

# *Tecnologías XML y Web Semántica*



Departamento de Informática  
Universidad de Oviedo

# *Ontologías y Descripción de Recursos*

RDF, RDF(S) y OWL  
Definición de Ontologías

## *Perspectivas y Aplicaciones*

Algunas aplicaciones  
Servicios Web Semánticos



Departamento de Informática  
Universidad de Oviedo



## *Esquema de la Sesión*

9-10:30h RDF - OWL

10:30-11h Descanso

11-12:30h Prácticas (Protégè)

12:30h-13h Descanso

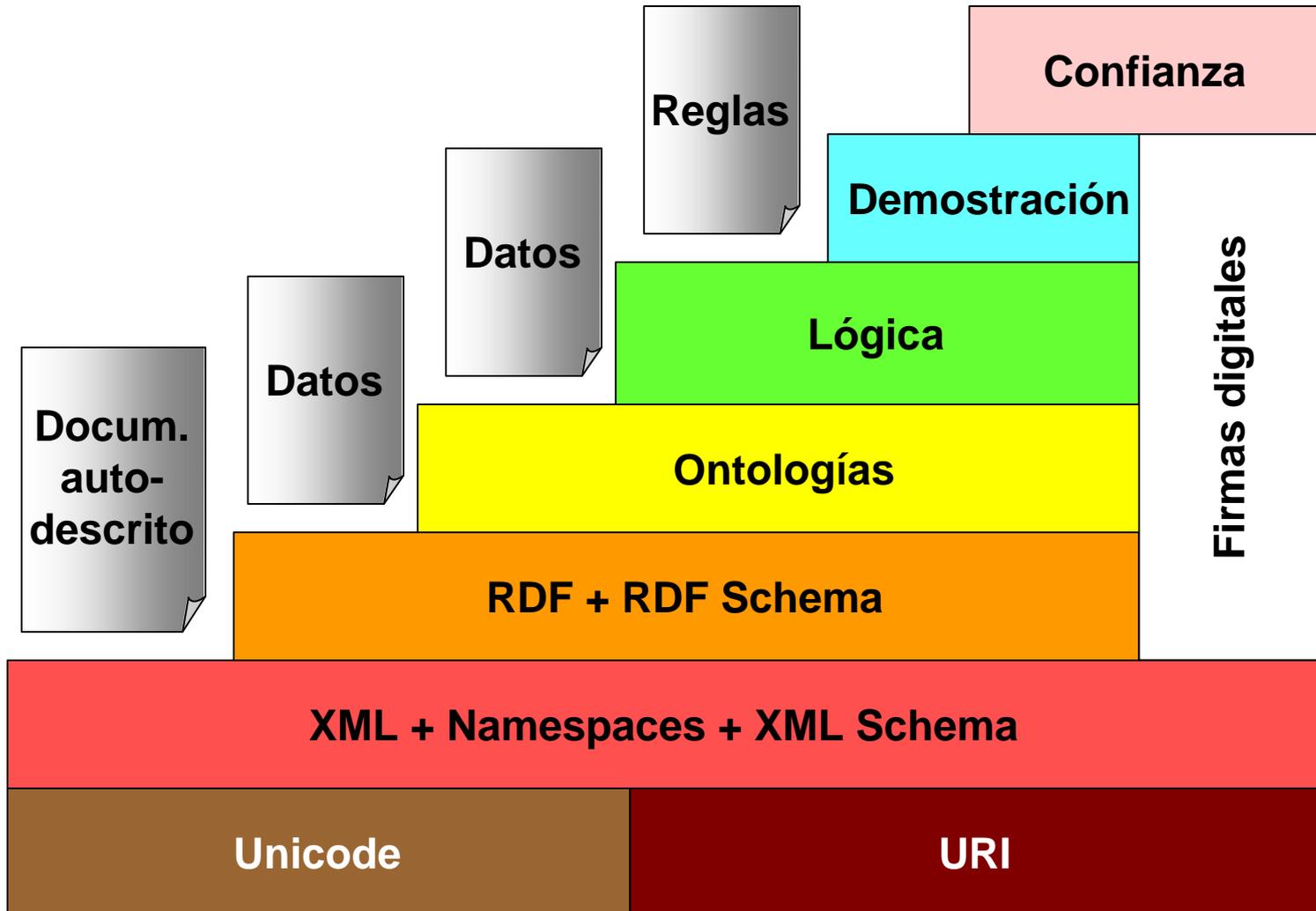
13-15h Perspectivas y aplicaciones de la Web Semántica



1ª Parte  
*Ontologías y Descripción de Recursos*



# Web Semántica





*RDF*



# RDF (Resource Description Framework)

Permite describir recursos mediante propiedades y valores de propiedades

Recurso: Cualquier cosa que pueda nombrarse mediante una URI

Propiedad: Característica o atributo de un recurso

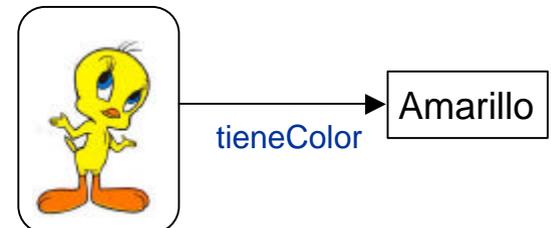
Tiene asociada una URI y un significado concreto

Puede relacionarse con otras propiedades

Enunciado: Asocia el valor de una propiedad a un recurso

Ejemplo: *Piolín es de color amarillo*

Representación gráfica





RDF define un modelo (= Grafo acíclico dirigido)

Nodos Orígen = URIs

Arcos = URIs

Nodos destino (URIs o literales)



Se podría representar como:

(<http://miDirección/#Piolín>, <http://www.sentidos.net/tieneColor>, "Amarillo")  
(<http://miDirección/#Piolín>, <http://purl.org/DC/Creator>, Warner)



El modelo RDF se puede representar en sintaxis XML (serialización)



```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://www.sentidos.net/"
  xmlns:dc="http://purl.org/DC/">
  <rdf:Description about="http://miDireccion/#piolin">
    <s:tieneColor>Amarillo</s:tieneColor>
    <dc:Creator resource="mailto:warner@cartoons.com" />
  </rdf:Description>
</rdf:RDF>
```



## *Ideas básicas de RDF*

Los conceptos fundamentales de RDF son:

Recursos

Propiedades

Enunciados



# *Ideas Básicas de RDF*

## *Recursos*

Un **recurso** es un objeto (*una cosa*) del que queremos decir algo

Ej. Autores, libros, editoriales, lugares, personas, hoteles....

