**



# Colecciones RDF

- RDF proporciona un soporte para describir grupos que contengan solamente miembros específicos usando las llamadas *colecciones* RDF. Una colección RDF es un grupo de cosas representadas como una estructura de lista en un grafo RDF. Esta se construye utilizando un *vocabulario de colección* predefinido, consistente en el tipo `rdf:List`, las propiedades `rdf:first` y `rdf:rest`, y el recurso predefinido `rdf:nil`.
- Para ilustrarlo, representemos la aseveración “Los estudiantes del curso 6.001 son Amy, Tim y John”



# rdf:parseType="Collection"

---

- Puede ser añadido como un atributo de una propiedad para indicar que el contenido de la propiedad es una **lista de recursos**.
- Por ejemplo:

```
<books rdf:parseType="Collection">  
  <Book rdf:ID="_0-06-099325-2" ...  
  </Book>  
  
  ....  
  
  <Book rdf:ID="_0-06-099325-2" ...  
  </Book>  
</books>
```

# rdf:parseType="Literal"

```
<?xml version="1.0"?>
<Article id="QuickBrownFox"
  xmlns="http://www.publishing.org">
  <paragraph>
    The quick brown <bold>fox</bold> jumped over the lazy dog.
    An important person once said "<quote>Now is the time for all
    good men to come to the aid of their country</quote>"
  </paragraph>
</Article>
```

QuickBrownFox.xml

```
<?xml version="1.0"?>
<Article rdf:ID="QuickBrownFox"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.publishing.org#">
  <paragraph rdf:parseType="Literal">
    The quick brown <bold>fox</bold> jumped over the lazy dog.
    An important person once said "<quote>Now is the time for all
    good men to come to the aid of their country</quote>"
  </paragraph>
</Article>
```

QuickBrownFox.rdf

Contenido que mezcla etiquetas de formato con cadana de texto

rdf:parseType="Literal"  
Indica que el contenido debe ser tratado simplemente como un literal.

¡OJO porque se pierde la habilidad para que una herramienta pueda beneficiarse de la estructura S/P/O!

# Uso de atributos

XML

```
<?xml version="1.0"?>
<River id="Yangtze"
  xmlns="http://www.geodesy.org/river"
  length="6300 kilometers"
  startingLocation="western China's Qinghai-Tibet Plateau"
  endingLocation="East China Sea"/>
```

Yangtze.xml



RDF

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#"
  xmlns:r="http://www.geodesy.org/river#"
  r:length="6300 kilometers"
  r:startingLocation="western China's Qinghai-Tibet Plateau"
  r:endingLocation="East China Sea"/>
```

Yangtze.rdf

**Fijarse en que son usados Atributos (propiedades) en lugar de etiquetas hijas**

**El formato RDF permite también usar atributos. Por supuesto una propiedad puede unicamene ser representada como un atributo si tiene un valor literal, no un valor estructurado.** AWA

# Equivalentes

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
</River>
```

```
<?xml version="1.0"?>
<River rdf:ID="Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/river#"
  xmlns:r="http://www.geodesy.org/river#"
  r:length="6300 kilometers"
  r:startingLocation="western China's Qinghai-Tibet Plateau"
  r:endingLocation="East China Sea"/>
```

# Ejercicios

---

- 1) Escribir la afirmación en lenguaje natural
- 2) Modificar el siguiente fichero XML para que sea adecuado en el modelo RDF.
- 3) ¿Cómo sería el modelo RDF en grafo?

```
<?xml version="1.0"?>
<River id="Yangtze"
  xmlns="http://www.geodesy.org/river">
  <length>6300 kilometers</length>
  <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
  <endingLocation>East China Sea</endingLocation>
  <Dam id="ThreeGorges"
    xmlns="http://www.geodesy.org/dam">
    <name>The Three Gorges Dam</name>
    <width>1.5 miles</width>
    <height>610 feet</height>
    <cost>$30 billion</cost>
  </Dam>
</River>
```

Yangtze2.xml

# Ejercicios

Alternativamente, suponer que alguien ya ha creado un documento que contiene información sobre Three Gorges Dam:

```
<?xml version="1.0"?>
<Dam rdf:ID="ThreeGorges"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns="http://www.geodesy.org/dam#"
      xml:base="http://www.china.org/geography/rivers">
  <name>The Three Gorges Dam</name>
  <width>1.5 miles</width>
  <height>610 feet</height>
  <cost>$30 billion</cost>
</Dam>
```

**Three-Gorges-Dam.rdf**

4) ¿Cómo se simplificaría la referencia?

**Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation**

# Ejercicios

- Modificar el siguiente fichero XML para que sea adecuado en el modelo RDF.

## Barnes\_and\_Noble\_BookCatalogue.xml

```
<?xml version="1.0"?>
<Catalogue xmlns="http://www.publishing.org#"
  xmlns:dc="http://purl.org/metadata/dublin-core#">
  <Book>
    <dc:Title>Lateral Thinking</dc:Title>
    <dc:Creator>Edward de Bono</dc:Creator>
    <dc:Date>1973</dc:Date>
    <ISBN>0-06-099325-2</ISBN>
    <dc:Publisher>Harper & Row</dc:Publisher>
  </Book>
  <Book>
    <dc:Title>Illusions: The Adventures of a Reluctant Messiah</dc:Title>
    <dc:Creator>Richard Bach</dc:Creator>
    <dc:Date>1977</dc:Date>
    <ISBN>0-440-34319-4</ISBN>
    <dc:Publisher>Dell Publishing Co.</dc:Publisher>
  </Book>
  <Book>
    <dc:Title>The First and Last Freedom</dc:Title>
    <dc:Creator>J. Krishnamurti</dc:Creator>
    <dc:Date>1954</dc:Date>
    <ISBN>0-06-064831-7</ISBN>
    <dc:Publisher>Harper & Row</dc:Publisher>
  </Book>
</Catalogue>
```

Ojo: malogradamente, Dublin Core no sigue la convección de nombres de RDF, apareciendo la primera letra de las propiedades en mayúsculas

# Ejercicios

- Modificar el siguiente fichero XML para que sea adecuado en el modelo RDF. Hacer uso de `rdf:parseType="Collection"`

```
<?xml version="1.0"?>
<Resume xmlns="http://www.resume.org#">
  <Person xmlns="http://www.person.org#">
    <name>John Smith</name>
    <phone>555-1212</phone>
    <email>abc@def.com</email>
  </Person>
  <experience>
    <Job>
      <startDate>1995-01-01</startDate>
      <endDate>1999-01-01</endDate>
      <title>researcher</title>
      <employer>Some Corp</employer>
      <employerHomepage>http://www.company.org</employerHomepage>
      <description>Cool stuff</description>
    </Job>
  </experience>
  <skills>
    <Skill name="Oracle" yrs="13"/>
    <Skill name="Java" yrs="3"/>
  </skills>
  <education>
    <Degree type="BS" school="RPI" year="1987"/>
  </education>
</Resume>
```

# Ejemplos de uso reales

- Foto RDF-Gen: Generador de ficheros RDF para describir fotografías y otras imágenes

**Foto RDF-Gen: Generador de ficheros RDF para describir fotografías y otras imágenes.**

¿Qué es Foto RDF-Gen?

RDF-Gen es una aplicación, que mediante javascript, genera el contenido de ficheros RDF que describen imágenes, especialmente fotografías. Ésta es la **última versión**: 0.3, correspondiente al día 2005-04-1. Puedes ver también la [versión en inglés](#).

¿Cómo puedo generar un fichero RDF-XML con Foto RDF-Gen? **Instrucciones.**

1. **Complete los campos del formulario.** En esta versión, **es imprescindible tener activado javascript**. Es recomendable completar todos los campos posibles, para aumentar la información sobre la imagen.
2. Pulsamos el botón "Generar Fichero RDF".
3. **Copiamos el contenido que se ha generado** en el campo de texto, justo debajo del apartado "Contenido RDF generado".
4. **Pegamos** el contenido en un **editor de texto plano** y lo guardamos con la extensión ".rdf".
5. Colocamos el fichero en internet, para que los robots de búsqueda cataloguen la fotografía con la información del fichero RDF.
6. Para asegurarnos de que el fichero es correcto, **comprobar la validez de la sintaxis del fichero** generado en [RDF Validation Service](#) del W3C, siguiendo [estas instrucciones](#).

**Introducción de datos.**

**Datos generales de la foto**

**URL** (obligatorio, por ejemplo: [http://www.ejemplo.org/foto\\_de\\_ejemplo.jpg](http://www.ejemplo.org/foto_de_ejemplo.jpg)):

**Título** (obligatorio, por ejemplo: *Fotografía de la Sierra de Gredos*):

**Descripción** (por ejemplo: *Desde el Valle del Tiétar, en el término municipal de Sotillo, se puede ver la Sierra de Gredos, poblada de pinos.*):

---

# Tema 4, Primera Parte

## “Metadatos”

# Objetivos de esta parte

---

- Comprender que son y para que sirven los metadatos.
- Comprobar la utilidad de la definición de metadatos para la descripción de todo tipo de recursos.
- Entender como se deben utilizar los metadatos para potenciar las finalidades de la Web semántica.
- Conocer algunos proyectos involucrados en la definición de estándares de metadatos: Dublin Core, FOAF y LOM
- Dar a conocer distintas herramientas de edición de metadatos.
- En definitiva, responder a las preguntas:
  - ¿cómo manejar los volúmenes de información que se están generando?
  - ¿Cómo tener capacidad expresiva generalizada?

# Introducción

---

El campo de la **Inteligencia Artificial** ha hecho pocos progresos en los últimos cuarenta años, así que si queremos que las máquinas nos entiendan hay que poner algo de nuestra parte: debemos incorporar **metadatos** (datos que describen a los datos) en los documentos. RDF/RDFS permite hacer esto en los documentos de la Web. El marcado de documentos mediante RDF/RDFS es ya un primer paso para que motores de búsqueda, navegadores, agentes inteligentes y humanos puedan emplear la información semántica de los documentos.

# ¿Qué son los metadatos? (I)

---

- Los **metadatos** pueden ser definidos como:
  - De forma simple: “**datos sobre datos**”.
  - **Información sobre recursos** (Web), que puede ser procesada y “entendida” por las máquinas. (Tim Berners-Lee, W3C, 1997)
- Un registro de metadatos consiste en un conjunto de atributos, o elementos, necesario para describir la fuente en cuestión.
  - Por ejemplo, un sistema de metadatos común entre los bibliotecarios --el catálogo de biblioteca -- contiene un conjunto de registros de metadatos con elementos que describen un libro u otra publicación en una biblioteca: autor, título, fecha de creación o publicación, materia, y la signatura topográfica especificando la localización de la publicación en el estante.
- La relación entre un registro de metadatos y el recurso al que describe puede darse de una de estas dos formas:
  1. los elementos pueden estar en un registro separado del documento, como en el caso del registro de un catálogo de bibliotecas;
  2. los metadatos pueden estar incluidos [*embedded*] en el propio recurso.

# ¿Qué son los metadatos? (II)

---

- **Problemática actual:** "Sobrecarga de información" que resulta de las grandes cantidades de datos digitales disponibles en línea, lo cual dificulta la obtención de información precisa mediante los servicios de búsqueda Web actuales.
- "La asociación de metadatos descriptivos normalizados a los objetos de la red tiene el potencial para mejorar sustancialmente las capacidades de localización/recuperación [*discovery*], facilitando búsquedas basadas en campos (p. ej. autor, título), permitiendo la indexación de objetos no textuales, y facilitando el acceso al contenido sustituido/referenciado [*surrogate*] que es distinto del acceso al contenido del propio recurso". (Weibel y Lagoze, 1997)
- En definitiva, el **uso de metadatos** para proporcionar información sobre los recursos Web, **facilita la búsqueda de recursos**, así como la **interoperabilidad semántica** entre ellos.

