

# Tecnologías para la Web Semántica

Arquitectura III  
RDF – RDFS

# XML

David Billington is a lecturer of Discrete Maths

## Opción 1

```
<course name="Discrete Maths">  
  <lecturer>David Billington</lecturer>  
</course>
```

## Opción 2

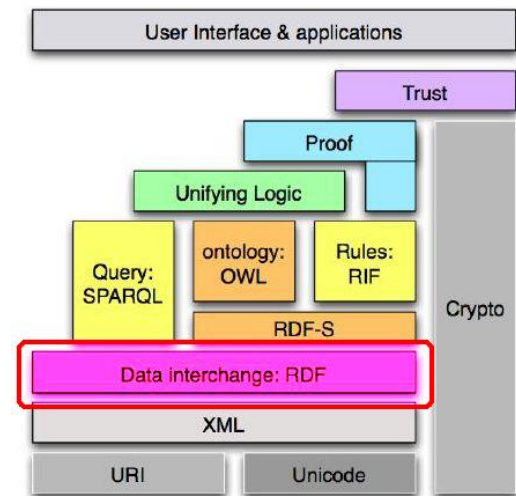
```
<lecturer name="David Billington">  
  <teaches>Discrete Maths</teaches>  
</lecturer>
```

**Anidamiento opuesto con la misma informacion!**  
**No existen procedimientos estandarizados para asignar significado**

# RDF

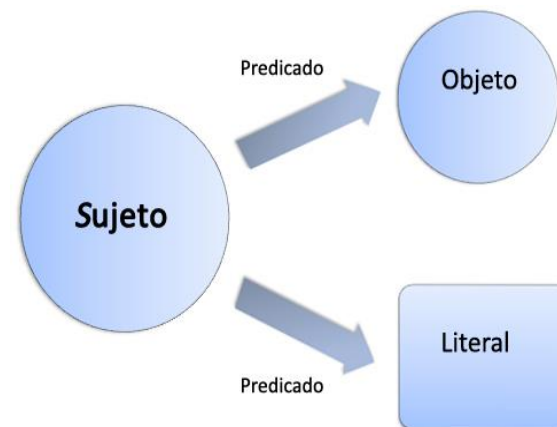
**R**esource **D**escription **F**ramework  
(infraestructura para la descripción de recursos)

- Recomendación de W3C
- Estandariza la definición y uso de metadatos (útil para la representación de datos)
- Usa la sintaxis de XML
- Soluciona las carencias de XML, y agrega semántica.
- Lenguaje centrado en propiedades no en recursos
- Posee semántica formal



# RDF – Tripletas

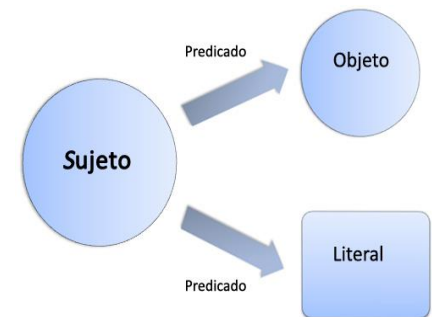
- ▶ Cada tripleta representa una declaración de una relación entre los elementos denotados por los vínculos.
- ▶ Cada tripleta tiene 3 partes:
  - Un sujeto
  - Un objeto
  - Un predicado (también llamado propiedad) que denota una relación.
- ▶ La dirección del arco es significativa: siempre apunta hacia el objeto.
- ▶ Conjunto tripletas: **Grafo**



# Recurso

- ▶ Podemos ver una “cosa” o “recurso” sobre lo que queremos hacer referencia.
  - Ej. authors, books, publishers, places, people, hotels
- ▶ Cada recurso tiene un **URI**, a Universal Resource Identifier
- ▶ Un URI puede ser
  - una URL (dirección web)
  - otra clase de identificador unico

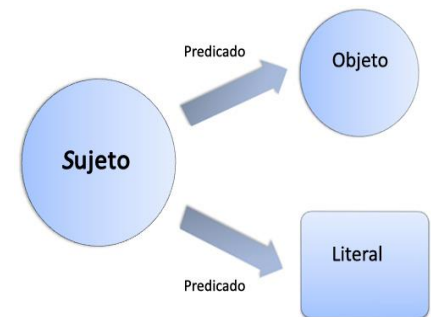
Para nosotros una URI es el identificador de un recurso web