Todo recurso tiene una URI

Una URI puede ser

Una URL

Un nombre único (URN)



# *Ideas Básicas de RDF*

## *Propiedades*

Las **propiedades** son un tipo especial de recurso

Describen relaciones entre recursos

Ej. "escritoPor", "edad", "título"

Las propiedades también se identifican mediante URIs

Ventajas de usar URIs

Esquema de nombres global, único

Se evita el problema de la homonimia



# *Ideas básicas de RDF*

## *Enunciados*

Los **enunciados** (statements) asocian propiedades a recursos

Un enunciado es una tripleta:

Objeto -- Propiedad -- Valor

Los valores de las propiedades pueden ser URIs o literales

Un literal es una cadena de caracteres



# Ejemplo de editor Visual: IsaViz

The screenshot displays the IsaViz RDF Editor interface. The main window, titled "Graph", shows a visual representation of an RDF graph. The graph consists of three nodes: a yellow rectangular node labeled "Amarillo", a green oval node labeled "http://miDireccion/#piolin", and another green oval node labeled "mailto:warner@cartoons.com". A blue curved arrow points from the "http://miDireccion/#piolin" node to the "Amarillo" node. A blue straight arrow points from the "mailto:warner@cartoons.com" node to the "http://miDireccion/#piolin" node, with the label "dc:Creator" positioned next to the arrow. A small floating window titled "N..." is positioned above the graph, containing navigation and zoom icons.

The "Overview" panel (bottom-left) provides a smaller, zoomed-out view of the same graph structure. The "Definitions" panel (bottom) contains a table of namespace definitions:

Prefix	URI	Display Prefix
dc	http://purl.org/DC/	<input checked="" type="checkbox"/>
s	http://www.sentidos.net/	<input checked="" type="checkbox"/>
	http://www.uni.org/	<input checked="" type="checkbox"/>
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	<input checked="" type="checkbox"/>
rdfs	http://www.w3.org/2000/01/rdf-schema#	<input checked="" type="checkbox"/>
...	...	...

Buttons for "Add NS Binding" and "Remove Selected" are located at the bottom right of the Definitions panel.



# Sintaxis abreviada

```
<rdf:Description rdf:ID="Piolin">  
  <s:tieneColor>Amarillo</s:tieneColor>  
  <rdf:type resource="#Canario" />  
</rdf:Description>
```

Si las propiedades no se repiten,  
pueden incluirse como atributos

```
<rdf:Description rdf:ID="Piolin" s:tieneColor="Amarillo">  
  <rdf:type resource="#Canario" />  
</rdf:Description>
```

La declaración de “**type**” puede  
incluirse en la etiqueta

```
<rdf:Canario rdf:ID="Piolin" s:tieneColor="Amarillo" />
```



Sintaxis no XML simplificada que facilita la representación de tripletas

Objetivo: Legibilidad por desarrolladores

Utilizada por CWM

```
@prefix : <http://miDireccion/#> .
```

```
@prefix dc: <http://purl.org/DC/> .
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
@prefix s: <http://www.sentidos.net/> .
```

```
:piolin    dc:Creator <mailto:warner@cartoons.com>.
```

```
:piolin    s:tieneColor "Amarillo" .
```



# RDF: Contenedores

## Tipos

Bag: Conjunto no ordenado (permite duplicados)

Seq: Lista ordenada (permite duplicados)

Alt: Valor único alternativo (elección de un elemento del contenedor)

Los elementos se indican con `<li>`

Pueden tener **ID** (pero no **about**)

Pueden aplicarse propiedades a los elementos con **aboutEach**

```
<rdf:Description rdf:ID="Película">
  <dc:title>El canario peleón</dc:title>
  <m:personajes>
    <rdf:Bag ID="ps1">
      <li resource="http://warner.com/#piolin" />
      <li resource="http://warner.com/#silvester" />
    </rdf:Bag>
  </m:personajes>
</rdf:Description>

<rdf:Description aboutEach="#ps1">
  <dc:Creator resource="mailto:warner@cartoos.com" />
</rdf:Description>
```



## *RDF: Reificación*

Permite definir sentencias sobre sentencias (orden superior)

Ej. El sitio Web de la EUITIO dice que Labra es el profesor de Lógica

(WebEUITIO, dice, s1)

(s1, Subject, labra)

(s1, Predicate, esProfesorDe)

(s1, Object, lógica)

Una descripción define implícitamente un **Bag** de sentencias

RDF define automáticamente las sentencias como recursos y las incluye en un Bag

Las sentencias se representan con el tipo predefinido **rdf:Statement**

Los atributos de **rdf:Statement** son:

**rdf:Subject**

**rdf:Predicate**

**rdf:Object**

Es posible añadir otros atributos a las sentencias



## *RDF: Tipos de Datos*

Es posible utilizar tipos de datos

En general se utilizan los tipos de XML Schema

Podrían utilizarse otros tipos de datos

Se usa la notación ^^

```
<rdf:Description rdf:about="949318">
```

```
  <uni:nombre>Jose Labra</uni:nombre>
```

```
  <uni:cargo>Profesor Titular</uni:cargo>
```

```
  <uni:edad rdf:datatype="&xsd:integer">35<uni:age>
```

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



## *RDF Schema*



# *RDF Schema*

## *Motivación*

RDF es un modelo universal que permite describir recursos definidos en ciertos vocabularios

RDF no asume ni define ningún tipo de semántica de un dominio concreto

En RDF Schema pueden definirse:

Clases y propiedades

Jerarquías y herencia entre clases

Jerarquías de propiedades



# RDF Schema

## Clases e individuos

Hay que distinguir entre:

Cosas concretas (individuos) del dominio.

Ej. "Jose Labra", "Lógica"

Clases o conceptos = Conjuntos de individuos que comparten algunas propiedades (***rdfs:Class***)

Ej. "Profesor", "Asignatura", "Estudiante", ...