# Estándares de metadatos

---

- Conjunto de componentes con nombre (elementos, términos,...) junto con una cierta documentación que indica su significado y uso, para poder definir metadatos.
- Especificación de cómo representar una instancia de metadatos en un formato digital
- Estándares de metadatos más destacados:
  - DC (uso para todo tipo de recursos)
  - FOAF (se utiliza principalmente para personas)
  - LOM (descripción de objetos educativos)

# Iniciativa de Metadatos Dublin Core (DCMI)

---

- **DCMI** se inicio en 1995 gracias a la colaboración de bibliotecarios, investigadores de librerías digitales, proveedores de contenidos y expertos en lenguajes de marcado.
  - pretendieron desarrollar estándares para describir los recursos de información y facilitar su recuperación.
  - Así nació un pequeño conjunto de descriptores de documentos y recursos.
- En la actualidad, los metadatos Dublin Core se han convertido en uno de los estándares más extendidos para la recuperación de información en la Web.
  - no sólo en el ámbito bibliotecario y documental, sino en otros muchos sectores.
- Dichos metadatos se puede utilizar con HTML, con XML y conjuntamente con otros lenguajes de descripción como RDF.
  - **RDF**: Es el modelo de descripción de metadatos desarrollado por el consorcio W3C.
  - DCMI reconoce la importancia potencial de RDF para la definición, mantenimiento y desarrollo del Dublin Core.
- *Schemas* RDF que se usan en proyectos o productos que usan metadatos DCMI:
  - *Schema* que define los términos para el Dublin Core Simple (los 15 elementos originales): (<http://purl.org/dc/elements/1.1/>)
  - *Schema* que define los elementos de matización, esquemas [*schemes*] de codificación y otros elementos: (<http://purl.org/dc/terms/>)
  - *Schema* que define el vocabulario del elemento *type* del Dublin Core: (<http://purl.org/dc/dcmitype/>)

# Elementos del Dublín Core (DC)

---

- El más importante es el que contiene los 15 elementos originales, ya que, son los más utilizados en la práctica (dichos elementos se convirtieron en norma **ISO 15836/2003** en febrero de 2003)
- Se indica en el espacio de nombres de la siguiente forma:  
**xmlns:dc="http://purl.org/dc/elements/1.1/"**
- Dichos elementos se pueden clasificar en las siguientes tres categorías, en función de lo que quieren indicar:
  1. **Elementos relacionados principalmente con el contenido del recurso**
    - dc:title, dc:subject, dc:description, dc:source, dc:language, dc:relation, dc:coverage
  2. **Elementos relacionados principalmente con el recurso cuando es visto como una propiedad intelectual**
    - dc:creator, dc:publisher, dc:contributor, dc:rights
  3. **Elementos relacionados principalmente con la instanciación del recurso**
    - dc:date, dc:type, dc:format, dc:identifier

# Elementos relacionados con el contenido del recurso

---

- **title**: El nombre del recurso.
- **subject**: palabra o frase clave que describe el contenido del recurso.
- **description**: Una descripción textual del recurso, siendo un resumen en el caso de un documento, o una descripción del contenido en el caso de una imagen.
- **source**: Secuencia de caracteres utilizado para identificar unívocamente otro recurso, a partir del cual proviene el recurso actual.
- **language**: Idioma del contenido intelectual del recurso. Es recomendable la utilización de los códigos definidos según la ISO 639-2, por ejemplo, es (castellano), en (inglés),...
- **relation**: Una referencia a otro recurso con el que está relacionado el recurso actual.
- **coverage**: La característica de cobertura espacial (nombre o coordenadas de un lugar), temporal (fecha o periodo temporal) o jurisdiccional (nombre de una entidad administrativa) del contenido intelectual del recurso.

# Elementos relacionados con el recurso visto como una propiedad intelectual

---

- **creator**: La persona u organización responsable de la creación del contenido intelectual del recurso.
- **publisher**: La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual.
- **contributor**: Una persona u organización que haya tenido una contribución intelectual significativa en la creación del recurso pero cuyas contribuciones son secundarias en comparación a las de las personas u organizaciones especificadas en el elemento creator.
- **rights**: Una referencia (URL, por ejemplo) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso.

# Elementos relacionados con la instanciación del recurso

---

- **date**: Una fecha de un evento en el ciclo de vida del recurso actual. Normalmente se referirá a la fecha de creación o cuando se puso a disposición del usuario en su forma actual. Para el formato utilizado para las fechas se recomienda seguir el que se indica en la ISO 8601.
- **type**: La función o el tipo de contenido del recurso. Para indicarlo se recomienda el uso de un vocabulario controlado, como por ejemplo el que se define en el esquema <http://purl.org/dc/dcmitype/>, en donde se tienen los términos asociados a sonido, imagen, texto, software, servicio, ... Para definir el formato físico o digital del recurso se utiliza el elemento “format”.
- **format**: El formato de datos de un recurso, usado para identificar el software y posiblemente, el hardware que se necesitaría para mostrar el recurso. Normalmente, se suele indicar el tipo de medio o el tamaño del recurso actual.
- **identifier**: Secuencia de caracteres usados para identificar unívocamente un recurso. Posibles ejemplos serían para un recurso en línea utilizar su URL o URN, para un libro usar su ISBN, ...

# Extensiones de Dublin Core

---

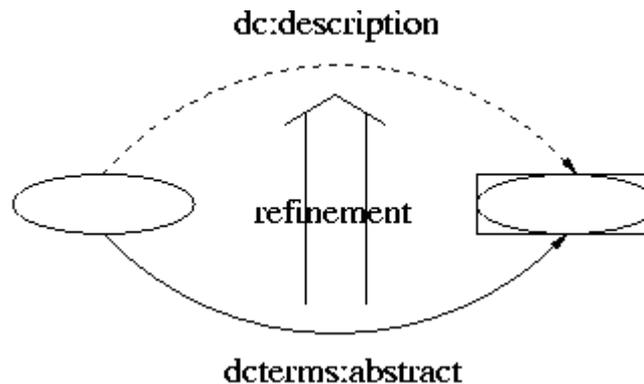
- Cuando nos encontramos con un metadato asociado al elemento dc:date, ¿qué tipo de significado se le quiere dar a dicha fecha? ¿es la fecha de creación? ¿la de modificación?
- La utilización de vocabularios controlados puede ser más efectiva utilizando **cualificadores** (“Dublin Core Qualifiers”).
  - Sin un esquema [*scheme*] de codificación designado específicamente, una materia, que puede haber seleccionado cuidadosamente de un vocabulario controlado, puede no distinguirse de un simple descriptor.
- DCMI ha definido dos clases de cualificadores:
  - **De refinamiento:** Dan un significado más específico a los elementos.
  - **De codificación:** Identifican esquemas que ayudan a la interpretación del valor de un elemento. Dichos esquemas pueden ser vocabularios controlados, notaciones formales o reglas de análisis sintáctico [parsing rules].
- Dichos cualificadores se encuentran definidos en el esquema <http://purl.org/dc/terms/>

# Cualificadores de refinamiento

- Haciendo uso de la propiedad *rdfs:subPropertyOf* (instancia de *rdf:Property* que se utiliza para indicar que una propiedad es una especialización de otra), se definen los distintos cualificadores de refinamiento.
- Ejemplo: *abstract* (resumen del contenido de un recurso) y *description* (descripción textual del recurso)

```
<rdf:Description rdf:about="http://purl.org/dc/terms/abstract">
```

```
<rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/description"/> </rdf:Description>
```



# Algunos elementos de refinamiento

Elemento DCMI	Cualificador de refinamiento	Descripción del cualificador
dc:title	dcterms:alternative	Alternativa al título formal del recurso. Puede ser una abreviatura o una traducción del título.
dc:description	dcterms:tableOfContents	Lista de las partes que componen el contenido del recurso.
	dcterms:abstract	Resumen del contenido del recurso.
dc:date	dcterms:created	Fecha de creación del recurso.
	dcterms:valid	Periodo de validez del recurso.
	dcterms:available	Periodo de tiempo en que el recurso se encontrará accesible
	dcterms:issued	Fecha de publicación del recurso
	dcterms:modified	Fecha en que el recurso ha sido modificado
	dcterms:dateAccepted	Fecha en que es aceptado el recurso (p.e., un artículo en una revista)
	dcterms:dateCopyrighted	Fecha de comunicado del “copyright”
	dcterms:dateSubmitted	Fecha de presentación del recurso (p.e., una tesis, un artículo,...)
dc:coverage	dcterms:spatial	característica de cobertura espacial (nombre o coordenadas de un lugar)
	dcterms:temporal	característica de cobertura temporal (fecha o periodo temporal)

# Ejercicio en clase

---

Especificar en RDF haciendo uso de DC el siguiente recurso bibliográfico:

- <http://www.wiley.com/legacy/compbooks/daconta/sw/>

# Herramientas de edición y visores de metadatos DC

---

- **“DC-dot”** es un editor de metadatos Dublin Core desarrollado por **Andy Powell** (UKOLN).
  - **UKOLN** es un centro perteneciente a la **Universidad de Bath**, que proporciona asesoramiento y servicios para bibliotecas, centros de educación y , en general, para las comunidades encargadas en **temas culturales, educativos o de información**.
  - Este editor además de crear los metadatos siguiendo el formato del estándar DC, **permite convertir** dichos metadatos **al formato de otros vocabularios** como, por ejemplo, RDF, IEEE LOM, IMS,...
  - Permite visualizar los metadatos generados en **distintos formatos** (HTML, XHTML, XML y RDF), y en **distintos idiomas** (castellano, inglés, alemán, francés, italiano y portugués).
- Otros generadores de metadatos Dublin Core son:
  - **Dublin Core Metadata Gen**
  - **Reggie** (<http://metadata.net/dstc/>)
  - Otras herramientas se pueden encontrar en (<http://es.dublincore.org/tools/>)
- **“Dublin Core Viewer”** es una extensión del navegador Firefox que permite obtener los metadatos DC que se tienen definidos para la página Web que se está visualizando.

# DC-dot



## Dublin Core metadata editor



Se muestran los metadatos en el formato elegido

```
<link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
<meta name="DC.title" content="The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management" />
<meta name="DC.creator" content="Michael C. Daconta" />
<meta name="DC.subject" content="informática" />
<meta name="DC.subject" content="Introducción de conceptos relacionados con la Web" />
```

Permite elegir el formato de salida: XHTML, HTML, XML o RDF

Los metadatos se pueden convertir a formatos de otros estándares, p.e., LOM

[\[how to use the XHTML description\]](#)

If necessary, edit the values in the boxes below, and

Convert metadata to

Display format

Metadata language

**Title**  
The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management

**Creator (author)**  
Michael C. Daconta

**Subject or keywords**  
informática

Cuadros de texto para rellenar los metadatos asociado a cada elemento del DC

# Dublin Core Viewer

The following DC metadata was found:

Element	Content	Scheme
title	Dublin Core Metadata Initiative (DCMI) Home Page	
description	The Dublin Core Metadata Initiative is an open for ...	
date	2005-03-07	
format	text/html	
contributor	Dublin Core Metadata Initiative	
language	en	

Details

The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models. DCMI's activities include consensus-driven working groups, global conferences and workshops, standards liaison, and educational efforts to promote widespread acceptance of metadata standards and practices.

Cuando una página Web contiene metadatos DC se muestra de color naranja el icono en el Navegador "Mozilla Firefox". Tras pulsar en dicho Icono nos aparece el cuadro que contiene dicha información

# Redes Sociales

---

- ¿Cómo encontrar personas:
  - con intereses similares a los de uno?
  - relacionadas a alguien?
  - autoridades en un tema?
  - etc.
- Redes sociales:
  - Información personal en la Web semántica.
  - Enlaces a información personal de personas conocidas.
- Solución: Uso de FOAF

# ¿Qué es FOAF?

---

- FOAF (“Friend of a Friend”) es un proyecto de Web Semántica, que permite la creación y el uso de páginas Web por personas y grupos, con la finalidad de tener disponible información personal de forma sencilla y simplificada.
- Entre la información que se puede encontrar se puede destacar la descripción de las personas, los vínculos que hay entre ellos y aquello en lo que trabajan y hacen.
- Se trata de una aplicación XML y RDF iniciada por Edd Dumbill, con el **propósito** de realizar documentos en los que lo que importe no sea lo que se conoce sino, **a quién se conoce**.
- FOAF depende, en gran medida, de ciertos estándares de W3C, como XML, XML Namespaces, RDF y OWL. Por ello, los documentos FOAF deben ser documentos RDF/XML bien formados.
- La idea básica es la de cualquier proyecto de Web Semántica, es decir, **si se publica información en documentos con el formato FOAF, entonces las máquinas serán capaces de usar dicha información** con los fines deseados por el usuario.