# Propiedades

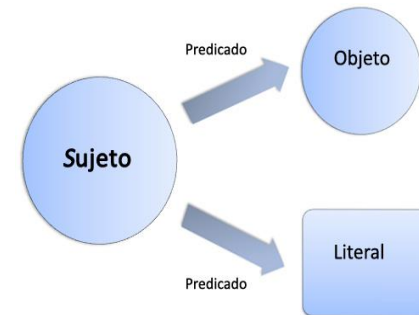
- ▶ Propiedades son una clase especial de recursos
- ▶ Describen relaciones entre recursos
  - Ej. “written by”, “age”, “title”, etc.
- ▶ También se identifican por URIs
- ▶ Brinda un esquema unico y global para nombrar a las cosas
- ▶ Reduce el problema del manejo de homónimos de la representación distribuida.

Homónimos son aquellos términos o palabras que, aunque se escriben o pronuncian de manera similar, tienen diferente valor gramatical, como por ejemplo: más-mas..



# Sentencias

- ▶ Las sentencias establecen las propiedades de los recursos
- ▶ Una sentencia es una tripleta del tipo objeto-atributo-valor
  - Consiste en un recurso, una propiedad y un valor
- ▶ Los valores pueden ser recursos o literales
  - Literales son valores atomicos (strings)

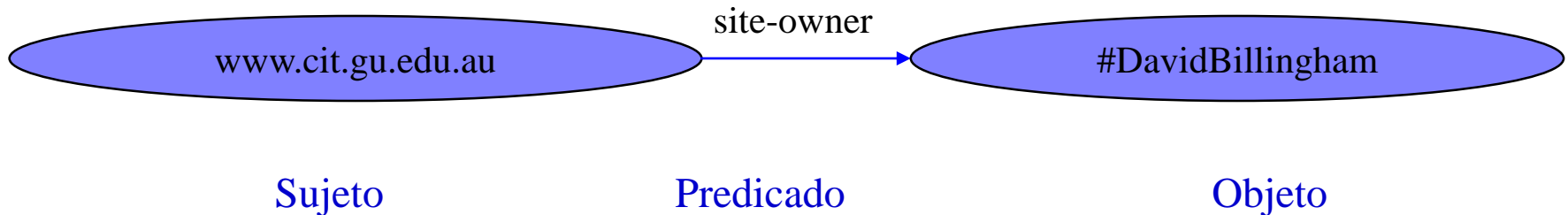


# RDF: Grafo

Ejemplo de sentencia:

Grafo conocido en IA  
como Red Semántica.

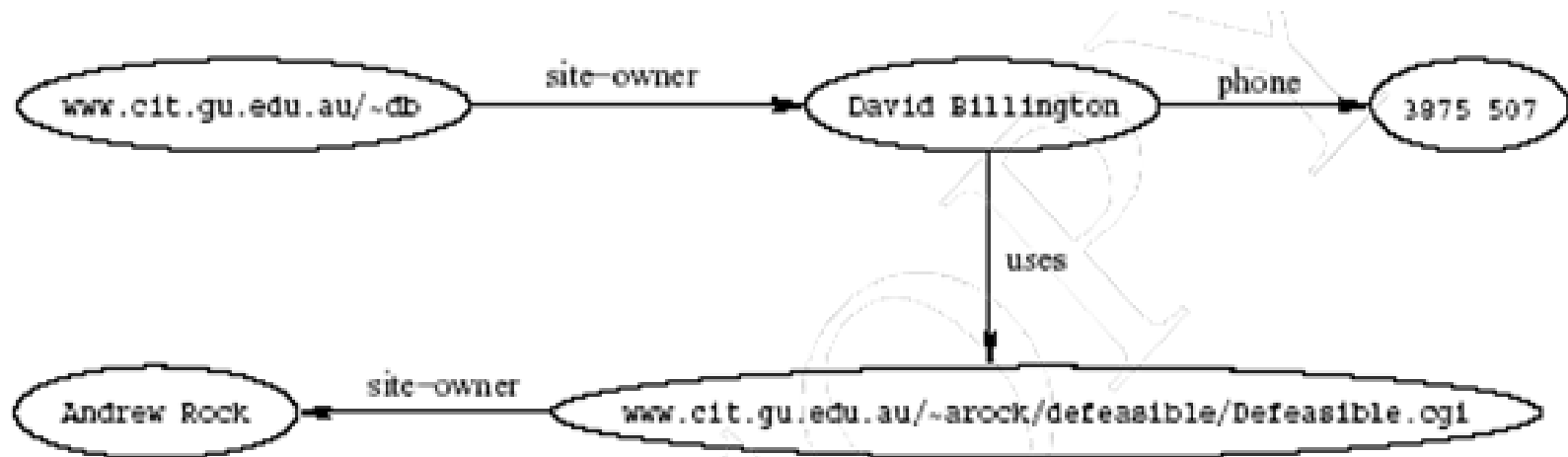
David Billingham is the owner of the Web page  
<http://www.cit.gu.edu.au/db>