Los individuos que pertenecen a una clase también se llaman *instancias*

La relación entre un individuo y la clase se define mediante ***rdf:type***



El uso de clases permite declarar restricciones y evitar enunciados que no tienen sentido:

## Restricciones de Rango

Ej. Juan da clase de Pepe

La propiedad "da clase de" se asigna a asignaturas

## Restricciones de Dominio

Ej. Álgebra da clase de Cálculo

La propiedad "da clase de" se aplica a profesores



Las clases pueden organizarse en jerarquías

***subclassOf*** define que una clase es una subclase de otra

A es una subclase de B si todo individuo de A pertenece a B

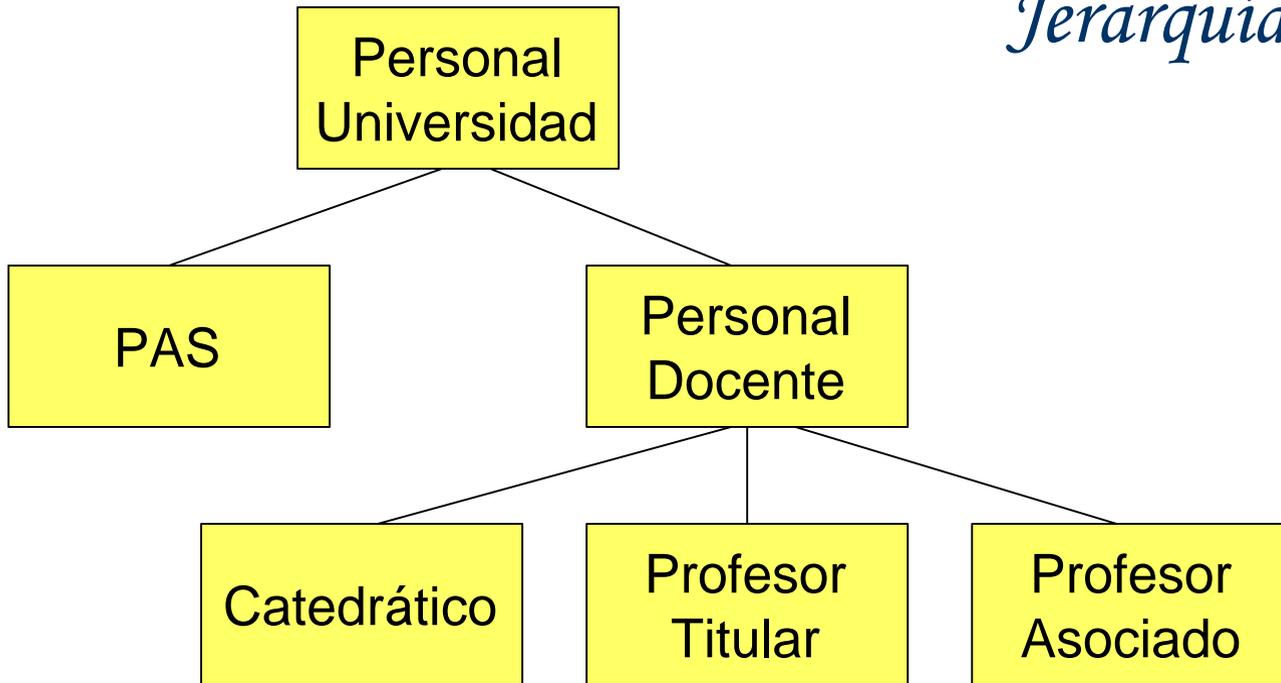
Entonces, B es una superclase de A

Una clase puede tener múltiples superclases



# RDF Schema

## Jerarquía de clases



### Herencia

*Un curso sólo puede impartirse por personal docente.*

*Pepe es Profesor Asociado*

*Pepe hereda la propiedad de poder impartir un curso de la clase "personal docente"*

La semántica de ***subclassOf*** está predefinida



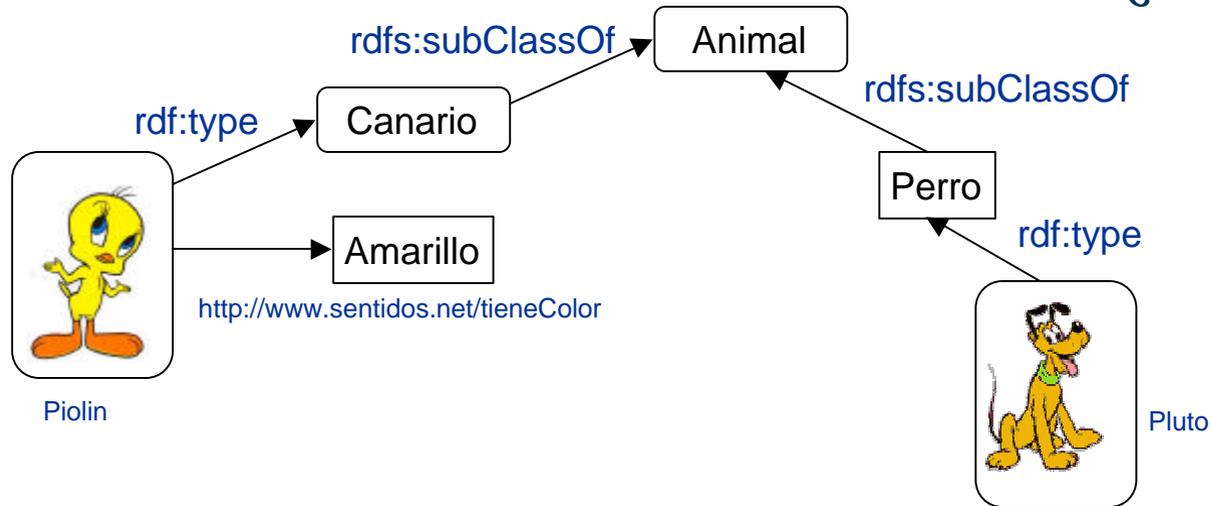
Jerarquías entre propiedades **subPropertyOf**

*Ej. Ser padre es una subpropiedad de ser progenitor*

*Si  $P$  es una subpropiedad de  $Q$  entonces,  $P(x,y)$  se cumple sólo si  $Q(x,y)$*



# RDF Schema Syntax XML



```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://www.sentidos.net/">
  <rdf:Description rdf:ID="Piolin">
    <s:tieneColor>Amarillo</s:tieneColor>
    <rdf:type resource="#Canario" />
  </rdf:Description>

  <rdf:Description rdf:ID="Canario"> <rdf:subClassOf rdf:resource="#Animal"/></rdf:Description>
  <rdf:Description rdf:ID="Pluto"><rdf:type rdf:resource="#Perro" /></rdf:Description>
  <rdf:Description rdf:ID="Perro"><rdf:subClassOf rdf:resource="#Animal" /></rdf:Description>
  <rdf:Description rdf:ID="Animal" />
</rdf:RDF>
```



# RDF Schema

## Restricciones

Propiedades básicas:

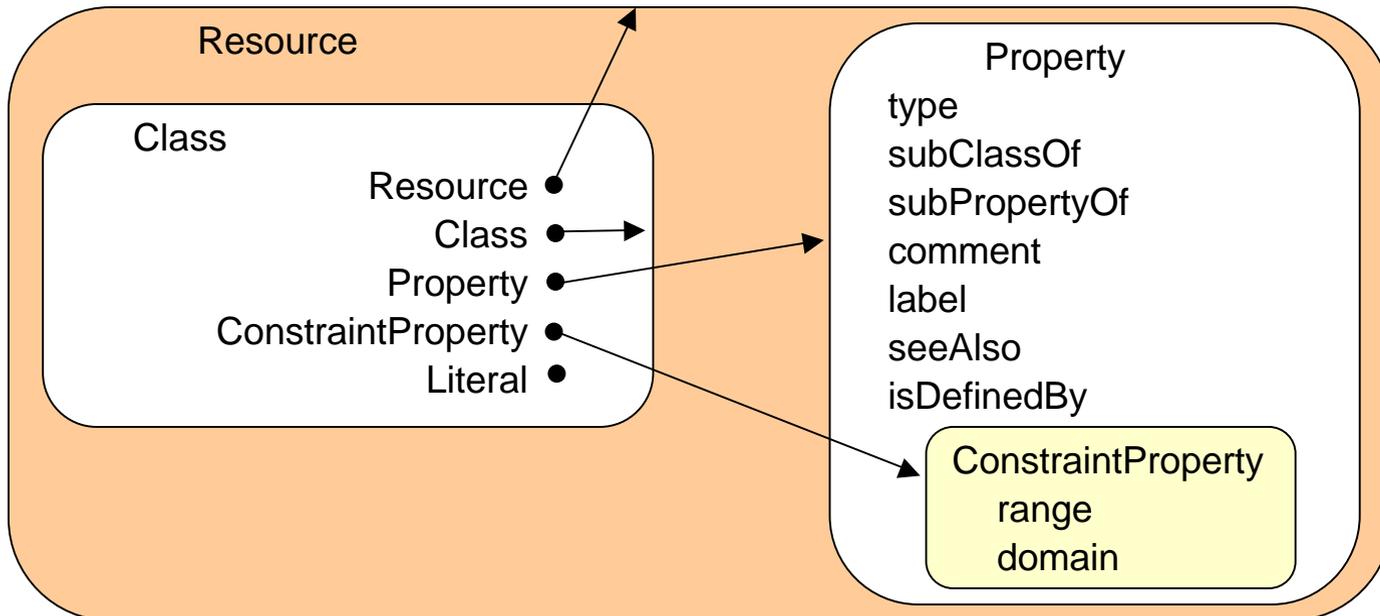
**type**: indica pertenencia (el valor debe ser instancia de class)

**subClassOf**: relación de subconjunto entre 2 clases

Relación transitiva

Una clase puede ser subclase de más de una clase (herencia múltiple)

Una clase no puede ser subclase de sí misma





## Otras propiedades

**seeAlso**: Especifica un recurso que puede proporcionar más información

**isDefinedBy**: Subpropiedad de **seeAlso** que permite indicar un recurso autor

**comment**: Asocia un comentario a un recurso

**label**: Asocia una etiqueta a un recurso

## Restricciones

**ConstraintResource**: Recurso que permite definir restricciones

**ConstraintProperty**: Propiedad que expresa una restricción

**range**: Se aplica a una propiedad. El valor debe ser una clase

**domain**: Se aplica a una propiedad. El valor debe ser una clase





# *Ontologías: OWL*



# Ontologías

El término *Ontología* se utiliza en filosofía como una *disciplina que estudia la naturaleza y organización de la realidad*

En Aristóteles (Metafísica, IV, 1) se define como *la ciencia del ser*

En Informática, se utiliza como un artefacto que define:

Un *vocabulario compartido* que describe un determinado dominio

Un *conjunto de supuestos* sobre los términos de dicho vocabulario, generalmente se utiliza un *lenguaje formal* manipulable automáticamente.



Normalmente las ontologías tienen dos componentes:

Nombres de conceptos

***Elefante:*** es un concepto cuyos miembros son animales

***Hervívoro:*** es un concepto cuyos miembros son exactamente aquellos animales que comen solamente plantas o partes de plantas

***Elefante\_Adulto:*** es un concepto cuyos miembros son elefantes que tienen una edad superior a 20 años

***Conocimiento de base o restricciones:***

Los ***Elefantes\_Adultos*** pesan más de 2000kg

Todos los ***Elefantes*** son elefantes africanos o indios

Ningún individuo es carnívoro y hervívoro



# Ontologías

## Ejemplos

Cyc (<http://www.cyc.com>).

Conceptos de sentido común para Inteligencia Artificial

Utiliza lógica de predicados mediante lenguaje CycL

Frame Ontology y OKBC Ontology

Disponibles en Ontolingua (<http://www-ksl-svc.stanford.edu/>)

Utiliza KIF (Knowledge Interchange Format)

Ontologías en campos concretos:

Lingüística: WordNet (<http://www.globalwordnet.org/>)

Medicina: GALEN (<http://www.opengalen.org/>)

etc.



# Ontologías para la Web

## Evolución

SHOE (Simple HTML Ontology Extensions) Univ. Maryland, 1996

Permite definir ontologías en documentos HTML

Objetivo = Facilitar búsquedas y anotaciones de documentos

XOL (XML Ontology exchange Language)

Intercambio de definiciones de ontologías entre sistemas

OIL (Ontology Inference Layer)

Sintaxis RDF(S) y primitivas de representación del conocimiento en marcos

Se basa en el uso de *description logics*

DAML (DARPA Agent Markup Language)

Proyecto americano de creación de lenguaje para ontologías

DAML-OIL. Proyecto conjunto que será la base de OWL

OWL (Web Ontology Language) desarrollado en W3C



Desarrollado por el consorcio W3C

3 niveles:

OWL Full. Unión de sintaxis OWL y RDF (sin restricciones)

No se garantiza la eficiencia ni siquiera la decidibilidad

OWL DL (Description Logics). Limita la expresividad intentando conseguir decidibilidad

Se pierde compatibilidad con RDF(S)

OWL Lite. Subconjunto de OWL DL más fácil de implementar (y se espera que más eficiente)

OWL DL se basa en sistema SHIQ de *description logics*

Semántica bien definida

Propiedades formales (decidibilidad, complejidad)

Algoritmos de razonamiento conocidos

Varios Sistemas que lo implementan



## Permite definir

- Clases (conceptos)

- Propiedades

  - Propiedades de tipos de datos

  - Propiedades de objetos

- Individuos (instancias)