# Elementos de FOAF

---

- El espacio de nombres asociado al proyecto FOAF es:  
xmlns:foaf="http://xmlns.com/foaf/0.1/"
- Clases más destacadas:
  - **foaf:Agent** (representa la clase de los agentes, p.e., persona, grupo, software o algo físico).
    - **foaf:Organization** (instituciones sociales)
    - **foaf:Person** (representa a las personas individualmente)
    - **foaf:Group** (se refiere a una agrupación de agentes)
    - La diferencia principal entre las clase **foaf:Organization** y **foaf:Group** es que la primera se refiere a agrupaciones de personas más sólidas (empresas, instituciones,...), mientras que la segunda indica agrupaciones creadas ad-hoc, con lo que, pueden considerarse menos duraderas o consistentes .
  - **foaf:Document** (representa todo tipo de documentos físicos o electrónicos)
    - **foaf:Image** (aquellos documentos que son imágenes: JPG, diagramas SVG,...)
  - **foaf:Project** (representa un proyecto colectivo)

# Clase foaf:Person (I)

---

- Todas las propiedades y las clases que se describen a continuación (se verán resaltadas en negrita), tiene como dominio la clase **foaf:Person**, y pertenecen al espacio de nombres `xmlns:foaf="http://xmlns.com/foaf/0.1/"`.
- **foaf:family\_name**
- **foaf:firstName** se utilizan para indicar
- **foaf:surname** el nombre de la persona
- **foaf:givenname**
- **foaf:img** relaciona una persona con una de sus imágenes, que será una instancia de la clase **foaf:Image**.
- **foaf:workInfoHomepage** indica el trabajo que realiza la persona dentro de la organización.
- **foaf:workplaceHomepage** indica la página de la organización en la que trabaja.
- **foaf:schoolHomepage** indica la página Web de la escuela en la que está cursando o curso sus estudios. Se puede incluir en este tipo tanto el instituto, la universidad, ...

# Clase foaf:Person (II)

---

- **foaf:page** relaciona algo con una página que lo trata. Su inversa es **foaf:topic**.
- **foaf:homepage** es una subpropiedad de la propiedad **foaf:page**, que relaciona algo con un documento Web público.
- Propiedades que indican proyectos en los que ha colaborado o ha realizado individualmente:
  - Anteriormente realizados: **foaf:pastProject**
  - En realización: **foaf:currentProject**
- **foaf:publications** indica una lista de documentos asociados a dicha persona. Algunos normalmente se encontrarán en su página del trabajo (**foaf:homepage**).
- **foaf:interest** representa el interés, por parte de un agente, hacia un **foaf:Document**, debido al tema que trata (**foaf:topic**).
- **foaf:plan** representa cualquier cosa que tiene planeado hacer una persona.
- **foaf:knows** indica que dicha persona conoce a otra persona. Es una de las propiedades más destacadas, ya que, mediante la mención de otras personas en el **foaf:knows** y el uso del enlace **rdfs:seeAlso**, se facilita la tarea a las herramientas de búsqueda de los FOAF de las diferentes personas.

# Otras clases y propiedades

---

- Clase **foaf:Image**
  - **foaf:thumbnail**: Relaciona una imagen y otra más grande, a la que hace referencia.
  - **foaf:depicts**: Es una relación entre una **foaf:Image** y algo que la imagen representa.
    - Su inversa es **foaf:depiction**. El uso de ésta es principalmente indicar el contenido de una imagen digital, por ejemplo, las personas o los objetos que se encuentran representados en un conjunto de fotografías.
- Clase **foaf:Group**
  - **foaf:member** indica los miembros que pertenecen a un grupo.

# Clase foaf:Agent

- Como superclase de la clase **foaf:Person** viene, principalmente, a cubrir aquellos aspectos que no puede hacerse simplemente con dicha clase. Es por ello que aparecen las siguientes propiedades:
- **foaf:birthday** indica la fecha de creación/nacimiento del Agente al que está asociado.
- **foaf:made** relaciona al agente con algo que ha hecho. Es la inversa de **foaf:maker**.
- **foaf:mbox** y **foaf:mbox\_sha1sum** representan la relación entre el propietario de una dirección de correo electrónico y ésta.
  - La diferencia entre ambas propiedades es que la segunda es una cadena obtenida a partir de la aplicación de la función matemática SHA1 a un identificador (URI) 'mailto:', con lo que, se utilizará si no se desea hacer visible la dirección a todo el mundo, mientras que si este hecho nos es indiferente, podemos utilizar la propiedad **foaf:mbox**.
- **foaf:nick** relaciona un agente, normalmente una persona, con un identificador que lo representa. FOAF proporciona una serie de propiedades, que asignan al agente un identificador de texto para cada uno de los siguientes sistemas de mensajes/Chat:

Propiedad	Sistema de mensajes / Chat
<b>foaf:jabberID</b>	Jabber
<b>foaf:aimChatID</b>	AIM de AOL
<b>foaf:icqChatID</b>	Chat ICQ
<b>foaf:yahooChatID</b>	Chat Yahoo
<b>foaf:msnChatID</b>	Chat MSN

Por como están definidas dichas propiedades se consideran todas ellas como propiedades de dominio único ('inverse functional property').

# Ejemplo

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:airport="http://www.daml.org/2001/10/html/airport-ont#"
  xmlns:wn="http://xmlns.com/wordnet/1.6/" xmlns:pos="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <rdf:Description rdf:about="">
    <dc:title>FOAF for J. Javier Samper</dc:title>
    <dc:description>Friend-of-a-Friend description for J. Javier Samper Zapater</dc:description>
    <dc:creator rdf:resource="#javis" />
  </rdf:Description>
  <foaf:Person rdf:ID="javis">
    <foaf:name>Jose Javier Samper Zapater</foaf:name>
    <foaf:title>Dr.</foaf:title>
    <foaf:firstName>Jose Javier</foaf:firstName>
    <foaf:surname>Samper</foaf:surname>
    <foaf:nick>javi</foaf:nick>
    <foaf:homepage rdf:resource="http://robotica.uv.es/~jsamper/" />
    <foaf:mbox rdf:resource="mailto:jose.j.samper@uv.es" />
    <foaf:phone rdf:resource="tel:+34-96-354-3567" />
    <foaf:depiction rdf:resource="http://alba.uv.es/javi/fotoUni1.bmp" />
    <foaf:location>
      <foaf:Geo rdf:value="Burjassot, Valencia Spain" />
    </foaf:location>
    <foaf:knows rdf:resource="#leo" />
  </foaf:Person>
</rdf:RDF>
```

# Foaf Explorer

**Foaf Explorer: Jose Javier Samper Zapater**

Location:

[ Validate RDF | Disclaimer | Privacy ]  
Bookmarklet: [Foaf Explorer](#)

[FOAF for J. Javier Samper]



**Jose Javier Samper Zapater**  
[ foafnaut rdfweb web view ]

**Knows:**

- Leo van den Berg **knows (Person):**  
[Jose Javier Samper Zapater](#)  
[ foafnaut rdfweb web view ]

**publications (seeAlso):**  
**Resource:**  
<http://robotica.uv.es/~jsamper/#public>

**past project:**  
**Resource Identifier:**  
[FOAF for J. Javier Samper](#)

**Creator (Person):**  
[Jose Javier Samper Zapater](#)  
[ foafnaut rdfweb web view ]

**RacerPro for Semantic Web**  
Fast and scalable reasoner on Complex ontology & OWL data  
[Ads by Google](#)

**Acord XML Integration**  
Build quickly any processing logic for Acord XML integration  
[Advertise on this site](#)

Internet

Inicio | Explorador ... | TemaWeb3G.ppt | websemanticasi... | claudio.ppt | AtomicProcess ... | Foaf Explorer: ... | http://robotica... | Eudora - [].vdb... | ES | 13:19

# Foaf Explorer

**Description:**  
The Robotics Institut is a laboratory of the University of Valencia, integrated within the same one. Most of its members belong to the Department of Computer science. The activities that are developed turn around telematics applications in diverse areas, which are headed by groups of investigation within the Robotics Institute:

**nearlest airport** [GeoURL](#) [Google](#) [MapQuest](#) [Airpoint](#):  
**sameIndividualAs:**  
<http://www.daml.org/cgi-bin/airport?VLC>

**seeAlso:**  
[Foaf Explorer for Valencia](#)

**elevation:**  
225

**longitude:**  
-0.4813888888888889

**latitude:**  
39.48916666666667

**icaoCode:**  
LEVC

**iataCode:**  
VLC

**location:**  
Valencia, Spain

**name:**  
Valencia

**based near** [GeoURL](#) [Google](#) [MapQuest](#) [Point](#):  
**longitude:**  
-0.41024

**latitude:**  
39.5106

**location (Geo):**  
**value:**  
Burjassot, Valencia Spain

**event (Birth):**  
**place:**  
Puerto Sagunto, Valencia, Spain

**date:**  
1968-03-19

**depiction:**  
<http://alba.uv.es/javi/fotoUnit1.bmp>

**phone:**  
<tel:+34-96-354-3567>

**Referenced Vocabularies**

- <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
The RDF Vocabulary (RDF)
- <http://xmlns.com/foaf/0.1/>
- <http://www.w3.org/2000/01/rdf-schema#>
- <http://purl.org/dc/elements/1.1/>  
The Dublin Core Element Set v1.1 namespace providing access to

# MapQuest desde FOAF

# Publicación de datos FOAF

---

- ¿Cómo mis datos FOAF pueden ser conocidos por otros usuarios?
  - Soy “conocido” por alguien “conocido”:
    - Vía “foaf:knows”
  - Publicando mi info en **FOAF Bulleting Board**:
    - Sitio con enlaces a decenas de archivos FOAF
  - Incluyendo en mi sitio Web un enlace a mi archivo FOAF:
    - `<link rel="meta" type="application/rdf+xml" title="FOAF" href="foaf.rdf" />`

# Herramientas para FOAF

---

- FOAF Explorer
  - Vista HTML de FOAF
- FOAFNaut
  - Visualización de relaciones entre personas
- Plink
  - Sitio de redes sociales
- FOAFBot, Interfaces IRC sobre FOAF

# LOM (“Learning Object Metadata”)

---

- Estándar que especifica Metadatos para Objetos Educativos.
- Desarrollado desde 1998 por IEEE Learning Technology Standards Committee (LTSC).
- Para este Estándar, un **objeto educativo** se define como cualquier entidad, digital o no, susceptible de ser usada en aprendizaje, educación o formación.
  - Una instancia de metadatos para un objeto educativo describe sus características relevantes.
  - Dichas **características se pueden agrupar** en las siguientes **categorías:** *general, ciclo de vida, meta-metadatos, técnica, uso educativo, derechos, relación, anotación y clasificación.*
  - **Dentro de cada categoría hay una jerarquía de elementos de datos** a los que se asignan los valores de metadatos.

# LOM (“Learning Object Metadata”)

---

- Propósito de este estándar:
  - **Facilitar la búsqueda, evaluación, adquisición y uso de los objetos educativos**, por ejemplo, por alumnos, profesores o procesos automáticos de software.
  - **Facilitar el intercambio y uso compartido de objetos educativos**, permitiendo el desarrollo de catálogos
  - En definitiva, un **alto grado de interoperabilidad semántica** en temas de educación y formación.
- Existen dos fundaciones muy involucradas en el fomento del uso del estándar LOM:
  - **ARIADNE [ARI]** es un proyecto europeo que persigue el desarrollo de herramientas y metodologías para la producción, gestión y almacenamiento de objetos educativos basados en el estándar LOM.
  - **IMS Global Learning Consortium [IMS]** es un consorcio dedicado al desarrollo de metadatos para temas educativos
    - Ha desarrollado una guía de implementación y buen uso del estándar LOM, así como un XML schema para LOM.

# Elementos del estándar LOM (I)

---

- La categoría **General** agrupa la información general que describe este objeto educativo en su conjunto. Dentro de esta categoría se encuentran, entre otros, los siguientes elementos:
  - **Identifier** (identificador): Una etiqueta, única que identifica este objeto educativo. Dicho identificador puede ser cualquiera de los siguientes elementos:
    - **Catalog** (catálogo): El nombre o denominación del esquema de identificación o catalogación para esta entrada. Un esquema de espacio de nombres. Ejemplos: “ISBN”, “URI”
    - **Entry** (entrada): El valor del identificador dentro del esquema de identificación o catalogación que designa o identifica este objeto educativo. Una cadena específica de un espacio de nombres. Ejemplos: “2-7342-0312”, “http://www.ieee.org/documents/12”
  - **Title** (título): El nombre asignado a este objeto educativo. Ejemplos: (“en”, “The life and works of Leonardo daVinci”), (“es”, “La vida y obra de Leonardo daVinci”).
  - **Language** (idioma): El idioma o idiomas humanos predominantes en este objeto educativo para la comunicación con el usuario. Ejemplos: “en”, “es”.
  - **Description** (descripción): Una descripción textual del contenido de este objeto educativo.
  - **Keyword** (palabra clave): Una palabra clave o frase que describe el tema principal del objeto educativo.
  - **Coverage** (ámbito): La época, cultura, zona geográfica o región a la que es aplicable este objeto educativo.

# Elementos del estándar LOM (II)

- La categoría **LifeCycle** (*Ciclo de Vida*) agrupa las características relacionadas con la historia y el estado actual del objeto educativo, y aquellas que le han afectado durante su evolución. Dentro de esta categoría se encuentran, entre otros, los siguientes elementos:
  - **Contribute** (contribución): Aquellas entidades (personas u organizaciones) que han contribuido al estado de este objeto educativo a lo largo de su ciclo de vida (por ejemplo, creación, edición, publicación). Dicha contribución se puede indicar mediante los siguientes elementos:
    - **Role** (tipo): Tipo de contribución (autor, desconocido, iniciador, terminador, revisor, editor, diseñador gráfico, desarrollador técnico, proveedor de contenidos, revisor técnico, revisor educativo, guionista, diseñador educativo o experto en la materia)
    - **Entity** (entidad): La identificación e información de las entidades (personas u organizaciones) que han contribuido a este objeto educativo. Se toma como espacio de valores vCard, tal y como se define en el IMC vCard 3.0 (RFC 2425, RFC 2426)
    - Ejemplo “BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area Administrator\nAssistant\nEMAIL\n;TYPE=INTERNET\nNet:jfriday@host.com\nEND:VCARD\n”
    - **Date** (fecha): La fecha de la contribución
- La categoría **Technical** (técnica) describe los requisitos y características técnicas de este objeto educativo. Dentro de esta categoría se encuentran, entre otros, el siguiente elemento:
  - **Format** (formato): El(los) tipo(s) de datos de (todos los componentes) este objeto educativo. Este elemento de datos debe ser utilizado para identificar el software necesario para acceder al objeto educativo. Como espacio de valores se deben utilizar Tipos MIME basado en el registro IANA (ver RFC2048) o 'no-digital'.
    - Ejemplos: “video/mpeg”, “application/x-toolbook”, “text/html”
- Como se puede apreciar existe una relación evidente en algunos de los elementos definidos en el estándar LOM y los elementos que se han descrito para la iniciativa DCMI. Veamos cual es el “mapeo” que existe entre ambos.

# Ejemplo: DC vs LOM

Veamos un ejemplo similar al visto en la sección del DCMI, pero en esta ocasión utilizando los elementos del estándar LOM:

```
<?xml version="1.0" encoding="UTF-8" ?>
<lom xmlns="http://ltsc.ieee.org/xsd/LOMv1p0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ltsc.ieee.org/xsd/LOMv1p0 http://www.rdn.ac.uk/oai/lom/lom.xsd">
  <general>
    <title>
      <string language="en">The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management</string>
    </title>
    <language>en</language>
    <description>
      <string language="en">Introducción de conceptos relacionados con la Web Semántica</string>
    </description>
    <keyword>
      <string language="en">informática</string>
    </keyword>
  </general>
  <lifeCycle>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>contributor</value>
      </role>
      <entity>BEGIN:vCard FN:Michael C. Daconta END:vCard</entity>
    </contribute>
  </lifeCycle>
  <technical />
  <educational />
</lom>
```

# Ejemplo: documento XML del estándar LOM

```
<?xml version="1.0" encoding="UTF-8" ?>
<lom xmlns="http://ltsc.ieee.org/xsd/LOMv1p0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ltsc.ieee.org/xsd/LOM
    http://ltsc.ieee.org/xsd/lomv1.0/20040413/lom.xsd ">
  <general>
    <title>
      <string>developerWorks : XML</string>
    </title>
    <description>
      <string> The XML zone on the developerWorks Web site is designed for developers. You'll find tools, samples,
      events, and links to XML community forums and Web sites.
    </string>
    </description>
    <keyword>
      <string>xml resources</string>
    </keyword>
    <keyword>
      <string>xml programming</string>
    </keyword>
  </general>
  <technical>
    <format>text/html</format>
    <location>http://www-106.ibm.com/developerworks/xml/</location>
  </technical>
  <educational>
    <learningResourceType>
      <source>DCMIType</source>
      <value>Text</value>
    </learningResourceType>
  </educational>
</lom>
```

standards information, education, news and

# Otros vocabularios para metadatos

---

- **RELATIONSHIP**

- Un vocabulario para describir relaciones entre personas.
- Espacio de nombres: (xmlns:rel="http://purl.org/vocab/relationship")
- Este vocabulario puede ser utilizado sin modificarse con documentos FOAF. Simplemente se trata de reemplazar las propiedades foaf:knows por una propiedad más específica de este vocabulario.

- **Basic Geo (WGS84 lat/long) Vocabulary**

- Se trata de un vocabulario RDF básico que proporciona a la comunidad de la Web Semántica, un espacio de nombres para la representación de la latitud, de la longitud y de otra información sobre entidades localizadas en el espacio, mediante el uso de WGS84("World Geodesic System" 1984) como datos de referencia.
- Espacio de nombres: (xmlns:geo="http://www.w3.org/2003/01/geo/wgs84\_pos#")
- Al tratarse de un vocabulario RDF se definen las propiedades de un tipo de cosa: un punto ('Point'), el cual sólo puede tener una latitud('lat'), una longitud('long') y una altitud('alt'), lo cual se puede expresar mediante el lenguaje OWL diciendo que estas propiedades RDF son funcionales.

- **WOT ("Web OfTrust RDF Ontology")**

- Vocabulario diseñado para facilitar el uso de las herramientas de criptografía de clave pública, como por ejemplo PGP o GPG, para firmar documentos RDF y poder documentar estas claves. De esta forma se tendrá la certeza de quién ha realizado las afirmaciones que aparecen en los documentos.
- Espacio de nombres: (xmlns:wot="http://xmlns.com/wot/0.1/")

- **BIO (Vocabulario para Información Biográfica)**

- Vocabulario para la descripción de información biográfica de personas.
- Espacio de nombres: (xmlns:bio="http://purl.org/vocab/bio/0.1/")

# Ejemplo: FOAF + Otros vocabularios