(<http://www.cit.gu.edu.au/db>, <http://mydomain.org/site-owner>, #DavidBillingham)



# RDF: Conjunto de sentencias como Red Semántica

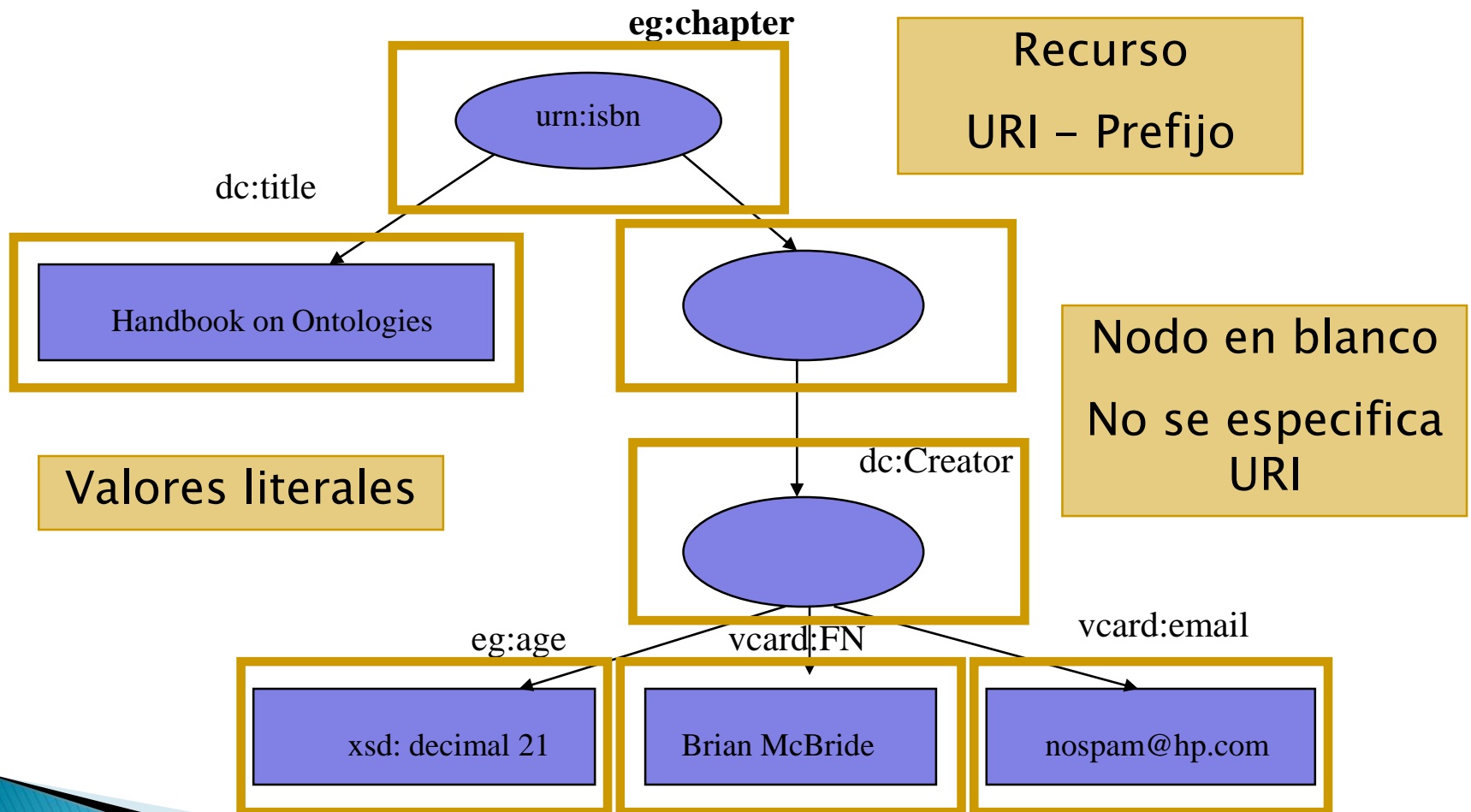


# Sentencias

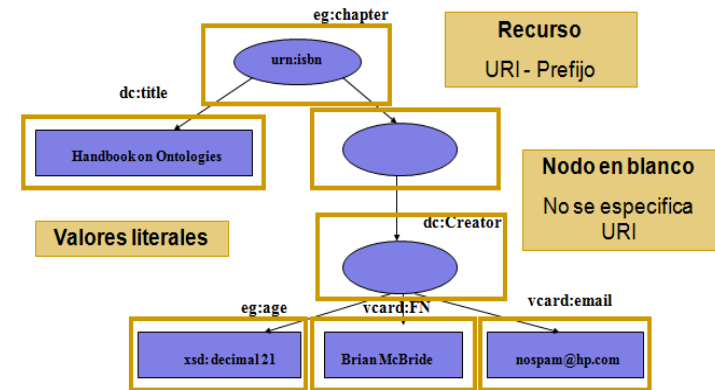
- ▶ Elemento XML con la etiqueta `rdf:RDF` cuyo contenido es un numero de elementos “description”.
- ▶ Cada elemento “description” realiza una afirmación acerca de un recurso. Se identifica de la siguiente manera:
  - Un atributo “about” referenciando un recurso existente
  - Un atributo ID creando un nuevo recurso
  - Sin un nombre, creando un recurso anonimo

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mydomain="http://www.mydomain.org/my-rdf-ns">
  <rdf:Description>
    <rdf:about="http://www.cit.gu.edu.au/~db">
      <mydomain:site-owner
        rdf:resource="#David Billington"/>
    </rdf:Description>
  </rdf:RDF>
```

# RDF



# RDF/XML