Contiene una serie de predicados predefinidos para la definición de ontologías:

- clases equivalentes, *sameAs*, etc

Pueden definirse clases a partir de su descripción:

- Enumeraciones (*rojo, verde o azul*)

- Restricciones (*todos los individuos que tienen más de dos hijos*)

- Enunciados lógicos (*Persona que no es estudiante y tiene ojos azules*)



## Pertenencia a Clases

Si  $x$  pertenece a  $A$  y  $A$  es una subclase de  $B$ , entonces  $x$  pertenece a  $B$

## Equivalencia de clases

Si  $A$  es equivalente a  $B$  y  $B$  es equivalente a  $C$ , entonces  $A$  es equivalente a  $C$

## Consistencia

$x$  pertenece a  $A$  y a  $B$ , pero  $A$  y  $B$  son disjuntas  $\Rightarrow$  Error

## Clasificación

Si para pertenecer a una clase  $A$  es necesario cumplir ciertas propiedades y  $x$  cumple dichas propiedades, entonces  $x$  pertenece a  $A$



# *Razonamiento en OWL*

El soporte para razonamiento permite:

- Chequear la consistencia de la base de conocimiento

- Chequear que no haya relaciones no pretendidas entre clases

- Clasificar automáticamente individuos como elementos de clases

Objetivo

- Diseñar grandes ontologías por varios autores

- Integrar y compartir ontologías de diversas fuentes

OWL incluye un modelo semántico que permite la creación de sistemas de razonamiento

- La lógica descriptiva es un subconjunto de la lógica de predicados que permiten sistemas de razonamiento eficientes, ej. Fact y RACER



## *Limitaciones de RDF Schema*

Ámbito local de propiedades: No permite restricciones que apliquen sólo a algunas clases. Ej. *Las vacas sólo comen hierba*

Clases disjuntas. Ej. La clase hombre y mujer

Combinaciones booleanas de clases. Definir clases mediante unión, intersección, complementario, etc.

*Ej. Persona es la unión de hombre y mujer*

Restricciones de cardinalidad. Ej. Una persona sólo tiene 2 progenitores

Características de propiedades. Ej. transitividad, unicidad, inversa, etc.



## *Sintaxis OWL*

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

También existen otras formas sintácticas más sencillas

Las ontologías comienzan por owl:Ontology

```
<owl:Ontology rdf:about="">  
  <rdfs:comment>Ejemplo de Ontología</rdfs:comment>  
  <owl:priorVersion  
    rdf:resource="http://www.uniovi.es/viejo"/>  
  <owl:imports  
    rdf:resource="http://www.uniovi.es/personas"/>  
  <rdfs:label>Ontología de la Universidad</rdfs:label>  
</owl:Ontology>
```

**owl:imports** es una propiedad transitiva



Las clases se definen mediante owl:Class

owl:Class es una subclase de rdfs:Class

Las clases disjuntas se definen mediante owl:disjointWith

```
<owl:Class rdf:about="#ProfesorAsociado">  
  <owl:disjointWith rdf:resource="#catedrático"/>  
  <owl:disjointWith rdf:resource="#titular"/>  
</owl:Class>
```

Clases equivalentes mediante equivalentClass

```
<owl:Class rdf:ID="Profesor">  
  <owl:equivalentClass rdf:resource="#PersonalDocente"/>  
</owl:Class>
```

owl:Thing es la clase más general

owl:Nothing es la clase vacía



## Propiedades en OWL

Hay 2 tipos de propiedades

**Propiedades de Objetos** relacionan un objeto con otro objeto. ej. "daClaseDe"

```
<owl:ObjectProperty rdf:ID="daClaseDe">
  <owl:domain rdf:resource="#Profesor"/>
  <owl:range rdf:resource="#Asignatura"/>
  <rdfs:subPropertyOf rdf:resource="#estáRelacionadoCon"/>
</owl:ObjectProperty>
```

**Propiedades de tipos de datos** relacionan un objeto con valores de tipos de datos (enteros, literales, etc.), ej. "edad"

Habitualmente, se utilizan los tipos de datos de XML Schema

```
<owl:DatatypeProperty rdf:ID="edad">
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
</owl:DatatypeProperty>
```



## *Propiedades en OWL*

### Propiedades inversas (**inverseOf**)

$$P \text{ inverseOf } Q \Rightarrow P(x,y) \Leftrightarrow Q(y,x)$$

Ej. daClaseDe -- tieneProfesor

### Propiedades equivalentes: **equivalentProperty**

Ej. daClaseDe -- enseña



# *Propiedades en OWL*

## Clases como restricciones de propiedades

Se puede declarar que la clase C satisface ciertas condiciones (todos sus individuos deben satisfacerlas)

Es lo mismo que decir que C es una subclase de C' donde C' incluye todos los objetos que satisfacen dichas condiciones

```
<owl:Class rdf:about="#personalDocente">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#daClaseDe"/>  
      <owl:someValuesFrom rdf:resource="#Asignatura"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```



**allValuesFrom** ( $\forall$ ) indica que todos los valores deben ser de un tipo

NOTA: Los que no tiene ningún valor, también cumplen la condición

**someValuesFrom** ( $\exists$ ) Al menos un valor de la propiedad debe tener un tipo

Ejemplo: Un estudiante es una persona que cursa al menos una asignatura

**hasValue** Al menos uno de los valores tiene un valor

**minCardinality**, **maxCardinality** restringen el número máximo/mínimo de valores



## Propiedades en OWL

Propiedades especiales:

**TransitiveProperty**. Si  $P(x,y)$  y  $P(y,z)$  entonces  $P(x,z)$

Ej. *antepasado*

**SimmetricProperty**. Si  $P(x,y)$  entonces  $P(y,x)$

Ej. "hermano"

**FunctionalProperty**. Como mucho tiene un valor para cada objeto. Si  $P(x,y)$  y  $P(x,z)$  entonces  $y = z$

Ej. "edad"

**InverseFunctionalProperty**. Dos objetos diferentes no pueden tener el mismo valor. Si  $P(x,y)$  y  $P(z,y)$  entonces  $x = z$

Ej. "dni"



# *Propiedades en OWL*

## *Combinaciones booleanas*

Combinaciones booleanas

complementOf, unionOf, intersectionOf

```
<owl:Class rdf:ID="personasUniversidad">  
  <owl:unionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#personalDocente"/>  
    <owl:Class rdf:about="#estudiantes"/>  
    <owl:Class rdf:about="#PAS"/>  
  </owl:unionOf>  
</owl:Class>
```



# *Propiedades en OWL*

## *Enumeraciones*

oneOf