```
<foaf:Person rdf:nodeID="juanito">
  <foaf:name>Juan Pérez</foaf:name>
  <foaf:title>Mr</foaf:title>
  <foaf:firstName>Juan</foaf:firstName>
  <foaf:surname>Pérez</foaf:surname>
  <rel:hasMet>
    <foaf:Person>
      <foaf:name>Pedro Rodriguez</foaf:name>
    </foaf:Person>
  </rel:hasMet>
  <rel:collaboratesWith rdf:nodeID="javii" />
  <foaf:based_near>
    <geo:Point>
      <geo:placename>London</geo:placename>
      <geo:region>GB-LND</geo:region>
      <geo:lat>51.5255</geo:lat>
      <geo:long>-0.081</geo:long>
    </geo:Point>
  </foaf:based_near>
  <foaf:mbox_sha1sum>a714a83db77c6ce85211beb56599adf2c4eaa62f</foaf:mbox_sha1sum>
  <wot:hasKey rdf:nodeID="KeyA" />
  <bio:event>
    <bio:Marriage>
      <bio:date>1990-03-03</bio:date>
      <bio:place>Northampton Register Office, Northampton, Northamptonshire, United Kingdom
      </bio:place>
    </bio:Marriage>
  </bio:event>
</foaf:Person>
<wot:PubKey rdf:nodeID="KeyA"/>
```

# Ejercicios de uso de metadatos

---

- **Uso de las herramienta DC-dot [DC-dot] o Dublin Core Metadata Gen [DC-Gen]:**
  - Crear un documento en distintos formatos (XML, XHTML, RDF) que contenga una descripción de algún recurso mediante metadatos DC.
  - Crear un documento XML describiendo el mismo recurso, pero en este caso utilizando el estándar LOM.
- **Uso de FOAF-a-matic [FOAF-a-matic] y FOAF Explorer [FOAF Explorer].**
  - Vuestro CV.

# Tema 4, 1ª parte: OWL

## “Lenguajes de marcado ontológico”

# Objetivos de esta parte

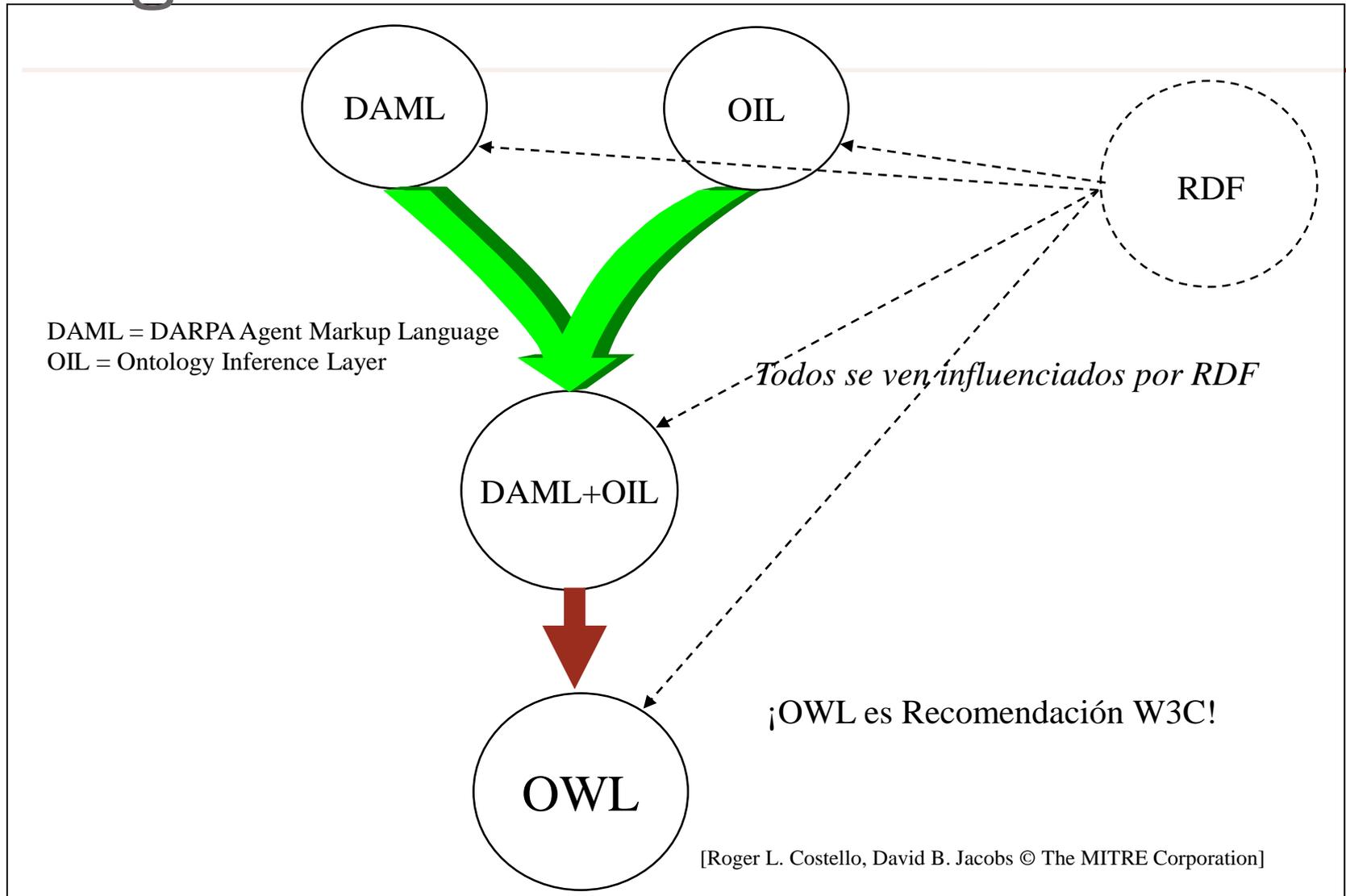
---

- Comprender las limitaciones de los lenguajes RDF y RDF Schema.
- Comprobar la utilidad del lenguaje OWL.
- Dar a conocer algunas herramientas que se utilizan en el proceso de creación de ontologías: editores y las máquinas de inferencia de conocimiento (razonadores).
- Presentar los lenguajes de consulta más utilizados.
- Introducir las distintas herramientas que se utilizarán en la práctica, para la creación y el mantenimiento de las ontologías.

# ¿Qué es OWL?

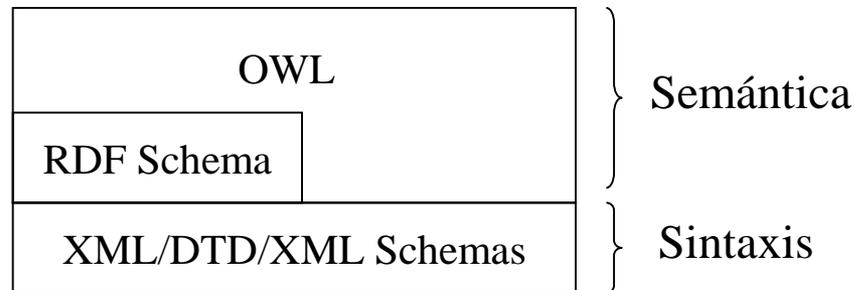
- OWL (“Web Ontology Language”) tiene como objetivo facilitar un modelo de marcado, construido sobre RDF y codificado en XML, que nos permita representar ontologías a partir de un vocabulario más amplio y una sintaxis más fuerte que la permitida por RDF.
- OWL se utiliza para representar el significado de términos pertenecientes a un vocabulario y definir las relaciones que existen entre ellos.
  - Es útil cuando la información retenida en las necesidades de los documentos debe ser procesada por las aplicaciones.
  - Posibilita un nivel avanzado de inferencia, mediante el uso de herramientas de razonamiento (obtienen conocimiento no explícito a partir de la información que se tiene).
- Tiene como base las experiencias previas realizadas con DAML-OIL.

# Origen de OWL



# Relación con RDF Schema

- Extiende RDFS para definir/relacionar ontologías de una forma más precisa:
  - Posibilita un nivel avanzado de inferencia para la Web Semántica
  - Permite:
    - definir y restringir clases (conceptos).
    - definir y restringir propiedades.
    - definir relaciones entre clases.
- OWL utiliza ciertas características definidas en el lenguaje RDF Schema como, por ejemplo, rdfs:range, rdfs:domain, rdfs:subPropertyOf y rdfs:subClassOf.



# Sublenguajes de OWL

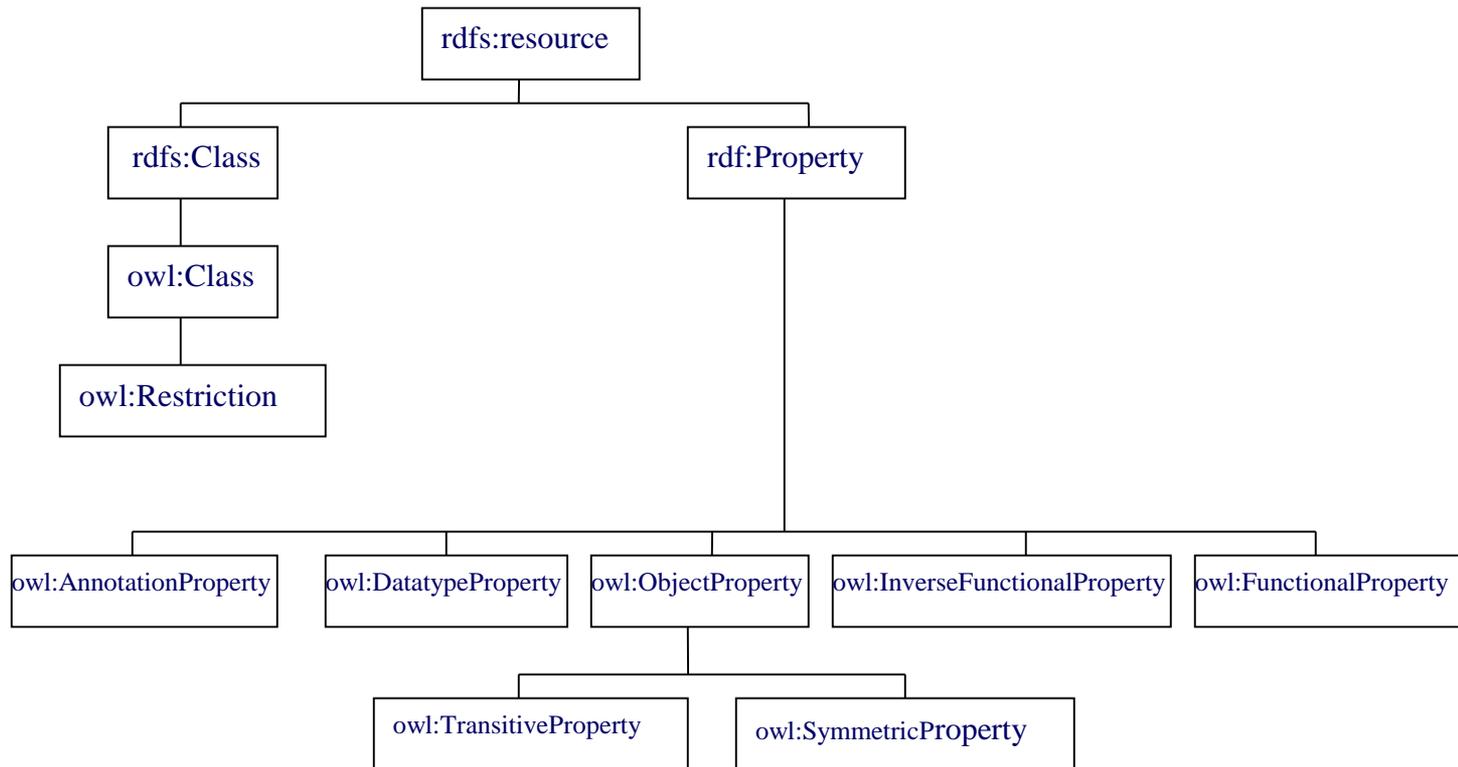
- **OWL-Full:** Contiene todas las construcciones del lenguaje OWL y, además, permite una total libertad en el uso de construcciones RDF.
  - No se garantiza la eficiencia ni siquiera la decidibilidad (“*decidability*”)
- **OWL-DL:** Limita la expresividad para conseguir que sea un lenguaje decidable.
  - Se basa en la Lógica Descriptiva (“Description Logic”).
  - No se pueden utilizar construcciones RDF.
- **OWL-Lite:** Subconjunto de OWL DL más fácil de implementar.
  - Se utiliza básicamente para clasificación jerárquica y restricciones simples.



# Elementos básicos de OWL

- Clases e Instancias.
- Propiedades.
- Especificaciones sobre la definición de propiedades.
- Restricciones sobre el uso de propiedades.

# Taxonomía del lenguaje OWL



# Lista de Propiedades

Property name	domain	range
owl:intersectionOf	owl:Class	rdf:List
owl:unionOf	owl:Class	rdf:List
owl:complementOf	owl:Class	owl:Class
owl:oneOf	owl:Class	rdf:List
owl:onProperty	owl:Restriction	rdf:Property
owl:allValuesFrom	owl:Restriction	rdfs:Class
owl:hasValue	owl:Restriction	<i>not specified</i>
owl:someValuesFrom	owl:Restriction	rdfs:Class
owl:minCardinality	owl:Restriction	xsd:nonNegativeInteger OWL Lite: {0,1} OWL DL/Full: {0,..,N}
owl:maxCardinality	owl:Restriction	xsd:nonNegativeInteger OWL Lite: {0,1} OWL DL/Full: {0,..,N}
owl:cardinality	owl:Restriction	xsd:nonNegativeInteger OWL Lite: {0,1} OWL DL/Full: {0,..,N}
owl:inverseOf	owl:ObjectProperty	owl:ObjectProperty
owl:sameAs	owl:Thing	owl:Thing
owl:equivalentClass	owl:Class	owl:Class
owl:equivalentProperty	rdf:Property	rdf:Property
owl:sameIndividualAs	owl:Thing	owl:Thing
owl:differentFrom	owl:Thing	owl:Thing
owl:disjointWith	owl:Class	owl:Class
owl:distinctMembers	owl:AllDifferent	rdf:List
owl:versionInfo	<i>not specified</i>	<i>not specified</i>
owl:priorVersion	owl:Ontology	owl:Ontology
owl:incompatibleWith	owl:Ontology	owl:Ontology
owl:backwardCompatibleWith	owl:Ontology	owl:Ontology
owl:imports	owl:Ontology	owl:Ontology

[Fuente: “Ontological Engineering”, Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho]

# Sintaxis XML

Las ontologías comienzan por `owl:Ontology`

OWL se basa en RDF (**utiliza sintaxis XML de RDF**).

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.uv.es/ontologiaEjemplo_v2.owl">
  <owl:Ontology rdf:about="http://www.uv.es/ontologiaEjemplo_v2.owl">
    <rdfs:comment>Ejemplo de Ontología</rdfs:comment>
    <owl:priorVersion
      rdf:resource="http://www.uv.es/ontologiaEjemplo_v1.owl"/>
    <owl:imports
      rdf:resource="http://www.uv.es/ontologiaImportada.owl"/>
    <rdfs:label>Primera Ontología</rdfs:label>
  </owl:Ontology>
```

Espacio de nombres del lenguaje owl

La ontología que queremos definir utilizará elementos definidos en la ontología <http://www.uv.es/ontologiaImportada.owl>

¡ `owl:imports` es una propiedad transitiva !

# Clases e Instancias

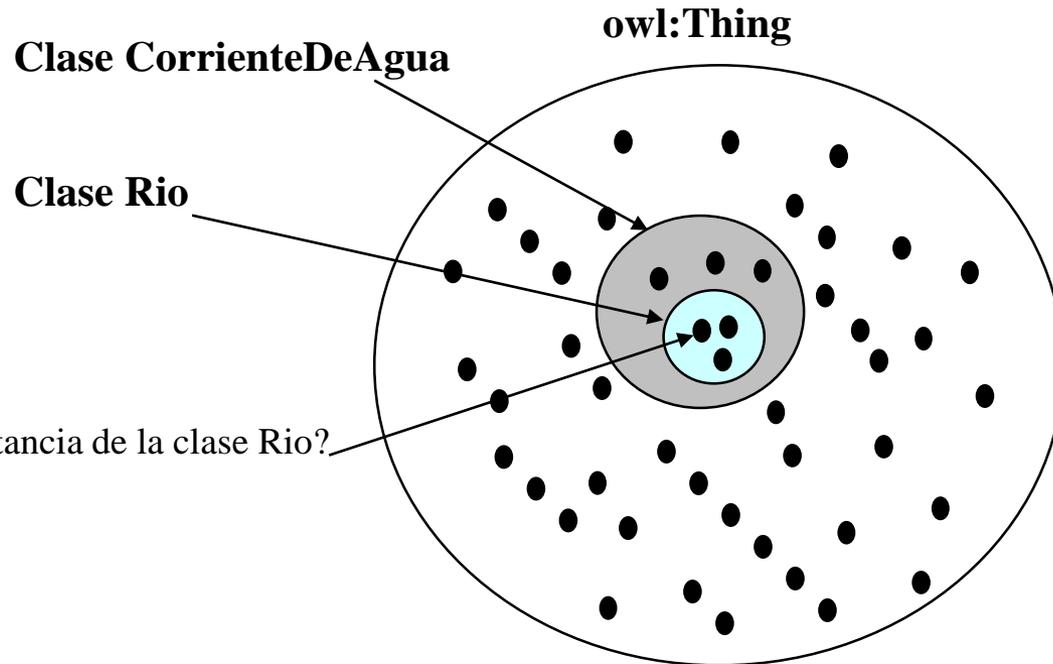
- Para definir los conceptos que se encuentran en una ontología se utilizan las clases OWL (**owl:Class**).
  - **owl:Class** es una subclase de **rdfs:Class**
- Dentro de cada clase se pueden definir instancias (“*individuals*”).
- En OWL se tiene dos clase predefinidas:
  - **owl:Thing** es la clase más general. Todas las instancias forman parte de owl:Thing
  - **owl:Nothing** es la clase vacía

# Clases

¡Definición de una clase de dos formas equivalentes!

```
<owl:Class rdf:ID="Rio">  
  <rdfs:subClassOf rdf:resource="#CorrienteDeAgua"/>  
</owl:Class>
```

```
<owl:Class rdf:ID="Rio">  
  <rdfs:subClassOf rdf:resource="#CorrienteDeAgua"/>  
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>  
</owl:Class>
```



¿Cómo definir una instancia de la clase Rio?