```
<rdf:Description rdf:about="urn:isbn:...">
  <dc:Title>Handbook on Ontologies</dc:Title>
  <eg:chapter>
    <rdf:Description>
      <dc:Creator>
        <rdf:Description>
          <vcard:FN>Brian McBride</vcard:FN>
          <vcard:email>nospam@hp.com</vcard:email>
          <eg:age rdf:datatype="&xsd;decimal">21</eg:age>
        </rdf:Description>
      </dc:Creator>
    </rdf:Description>
  </eg:chapter>
</rdf:Description>
```

Un nodo en blanco es un recurso para el cual no se detalla su URI. Empleados para representar estructuras compuestas, relaciones N-arias.

# RDF: Beneficios

- ▶ RDF tiene suficiente poder expresivo, como base sobre la cual otras capas de la arquitectura de la Web Semantica se pueden construir.
- ▶ La WS no se va a programar en RDF pero si con herramientas que van a traducir en forma automatica representaciones de más alto nivel en RDF.
- ▶ Con RDF la información mapea sin ambigüedad a un modelo.
- ▶ Como RDF es un estandar, trabajar sobre RDF equivale a trabajar en HTM en los primeros tiempos de la Web.

# RDF: Desventajas

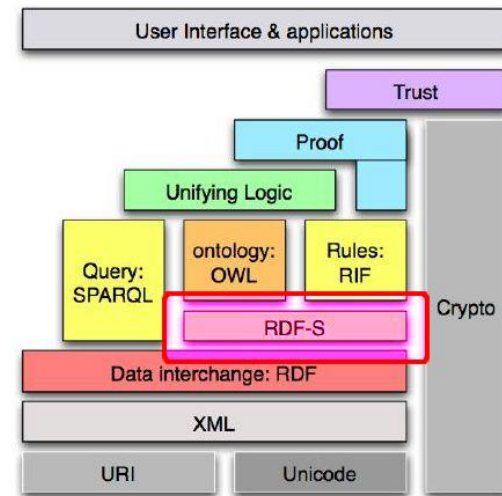
- ▶ RDF permite la afirmación de sentencias simples que consisten en **sujeto-predicado-objeto**.
- ▶ No describe lo que estos elementos significan sino que describen las relaciones que existen entre ellos.