```
<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>
```



## *Individuos en OWL*

Se declaran igual que en RDF

```
<rdf:Description rdf:ID="949352">  
  <rdf:type rdf:resource= "#personalDocente"/>  
</rdf:Description>
```

```
<personalDocente rdf:ID="949352">  
  <uni:edad rdf:datatype="&xsd;integer">35<uni:edad>  
</personalDocente>
```



## *Otras Características de OWL*

### No se asume nombres únicos

Si 2 individuos tienen nombre diferente no quiere decir que sean diferentes

Ej. Si se declara que cualquier curso sólo tiene un profesor y que un curso tiene 2 profesores, entonces se infiere que ambos profesores son el mismo.

Para indicar que son diferentes, hay que declararlo explícitamente mediante "[differentFrom](#)", "[distinctMembers](#)"

OWL admite los tipos de datos predefinidos de XML

Schema pero no trabaja con los tipos definidos por el usuario

Información de versiones

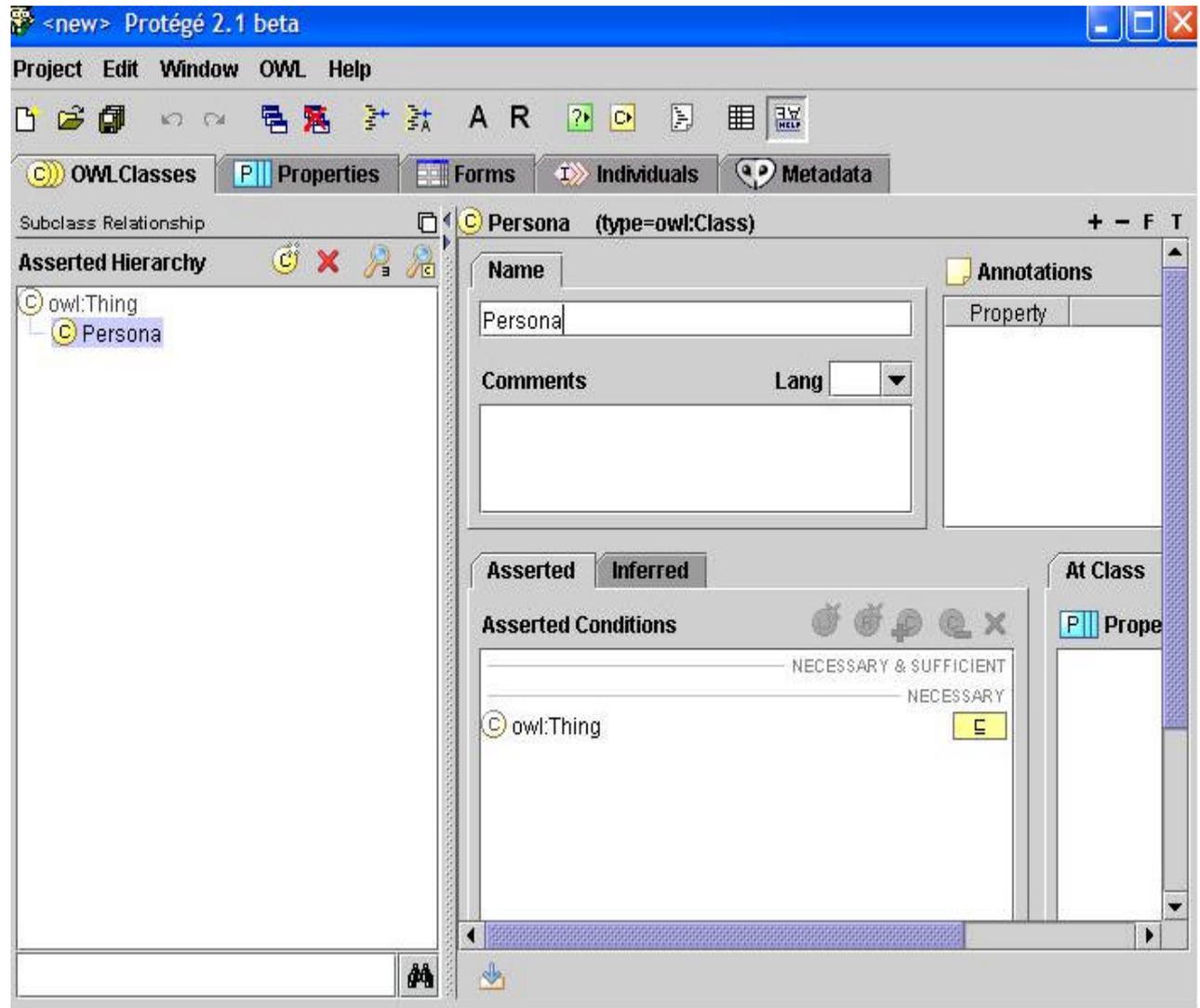


OILED

Ideagraph

Triple20

Protégè





## Sistema de módulos

Mecanismo de importación primitivo (sin ocultación de información)

No se pueden declarar *valores por defecto*

Adopción de *Open World Assumption* limita capacidad de razonamiento

No asunción de nombres únicos (individuos con diferentes nombres no tienen porqué ser diferentes)

Sistemas de reglas



# *Lógica y Sistemas de Inferencia*



# *Lógica Descriptiva*

La base de RDF(S) y de OWL es la lógica descriptiva

Es un subconjunto de lógica de predicados

Búsqueda de compromiso entre computabilidad y capacidad expresiva

Describe el dominio mediante clases (conceptos), propiedades e individuos

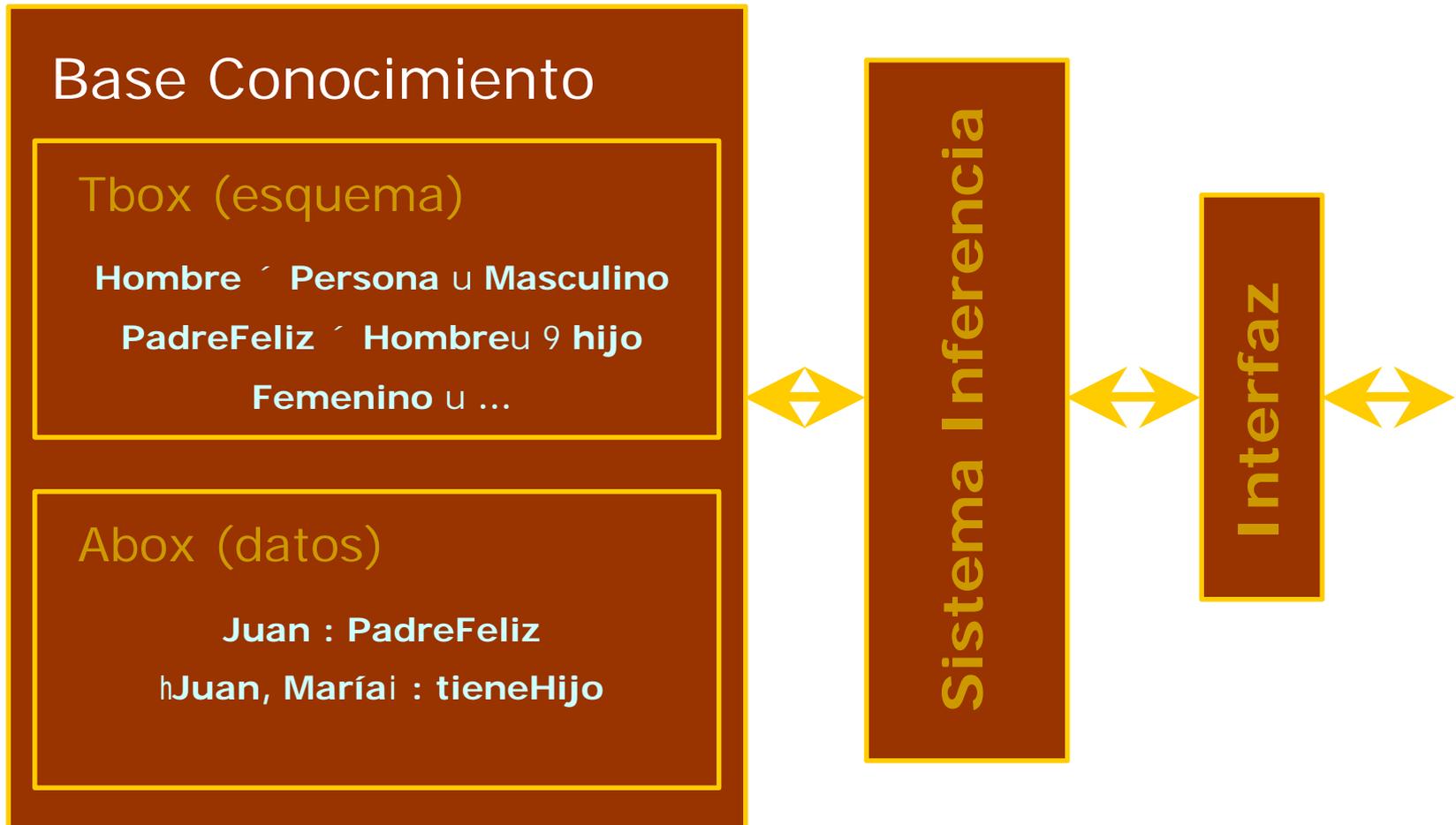
Semántica formal bien definida

- Demostraciones de consistencia y completud

- Varias implementaciones



# Lógica Descriptiva





# *Lógica Descriptiva*

La Base de Conocimiento está formada por:

TBox = Conjunto de axiomas que describen estructura del dominio (esquema de clases)

PadreFeliz  $\equiv$  Hombre  $\uparrow$   $\exists$ tieneHijo . Femenino  $\uparrow$  ...

Elefante  $\underline{\textcircled{2}}$  Animal  $\uparrow$  Grande  $\uparrow$  Gris

transitive(antepasado)

ABox = Conjunto de axiomas que describen una situación concreta

Juan : PadreFeliz

<Juan, María> : tieneHijo



Además de Lógica Descriptiva, existen otros subconjuntos de lógica: Cláusulas Horn

Las C. Horn son la base de la programación lógica

Existen implementaciones eficientes

## C. Horn vs. L. Descriptiva

En lógica descriptiva no se puede expresar que un estudiante que vive y estudia en el mismo sitio es un estudiante local

**estudia(X,Y), vive(X,Z), cerca(Y,U), cerca(Z,U) <sup>Ⓡ</sup>  
estudianteLocal(X)**

En C. Horn no se puede enunciar que una persona es un hombre o una mujer (pero no ambos)



# *Reglas Monótonas vs. no monótonas*

Ejemplo: Un vendedor *on-line* quiere hacer un descuento especial si es cumpleaños del cliente

Solución 1

*R1. Si cumpleaños entonces descuento*

*R2. Si no cumpleaños entonces no descuento*

¿Qué ocurre si una persona no declara cuándo es su cumpleaños?

*R2'. Si no se sabe si es cumpleaños entonces no descuento*

R2' no está en lógica de primer orden, se basa en información incompleta.



## *Sistemas de Inferencia*

CWM. Desarrollado por Tim Berners Lee en Python

Incluye sistema de inferencia

Sintaxis n3 y RDF

Euler. Sistema de inferencia Admite n3

SWISH (Semantic Web Inference System in Haskell)

JENA. API Java para RDF. Incluye sistema de inferencia

SwiProlog. Incluye librerías de Prolog y el sistema Triple20 que permite editar OWL

MetaLog. Basado en Prolog, permite sintaxis en pseudo-lenguaje natural

RACER. Sistema de inferencia implementado en Lisp

Funciona como un servicio http



2ª Parte

*Perspectivas y Aplicaciones de la Web Semántica*



## RSS 1.0 es un vocabulario de RDF

Creación de resúmenes de sitios Web (syndication)

NOTA: Existe RSS 0.92, 0.93 y 2.0 que nose basa en RDF