# Instancias

Veamos dos definiciones equivalentes de un río que se denomina Yangtze:

```
<?xml version="1.0"?>
<Rio rdf:ID="Yangtze"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns="http://www.geodesy.org/water/naturally-occurring#">
</Rio>
```

```
<?xml version="1.0"?>
<owl:Thing rdf:ID="Yangtze"
            xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:owl="http://www.w3.org/2002/07/owl#"
            xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <rdf:type rdf:resource="http://www.geodesy.org/water/naturally-occurring#Rio"/>
</owl:Thing>
```

"Yangtze es una cosa (**owl:Thing**). En particular, Yangtze es un **Rio**."

# Tipos de Propiedades

- Las propiedades en OWL pueden ser de dos tipos:
  - **ObjectProperty**: relacionan un recurso con otro.
  - **DatatypeProperty**: relaciona una instancia de una clase con un cierto valor (un RDFS Literal o un tipo de dato XML Schema).
- Para su definición se pueden utilizar las propiedades [rdfs:range](#), [rdfs:domain](#) y [rdfs:subPropertyOf](#).

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.uv.es/ontologias/ontologiaAgua.owl">
  <owl:ObjectProperty rdf:ID="desembocaEn">
    <rdfs:domain rdf:resource="#Rio"/>
    <rdfs:range rdf:resource="#CuerpoDeAgua"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="longitud">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
  </owl:DatatypeProperty>
  ...
</rdf:RDF>
```

# Características de Propiedad

- **Transitividad:**
  - Si  $P(x,y)$  y  $P(y,z)$  entonces  $P(x,z)$
  - owl:TransitiveProperty
- **Simetría:**
  - Si  $P(x,y)$  entonces  $P(y,x)$
  - owl:SymmetricProperty
- **Propiedad Funcional:**
  - Si  $P(x,y)$  y  $P(x,z)$  entonces  $y = z$
  - owl:FunctionalProperty
- **Propiedad Inversa:**
  - $P$  es inversa de  $Q \Rightarrow [ P(x,y) \Leftrightarrow Q(y,x) ]$
  - owl:inverseOf
- **Propiedad Funcional Inversa:**
  - Si  $P(x,y)$  y  $P(z,y)$  entonces  $x = z$
  - owl:InverseFunctionalProperty

# TransitiveProperty

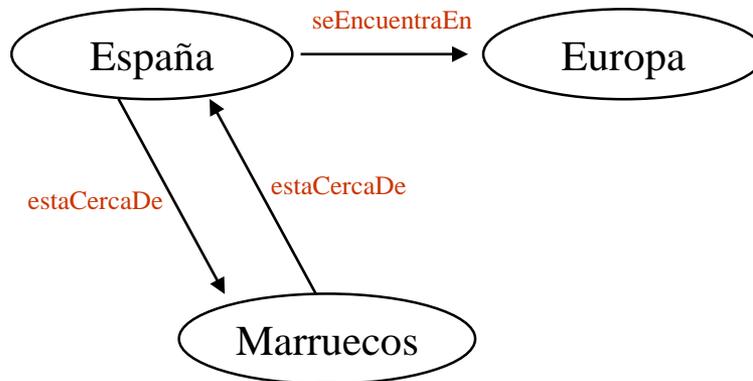
```
<owl:ObjectProperty rdf:ID="seEncuentraEn">  
  <rdf:type rdf:resource="&owl;TransitiveProperty" />  
  <rdfs:domain rdf:resource="&owl;Thing" />  
  <rdfs:range rdf:resource="#Region" />  
</owl:ObjectProperty>  
<Region rdf:ID="Valencia">  
  <seEncuentraEn rdf:resource="#España" />  
</Region>  
<Region rdf:ID="España">  
  <seEncuentraEn rdf:resource="#Europa" />  
</Region>
```



# SymmetricProperty

```
<owl:ObjectProperty rdf:ID="estaCercaDe">  
  <rdf:type rdf:resource="&owl;SymmetricProperty" />  
  <rdfs:domain rdf:resource="#Region" />  
  <rdfs:range rdf:resource="#Region" />  
</owl:ObjectProperty>
```

```
<Region rdf:ID="España">  
  <seEncuentraEn rdf:resource="#Europa"/>  
  <estaCercaDe rdf:resource="#Marruecos"/>  
</Region>
```



# FunctionalProperty (I)

Añadimos a la propiedad “desembocaEn”, la característica “owl:FunctionalProperty”

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.uv.es/ontologias/ontologiaAgua.owl">
  <owl:ObjectProperty rdf:ID="desembocaEn">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#Rio" />
    <rdfs:range rdf:resource="#CuerpoDeAgua" />
  </owl:ObjectProperty>
  ...
</rdf:RDF>
```

# FunctionalProperty (II)

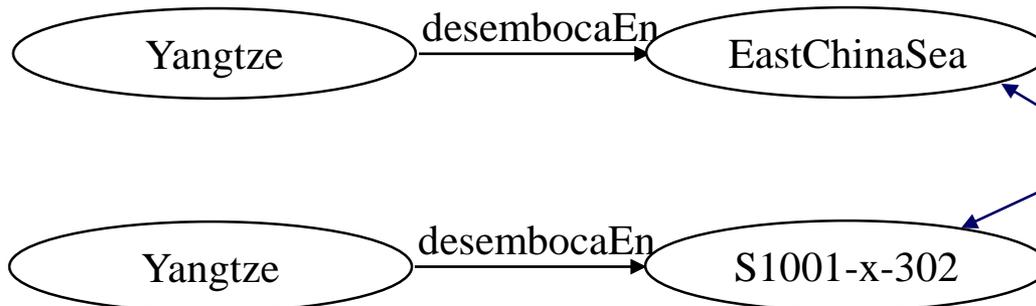
Supongamos que hay dos documentos que describen el rio Yangtze:

```
<?xml version="1.0"?>
<Rio rdf:about="http://www.china.org/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.uv.es/ontologias/ontologiaAgua.owl#">
  <desembocaEn rdf:resource="http://www.china.org/geography#EastChinaSea"/>
</Rio>
```

doc1.rdf

```
<?xml version="1.0"?>
<Rio rdf:about="http://www.china.org/rivers#Yangtze"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <desembocaEn rdf:resource="http://www.national-geographic.org#S1001-x-302"/>
</Rio>
```

doc2.rdf



Se infiere que estos dos valores se refieren a lo mismo

[Roger L. Costello, David B. Jacobs © The MITRE Corporation]

# inverseOf

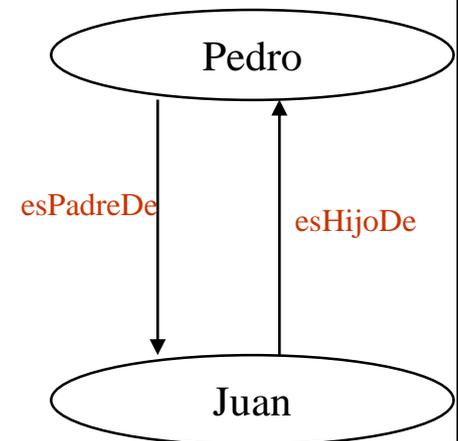
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.uv.es/ontologias/familia.owl">
  <owl:ObjectProperty rdf:ID="esHijoDe">
  <owl:ObjectProperty rdf:ID="esPadreDe">
    <owl:inverseOf rdf:resource="#esHijoDe" />
    <rdfs:domain rdf:resource="#Persona"/>
    <rdfs:range rdf:resource="#Persona"/>
  </owl:ObjectProperty>
```

...

```
</rdf:RDF>
```

```
<?xml version="1.0"?>
<Persona rdf:ID="Pedro"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.uv.es/ontologias/familia.owl">
  <esPadreDe rdf:resource="http://www.mifamilia.com/datos#Juan"/>
</Persona>
```

¡ Si Pedro “**es padre de**” Juan, entonces,  
Juan “**es hijo de**” Pedro !



# InverseFunctionalProperty (I)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.uv.es/ontologias/ontologiaAgua.owl">
  <owl:ObjectProperty rdf:ID="desembocaEn">
    <rdfs:domain rdf:resource="#Rio"/>
    <rdfs:range rdf:resource="#CuerpoDeAgua"/>
  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="seAbasteceDe">
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
    <owl:inverseOf rdf:resource="#desembocaEn"/>
    <rdfs:domain rdf:resource="#CuerpoDeAgua"/>
    <rdfs:range rdf:resource="#Rio"/>
  </owl:ObjectProperty>
  ...
</rdf:RDF>
```

Añadimos en el documento una nueva propiedad

# InverseFunctionalProperty (II)

Supongamos que hay dos documentos que describen dos grandes extensiones de agua:

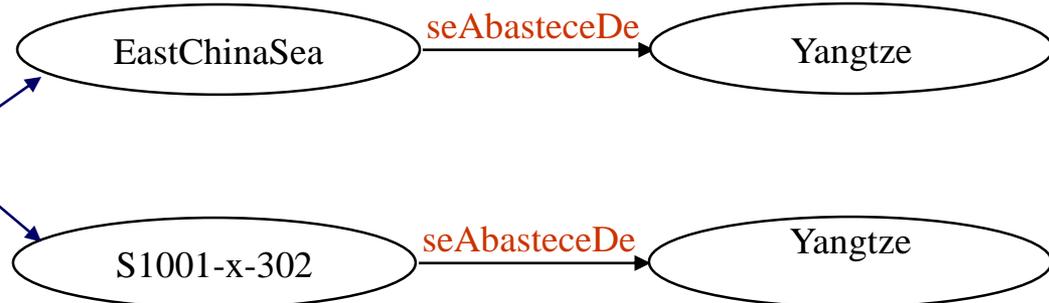
```
<?xml version="1.0"?>
<Mar rdf:about=" http://www.china.org/geography#EastChinaSea "
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns="http://www.uv.es/ontologias/ontologiaAgua.owl#">
  <seAbasteceDe rdf:resource="http://www.china.org/rivers#Yangtze"/>
</Mar>
```

**docEastChinaSea.rdf**

```
<?xml version="1.0"?>
<Mar rdf:about=" http://www.national-geographic.org#S1001-x-302"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <seAbasteceDe rdf:resource="http://www.china.org/rivers#Yangtze"/>
</Mar>
```

**docS1001.rdf**

Se infiere que estos dos  
valores se refieren a lo  
mismo



[Roger L. Costello, David B. Jacobs © The MITRE Corporation]