**RDF  
SCHEMA**

# RDFS

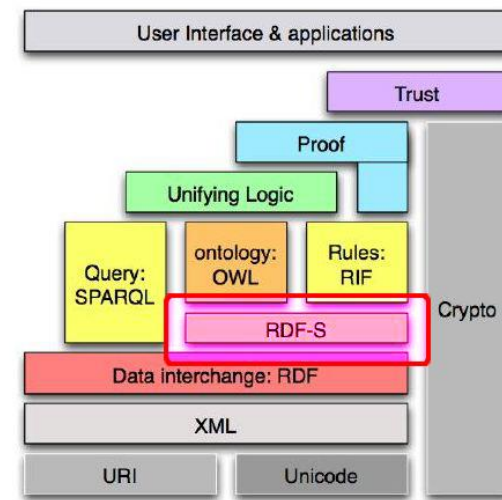
RDF trabaja con un Schema (RDFS): definición de vocabulario



RDFS no provee clases ni propiedades particulares de una aplicación, sino que otorga un framework para describir esas clases y propiedades.

Expresiones RDF SCHEMA son expresiones RDF válidas

# RDFS

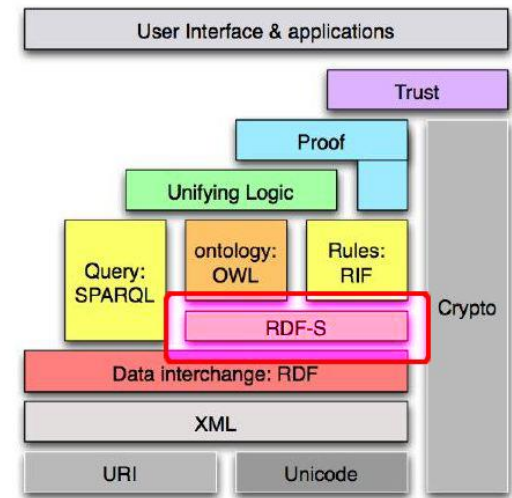


- ▶ Introduce conceptos ontológicos simples
  - Introduce el concepto de clase
  - Define cómo los recursos pueden describirse como pertenecientes a una o más clases
  - Describe jerarquía de clases y propiedades
  - Define dominio y rango de propiedades



# RDFS

Las clases en un RDF Schema son comparables a las clases en lenguajes de programación orientada a objetos.



Los recursos pueden ser definidos como instancias de clases o subclases de clases.

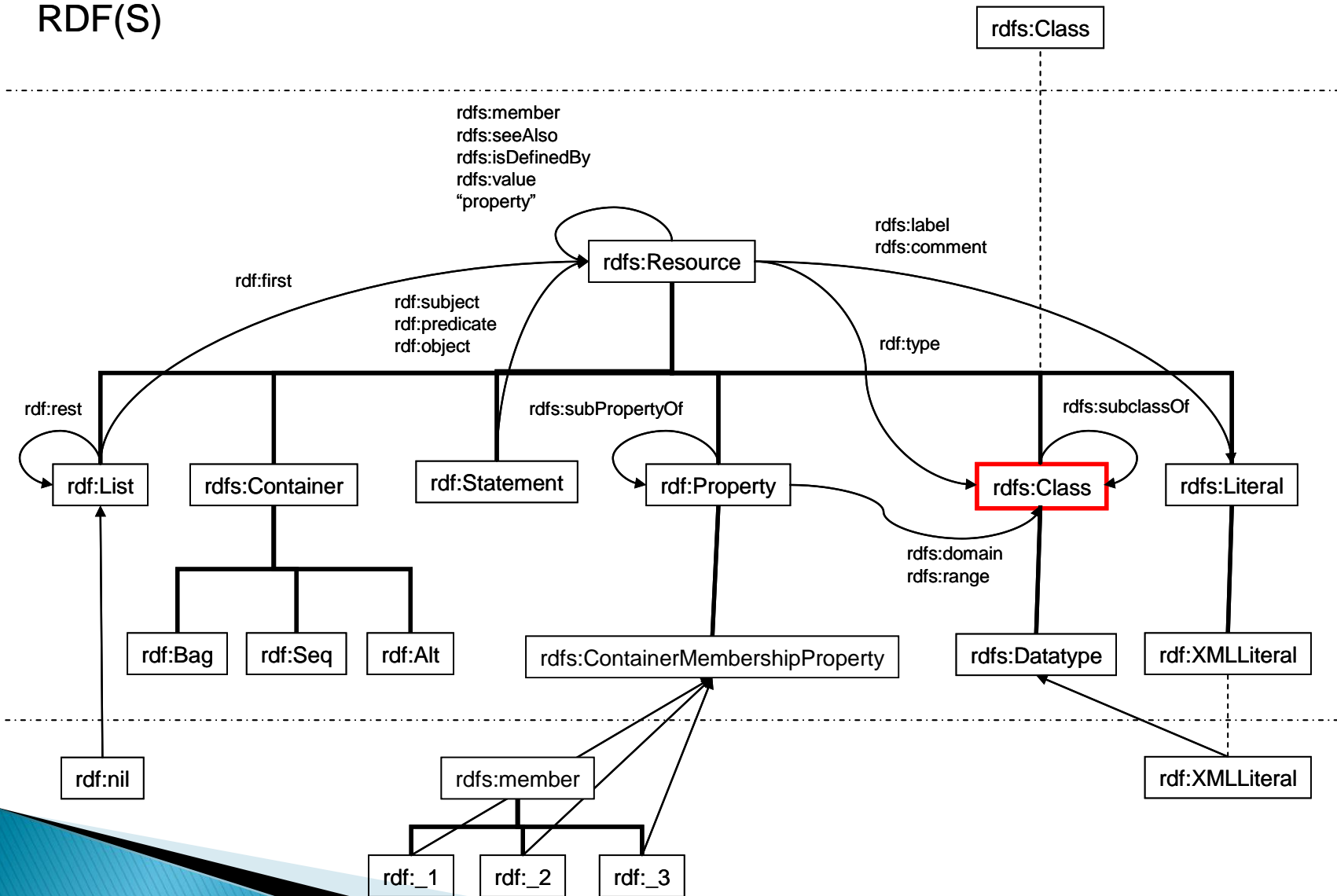
# RDFS

- ▶ Una ontología en RDF(S) debe comenzar con un nodo raíz RDF donde se incluyen los *namespaces* para las ontologías RDF y RDF(S) (ontologías de representación del conocimiento).
- ▶ El uso de los *namespaces* permite utilizar los prefijos rdf y rdfs para las primitivas que pertenecen a RDF y RDFS

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
          xml:base="http://www.geodesy.org/water/naturally-
occurring">
```