```
- <rdf:RDF>
- <channel rdf:about="http://www.webweavertech.com/ovidiu/weblog/">
  <title>Ovidiu Predescu's Weblog</title>
  <link>http://www.webweavertech.com/ovidiu/weblog/</link>
  <description>Technology ramblings</description>
  <dc:language>en-us</dc:language>
  <dc:creator>ovidiu</dc:creator>
  <dc:date>2004-03-02T20:17:36-08:00</dc:date>
  <admin:generatorAgent rdf:resource="http://www.movabletype.org/?v=2.64"/>
  <admin:errorReportsTo rdf:resource="mailto:ovidiu@webweavertech.com"/>
  <sy:updatePeriod>hourly</sy:updatePeriod>
  <sy:updateFrequency>1</sy:updateFrequency>
  <sy:updateBase>2000-01-01T12:00+00:00</sy:updateBase>
- <items>
- <rdf:Seq>
  <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000318.html"/>
  <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000317.html"/>
  <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000316.html"/>
  <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000314.html"/>
  <rdf:li rdf:resource="http://www.webweavertech.com/ovidiu/weblog/archives/000307.html"/>
</rdf:Seq>
```



# Aplicaciones de RDF

## FOAF

FOAF = Friend of a Friend (<http://rdfweb.org>)

Vocabulario para definir páginas Personales

The screenshot shows a web browser window with the title "FoaF Tools > Explorer". The main content area is titled "FoaF Explorer" and features a central graphic of a blue circle with three arrows forming a clockwise loop. Inside the circle are three stylized faces: a red one at the top, a purple one on the left, and a green one at the bottom. Dashed yellow lines connect the faces in a triangular pattern. Below the graphic is a dashed box containing the text "Explore a FoaF neighbourhood", a "Location:" label, an empty text input field, and an "Explore" button. At the bottom of the dashed box, the text "Example neighbourhoods" is partially visible.



# Aplicaciones de RDF

## Adobe XMP

XMP = eXtensible Metadata Platform

Incluir meta-información en imágenes

### Extensible Metadata Platform (XMP)

**XMP main**

**Downloads**

- [XMP SDK](#)
- [Custom File Info panels](#)

**XMP information**

- [Overview](#)
- [In depth](#)
- [Reviews and news](#)
- [Related applications](#)
- [Subscribe to the XMP newsletter](#)

**Support**

- [Developer forums](#)
- [Developer support](#)

Powered By



**Adding intelligence to media**

"As any content or production professional knows, developing a workflow that actually works can be a major challenge. Keeping track of important files and assets at each stage is critical. Effective file management is an important and necessary part of the creative process, but the available tools have never been adequate.

Adobe's Extensible Metadata Platform (XMP) is a labeling technology that allows you to embed data about a file, known as metadata, into the file itself. With XMP, desktop applications and back-end publishing systems gain a common method for capturing, sharing, and leveraging this valuable metadata — opening the door for more efficient job processing, workflow automation, and rights management, among many other possibilities. With XMP, Adobe has taken the "heavy lifting" out of metadata integration, offering content creators an easy way to embed meaningful information about their projects and providing industry partners with standards-based building blocks to develop optimized workflow solutions.

**Find out more: [What is XMP?](#)**

***"[XMP] is an important piece that brings the Semantic Web closer to realization."***

– Eric Miller, W3C Semantic Web Activity Lead



# *Aplicaciones de RDF*

## *Dublin Core*

Dublin Core Metadata Initiative (<http://www.dcmi.org>)

Utilizado para la catalogación de documentos

Espacio de nombres: <http://purl.org/dc/elements/1.1/>

Conjunto de elementos básicos cuyo significado es compartido

Contenido: [Coverage](#), [Description](#), [Type](#), [Relation](#), [Source](#), [Subject](#), [Title](#)

Propiedad Intelectual: [Contributor](#), [Creator](#), [Publisher](#), [Rights](#)

Instanciación: [Date](#), [Format](#), [Identifier](#), [Language](#)

Cada elemento básico admite una serie de cualificadores

Refinamiento de elementos

Ejemplo: [Date.created](#), [Description.tableOfContents](#)

Esquema de codificación

Ejemplos: [Identifier.URI](#), [Date.DCMIPeriod](#)



### vCard: Información personal (<http://www.imc.org>)