# Restricciones sobre Propiedades

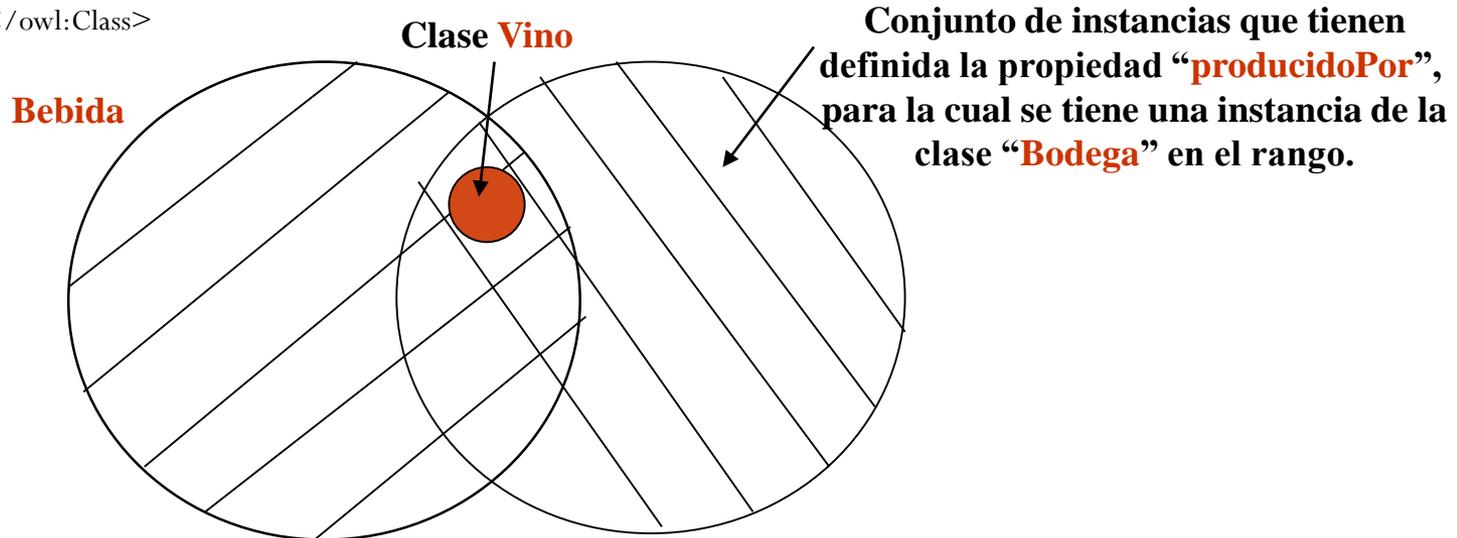
---

- owl:allValuesFrom
- owl:someValuesfrom
- Cardinalidad
  - owl:cardinality
  - owl:minCardinality
  - owl:maxCardinality
- owl:hasValue

# owl:allValuesFrom

( $\forall$ ) indica que todos los valores que se encuentren en el rango de la propiedad deben ser de un determinado tipo.

```
<owl:Class rdf:ID="Vino">
  <rdfs:subClassOf rdf:resource="#Bebida"/>
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="#producidoPor"/>
      <owl:allValuesFrom rdf:resource="#Bodega"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```



# owl:someValuesFrom

( $\exists$ ) Al menos un valor (en el rango) de la propiedad debe tener un determinado tipo.

```
<owl:Class rdf:ID="Vino">
  <rdfs:subClassOf rdf:resource="#Bebida"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#producidoPor"/>
      <owl:someValuesFrom rdf:resource="#Bodega"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

allValuesFrom	Para todos los vinos, <b><u>si tienen productores</u></b> , todos los productores serán bodegas. <ul style="list-style-type: none"><li>• No se requiere que un vino tenga un productor.</li></ul>
someValuesFrom	Para todos los vinos, <b><u>tienen al menos un productor</u></b> que es una bodega. <ul style="list-style-type: none"><li>• Puede haber otros productores que no sean bodegas.</li></ul>

# Cardinalidad

`owl:cardinality` permite especificar exactamente el número de valores que se pueden tomar para una propiedad.

```
<owl:Class rdf:ID="AnyoCosecha"/>
<owl:ObjectProperty rdf:ID="tieneAnyoCosecha">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Cosecha"/>
  <rdfs:range rdf:resource="#AnyoCosecha"/>
</owl:ObjectProperty>
```

Cada instancia de Cosecha tiene asociada, **como mucho**, una instancia de AnyoCosecha

```
<owl:Class rdf:ID="Cosecha">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#tieneAnyoCosecha"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Cada instancia de Cosecha tiene asociada, **exáctamente**, una instancia de AnyoCosecha

De igual forma, mediante las propiedades `owl:minCardinality` y `owl:maxCardinality`, se pueden restringir el número mínimo y máximo de valores que puede tomar una propiedad.

# owl:hasValue

Permite especificar clases basándose en la existencia de determinados valores que se pueden tomar para una cierta propiedad.

```
<owl:Class rdf:ID="Wine"/>
<owl:Class rdf:ID="WineSugar"/>
<owl:ObjectProperty rdf:ID="hasSugar">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Wine" />
  <rdfs:range rdf:resource="#WineSugar"/>
</owl:ObjectProperty>
<WineSugar rdf:ID="Dry" />

<owl:Class rdf:ID="Burgundy">
  <rdfs:subClassOf rdf:resource="#Wine" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasSugar"/>
      <owl:hasValue rdf:resource="#Dry" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

La propiedad **hasSugar** debe tener al menos un valor (la instancia **Dry**)

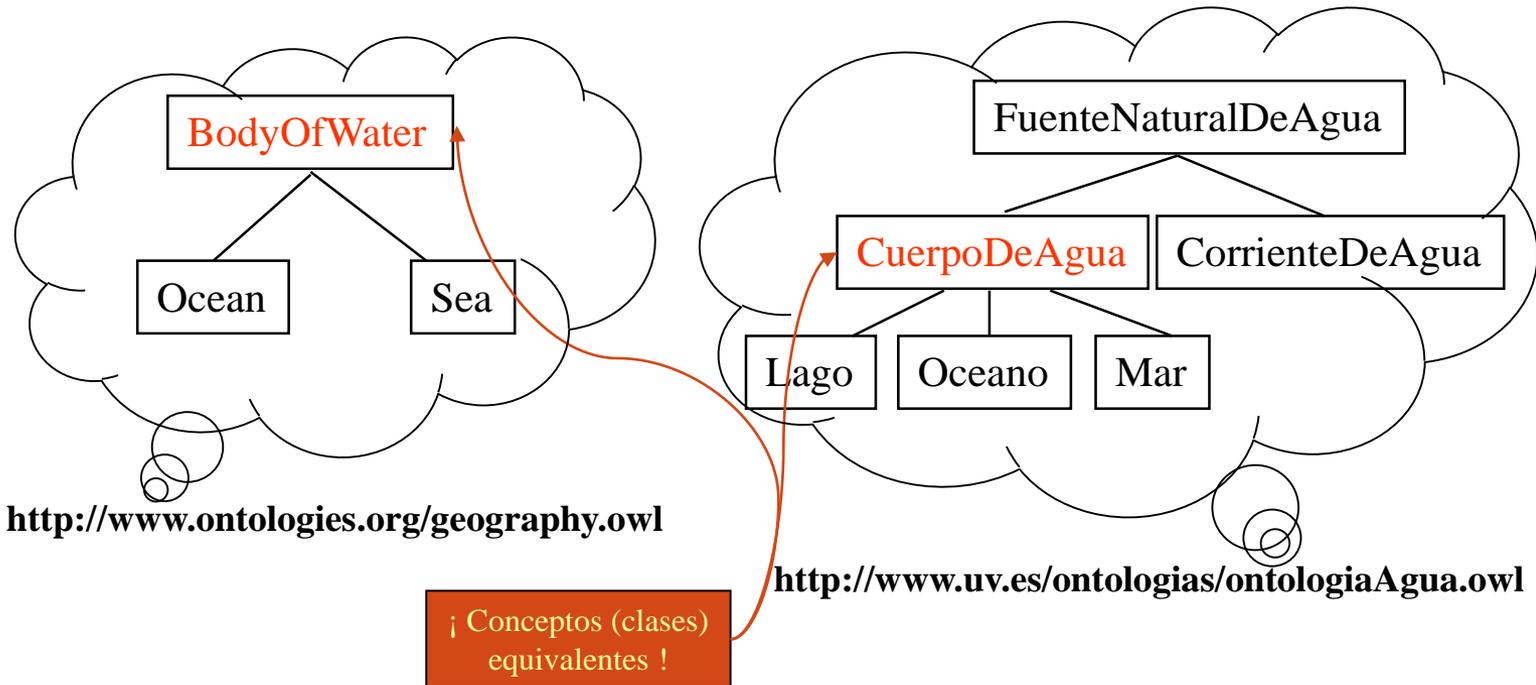
# Mapeado entre Ontologías

---

- Clases Equivalentes
- Propiedades Equivalentes
- Igualdad entre Instancias
- Instancias diferentes

# Clases Equivalentes (I)

- Para indicar que dos clases son equivalentes se utiliza la propiedad `owl:equivalentClass`.
- **Ejemplo:** Supongamos que existen dos ontologías distintas en las que se tienen definidos conceptos sobre grandes extensiones de agua. Tras analizar las definiciones de los distintos conceptos que aparecen en las ontologías, hemos encontrado dos conceptos equivalentes



# Classes Equivalentes (II)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.uv.es/ontologias/ontologiaAgua.owl">
  <Sea rdf:ID="CuerpoDeAgua">
    <rdfs:subClassOf rdf:resource="#FuenteNaturalDeAgua"/>
    <owl:equivalentClass
      rdf:resource="http://www.ontologies.org/geography.owl#BodyOfWater"/>
  </owl:Class>
  ...
</rdf:RDF>
```

# Propiedades Equivalentes

Para indicar que dos propiedades son equivalentes se utiliza la propiedad **owl:equivalentProperty**.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.uv.es/ontologias/ontologiaAgua.owl">

  <owl:DatatypeProperty rdf:ID="nombre">
    <owl:equivalentProperty rdf:resource="http://pur1.org/metadata/dublin-core#Title"/>
    <rdfs:domain rdf:resource="#FuenteNaturalDeAgua"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>

  ...

</rdf:RDF>
```

# Igualdad entre Instancias

De igual forma que se hace con clases, se puede definir igualdad entre Instancias de clases, mediante el uso de la propiedad **owl:sameIndividuals**

```
<?xml version="1.0"?>
<Sea rdf:ID="EastChinaSea"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <owl:sameIndividuals rdf:resource="http://www.nationalgeographic.org#S1001-x-302"/>
  ...
</Sea>
```

Ambas instancias se encuentran en distintos documentos.