# RDFS model

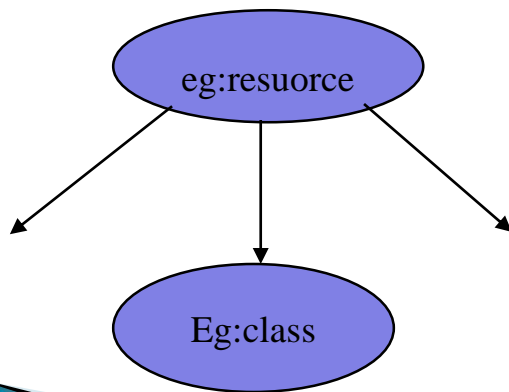
RDF(S)



# RDFS

## Clase

- ▶ Una clase representa una colección de recursos
- ▶ Son recursos en si mismas identificados por URIs
- ▶ Un recurso comienza a ser un miembro de una clase utilizando la propiedad *rdf:type*
- ▶ Conceptos son clases y subclasses en RDF(S)
- ▶ Se referencian por nombre o URL a un recurso web



```
<rdfs:Class rdf:ID="Travel" />  
  <rdfs:comment> A journey from place to  
  place</rdfs:comment>  
</rdfs:Class>
```

# RDFS

## Clase

*rdfs:subClassOf* indica que una clase es subclase de otra

```
<rdf:RDF
xml:base="http://example.org/example">
  <rdfs:Class rdf:ID="Persona"/>
  <rdfs:Class rdf:ID="Hombre">
    <rdfs:subClassOf rdf:resource="#Persona">
  </rdfs:Class>
</rdf:RDF>
```