```
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:vCard = "http://www.w3.org/2001/vcard-rdf/3.0#">
<rdf:Description rdf:about = "http://www.mafia.it/#vito" >
  <vCard:FN>Vito Corleone</vCard:FN>
  <vCard:TITLE>Jefe de la pizzería</vCard:TITLE>
  <vCard:TEL rdf:parseType="Resource">
    <rdf:value>+985 203040 </rdf:value>
  </vCard:TEL>
  <vCard:EMAIL rdf:parseType="Resource">
    <rdf:value>vito@mafia.it</rdf:value>
  </vCard:EMAIL>
  <vCard:ADR rdf:parseType="Resource">
    <vCard:Street>C/ Génova Nº1 </vCard:Street>
    <vCard:Locality>Oviedo </vCard:Locality>
    <vCard:Country>España</vCard:Country>
  </vCard:ADR>
</rdf:Description>
</rdf:RDF>
```



# *Web Semántica + Servicios Web*

Reto: Descripción de servicios Web mediante tecnologías de Web Semántica

WSDL describe la interfaz

...pero es necesario otro tipo de descripciones:

- Descubrimiento automático

- Composición e interoperación entre servicios

- Monitorización de servicios

- Procesos

- Calidad de Servicio

- etc.

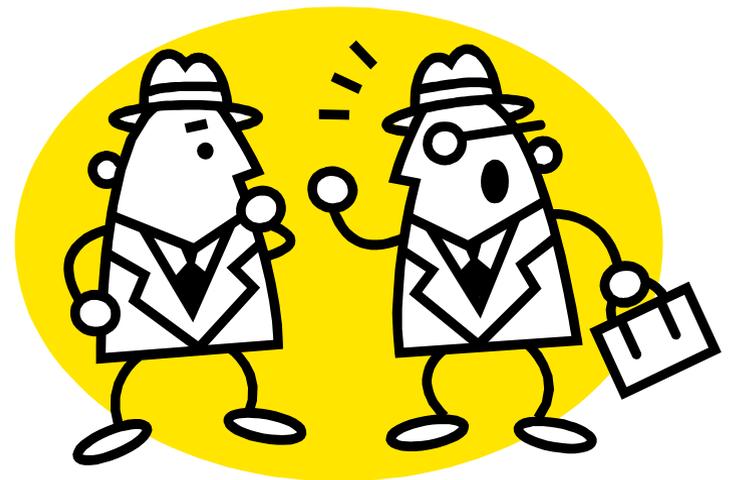
Primeros pasos: OWL-S

<http://www.daml.org/services/owl-s/1.0/>



# *Mini-Debate*

## *Aplicaciones de la Web Semántica*





## *Selección de Enlaces*



Página del consorcio: <http://www.w3c.org/RDF>

SemanticWeb: <http://www.semanticweb.org>

Directorio de Ontologías: <http://www.schemaweb.info>

Ontologías: <http://www.ontology.org>

Dublin Core: <http://www.dcmi.org>

Darpa Markup Language: <http://www.daml.org>

Open Directory Project: <http://www.dmoz.org>

OntoWeb: <http://www.ontoweb.org>

Topic Maps: <http://easytopicmaps.com>



*Fin de la Presentación*