# Instancias diferentes

Para indicar que un conjunto de instancias son distintas entre sí, podemos hacerlo de dos formas:

```
<owl:Class rdf:ID="ColorVino"/>
<ColorVino rdf:ID="Tinto"/>
<ColorVino rdf:ID="Blanco">
  <owl:differentFrom rdf:resource="#Tinto"/>
</ColorVino>
<ColorVino rdf:ID="Rosado">
  <owl:differentFrom rdf:resource="#Tinto"/>
  <owl:differentFrom rdf:resource="#Blanco"/>
</ColorVino>
```

```
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <ColorVino rdf:about="#Tinto" />
    <ColorVino rdf:about="#Blanco" />
    <ColorVino rdf:about="#Rosado" />
  </owl:distinctMembers>
</owl:AllDifferent>
```

Uso de  
**owl:differentFrom**

Uso de  
**owl:AllDifferent**  
junto con  
**owl:distinctMembers**

# Definición de Clases Complejas

---

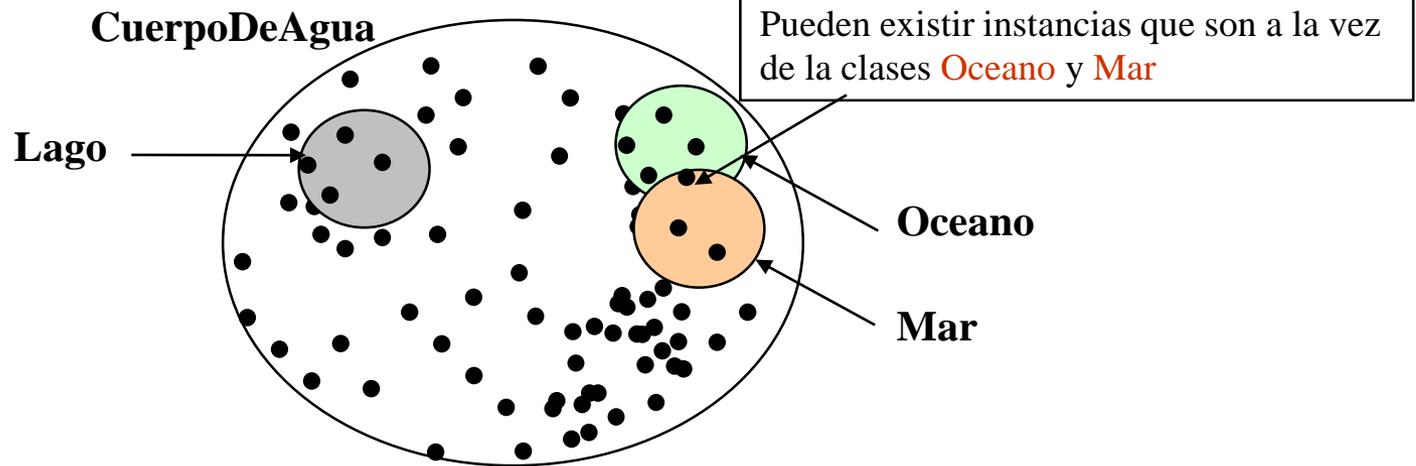
- owl:disjointWith (clases disjuntas)
- owl:intersectionOf
- owl:unionOf
- owl:complementOf
- owl:oneOf (clases enumeradas)

# Clases Disjuntas

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.uv.es/ontologias/ontologiaAgua.owl">
  <owl:Class rdf:ID="Lago">
    <rdfs:subClassOf rdf:resource="#CuerpoDeAgua"/>
    <owl:disjointWith rdf:resource="#Oceano"/>
    <owl:disjointWith rdf:resource="#Mar"/>
  </owl:Class>
```

La clase **Lago** es  
disjunta de las clases  
**Oceano** y **Mar**

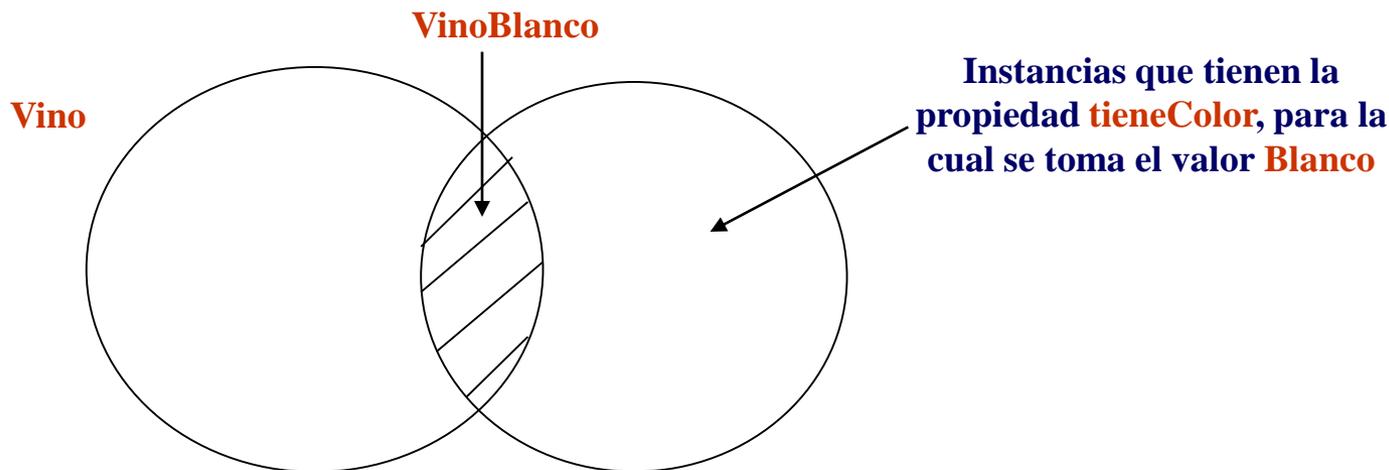
```
</rdf:RDF>
```



# owl:intersectionOf

```
<owl:Class rdf:ID="VinoBlanco">  
  <owl:intersectionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#Vino"/>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#tieneColor"/>  
      <owl:hasValue rdf:resource="#Blanco"/>  
    </owl:Restriction>  
  </owl:intersectionOf>  
</owl:Class>
```

Las instancias de “VinoBlanco” son aquellas instancias de la clase “Vino” que para la propiedad “tieneColor” toman el valor “Blanco”



# owl:unionOf

```
<owl:Class rdf:ID="FrutaDulce">
  <owl:disjointWith rdf:resource="#FrutaNoDulce"/>
</owl:Class>
<owl:Class rdf:ID="Fruta">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#FrutaDulce"/>
    <owl:Class rdf:about="#FrutaNoDulce"/>
  </owl:unionOf>
</owl:Class>
```

Dentro de la clase **Fruta** se tendrán instancias de la clase **FrutaDulce** e instancias de la clase **FrutaNoDulce**

Veamos que sucede si definimos la clase Fruta de la siguiente forma:

```
<owl:Class rdf:ID="FrutaDulce">
  <owl:disjointWith rdf:resource="#FrutaNoDulce"/>
</owl:Class>
<owl:Class rdf:ID="Fruta">
  <rdfs:subClassOf rdf:resource="#FrutaDulce"/>
  <rdfs:subClassOf rdf:resource="#FrutaNoDulce"/>
</owl:Class>
```

Estamos diciendo que la clase **Fruta** es un subconjunto de la intersección de las clases **FrutaDulce** y **FrutaNoDulce**

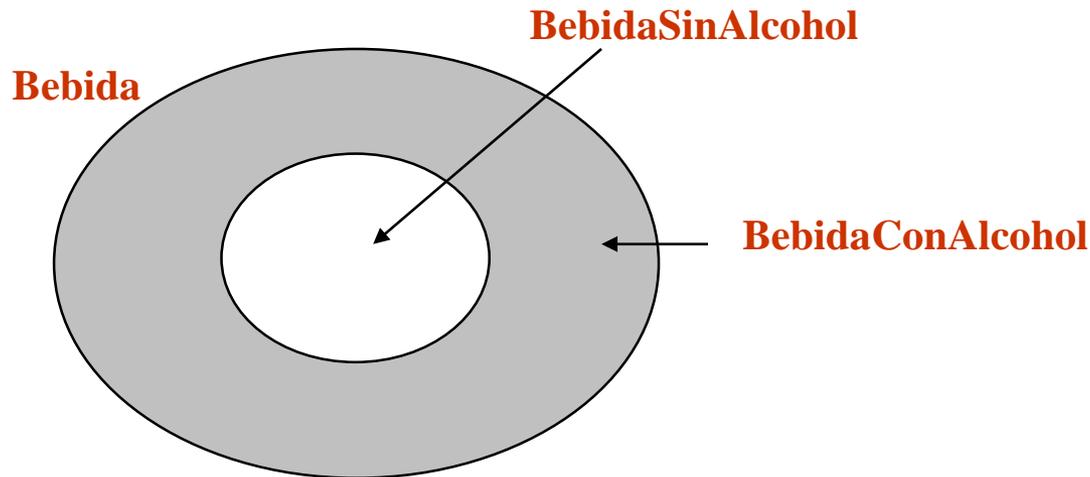
**CONSECUENCIA:**

**¡ No puede darse ninguna instancia de la clase Fruta !**

# owl:complementOf

Selecciona todas las instancias que no pertenecen a una cierta clase.

```
<owl:Class rdf:ID="BebidaSinAlcohol">
<owl:Class rdf:ID="BebidaConAlcohol">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Bebida"/>
    <owl:Class>
      <owl:complementOf rdf:resource="#BebidaSinAlcohol"/>
    </owl:Class>
  </owl:intersectionOf>
</owl:Class>
```



# owl:oneOf

Sólo se debe utilizar en los casos en que la clase enumerada tenga un número de instancias no variable (al menos a corto plazo) y no muy grande.

```
<owl:Class rdf:ID="NombreDia"  
  <owl:oneOf rdf:parseType="Collection">  
    <owl:Thing rdf:about="#Lunes"/>  
    <owl:Thing rdf:about="#Martes"/>  
    <owl:Thing rdf:about="#Miércoles"/>  
    <owl:Thing rdf:about="#Jueves"/>  
    <owl:Thing rdf:about="#Viernes"/>  
    <owl:Thing rdf:about="#Sábado"/>  
    <owl:Thing rdf:about="#Domingo"/>  
  </owl:oneOf>  
</owl:Class>
```

El conjunto de los nombre de días es un caso evidente de clase enumerada.

Cualquier nombre de instancia que no se encuentre definido en la clase enumerada no será válido.

Se ha utilizado **owl:Thing** para referenciar cada elemento de la clase enumerada.

```
<owl:Class rdf:ID="NombreDia"  
  <owl:oneOf rdf:parseType="Collection">  
    <NombreDia rdf:about="#Lunes"/>  
    <NombreDia rdf:about="#Martes"/>  
    ...  
  </owl:oneOf>  
</owl:Class>
```

También se pueden definir los elementos del conjunto especificando directamente su tipo (**NombreDia**)



# Referencias

---

- Libros:

- Daconta, Michael C.; Obrst, Leo, J. and Smith, Kevin, T “The semantic Web. A guide to the future of XML, Web services, and knowledge management”.Ed. Wiley, ISBN: 0-471-43257-1. (2003)
- Hjelm, Johan, “Creating the Semantic Web with RDF, Professional Developer’s Guide”. ISBN 0-471-40259-1, Wiley (2001)

- Documentos de interés:

- Naming and addressing: URIs, URLs...  
<http://www.w3.org/Addressing/#9808uri>
- Resource Description Framework. W3C RDF Home page. <http://www.w3.org/RDF/>
- <http://www.sidar.org/recur/desdi/traduc/es/rdf/rdfesp.htm> (CASTELLANO)
- Tutorial. XML Design. (Gentle Transition from XML to RDF). Roger L. Costello, David B. Jacobs @2003 the MITRE Corporation. Sponsored by DARPA
- Tutorial. A Quick Introduction to. OWL Web Ontology Language. Roger L. Costello. David B. Jacobs.
- UPM, apuntes  
<http://www.sia.eui.upm.es/~jbarrasa/teaching/>

# Referencias (parte de los metadatos)

---

- Repositorio de esquema de metadatos y ontologías (<http://www.schemaweb.info/>)
- **Dublin Core:**
  - <http://dublincore.org/> (página de la iniciativa DCMI)
  - <http://es.dublincore.org/> (página en castellano sobre DCMI)
  - registro de los metadatos definidos en DCMI (<http://dublincore.org/dcregistry/>)
  - [DC-dot] Editor de metadatos DC-dot (<http://www.ukoln.ac.uk/metadata/dcdot/>)
  - [DC-Gen] Dublin Core Metadata Gen: Generador de metadatos Dublin Core (<http://www.webpossible.com/utilidades/dublincore-metadata-gen/>)
- **FOAF:**
  - <http://xmlns.com/foaf/0.1/> (especificación del vocabulario FOAF)
  - <http://rdfweb.org/topic/FoafProject> (información, herramientas, ... sobre el proyecto FOAF)
  - [FOAF-a-matic] Aplicación de Javascript que permite crear un descripción FOAF (<http://www.ldodds.com/foaf/foaf-a-matic>)
  - [FOAF Explorer] Visualizador de la información de un fichero FOAF (<http://xml.mfd-consult.dk/foaf/explorer/>)
- **LOM:**
  - <http://ltsc.ieee.org/wg12/index.html> (página del comité LTSC del IEEE dedicada al estándar LOM)
  - <http://www.gist.uvigo.es/~lanido/LOMes/> (página desarrollada por la Universidad de Vigo que ofrece información sobre LOM en castellano)
  - [IMS] IMS Global Learning Consortium (<http://www.imsglobal.org/metadata/index.html>)
  - [ARI] Fundación ARIADNE (<http://www.ariadne-eu.org/>)