# RDFS – Clases y subclases

## Clase

```
<?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#"
  xml:base="http://www.turismo.com/alojamientos#">
  <rdfs:Class rdf:ID="alojamiento" />
  <rdfs:Class rdf:ID="hotel">
    <rdfs:subClassOf rdf:resource="#alojamiento"/>
  </rdfs:Class>
</rdf:RDF>
```

# RDFS Propiedades

## Propiedades

- ▶ Propiedades RDF son recursos
- ▶ *rdf:Property* es la clase de todas las propiedades

```
<rdf:RDF
xml:base="http://example.org/example">
  <rdf:Property rdf:ID="socioDe"/>
  <rdf:Property rdf:ID="maridoDe">
    <rdfs:subPropertyOf rdf:resource="#socioDe">
  </rdf:Property>
</rdf:RDF>
```

# RDFS

## Propiedades

- ▶ Atributos de instancia de clases se definen como propiedades en RDF(S)
- ▶ El dominio de estas propiedades es la clase a la que pertenece el atributo y el rango es el tipo del valor del atributo
- ▶ No se definen restricciones de cardinalidad ni valores por omisión
- ▶ Propiedades en RDF son recursos

```
<rdf:Property rdf:ID="arrivalDate">  
  <rdfs:domain rdf:resource="#Travel"/>  
  <rdfs:range rdf:resource="&xsd;date"/>  
</rdf:Property>  
  
<rdf:Property rdf:ID="companyName">  
  <rdfs:domain rdf:resource="#Travel"/>  
  <rdfs:range rdf:resource="&rdfs;Literal"/>  
</rdf:Property>
```



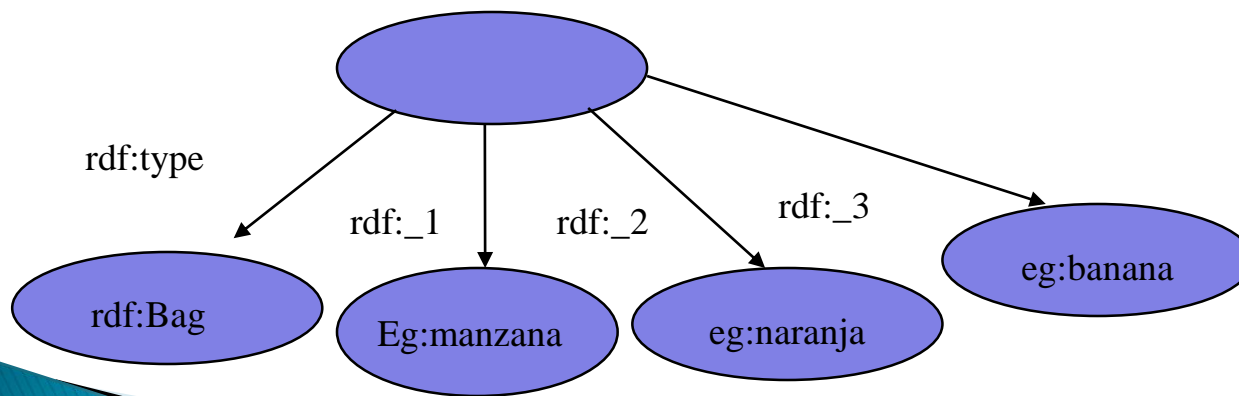
# RDFS Propiedades

- ▶ Atributos de clase se representan de manera similar.
- ▶ El dominio de la propiedad se define como `rdfs:class`, y se incluye el valor de la propiedad en la definición de la clase.

# RDF(S) Contenedores

*rdfs:Container* es un recurso que contiene literales o recursos.

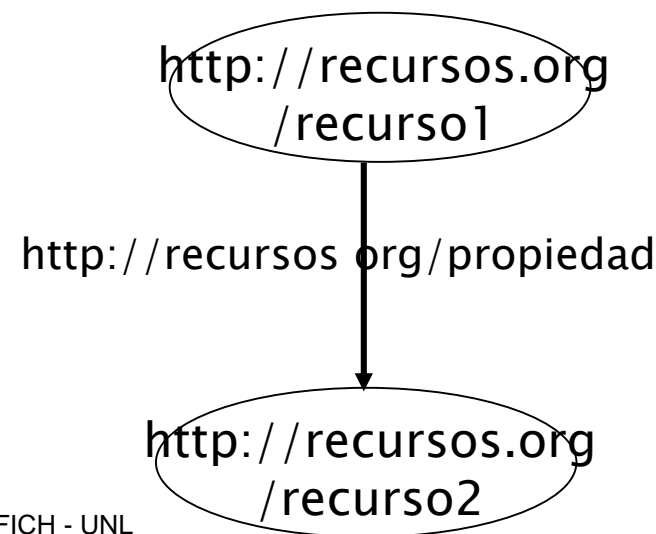
- *rdf:Seq* (el orden es relevante)
- *rdf:Bag* (el orden no es relevante)
- *rdf:Alt* (su utilización es la selección de uno de los miembros del contenedor)



```
<rdf:RDF
  xml:base="http://example.org
/example">
  <rdf:Seq>
    <rdfs:li
      rdf:resource="manzana">
    <rdfs:li
      rdf:resource="naranja">
    <rdfs:li
      rdf:resource="banana">
  </rdf:Seq>
</rdf:RDF>
```

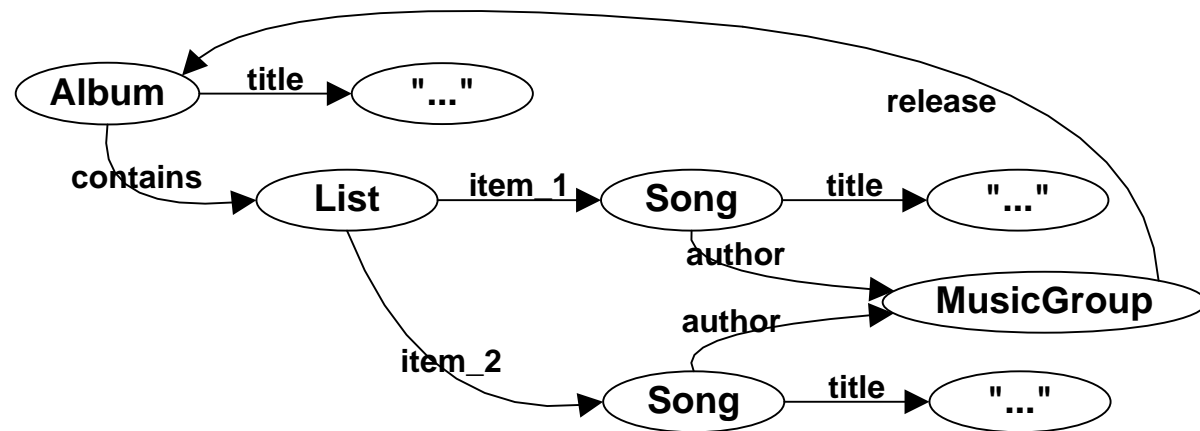
# RDF

- ▶ Una referencia URI o un literal utilizado como nodo, identifica lo que el nodo representa.
- ▶ Una referencia URI utilizada como predicado identifica una relación entre los elementos representados por los nodos que conecta.
- ▶ Una referencia URI predicado puede también ser un nodo en el grafo.



# RDF

## ► Ejemplo grafo RDF:



# Class y Property: diferentes *namespaces*

- ▶ *Class* se coloca en *rdfs namespace*.
- ▶ *Property* se coloca en *rdf namespace*.

# Componentes para una WS

- ▶ **[XML](#)** nos da la **sintaxis** para documentos estructurados, pero no agrega semántica.
- ▶ **[XML Schema](#)** restringe la estructura de documentos XML y **extiende** a XML con datatypes.
- ▶ **[RDF](#)** es un **modelo de datos** para objetos ("recursos") y relaciones entre ellos. Provee **semántica simple** para este modelo de datos, y puede ser representado con sintaxis de XML.
- ▶ **[RDF Schema](#)** es un **vocabulario para describir clases y propiedades** de recursos RDF, usando **semántica** para jerarquías generalizadas de esas propiedades y clases.
- ▶ **[OWL](#)** **agrega vocabulario** para describir propiedades y clases: entre otros, relaciones entre clases, cardinalidad, igualdad, características de propiedades, etc.