

# Ingeniería de Aplicaciones Web

*Diego C. Martínez*

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur

# La web y sus versiones

## Web 1.0



## Web 2.0



# Web 2.0



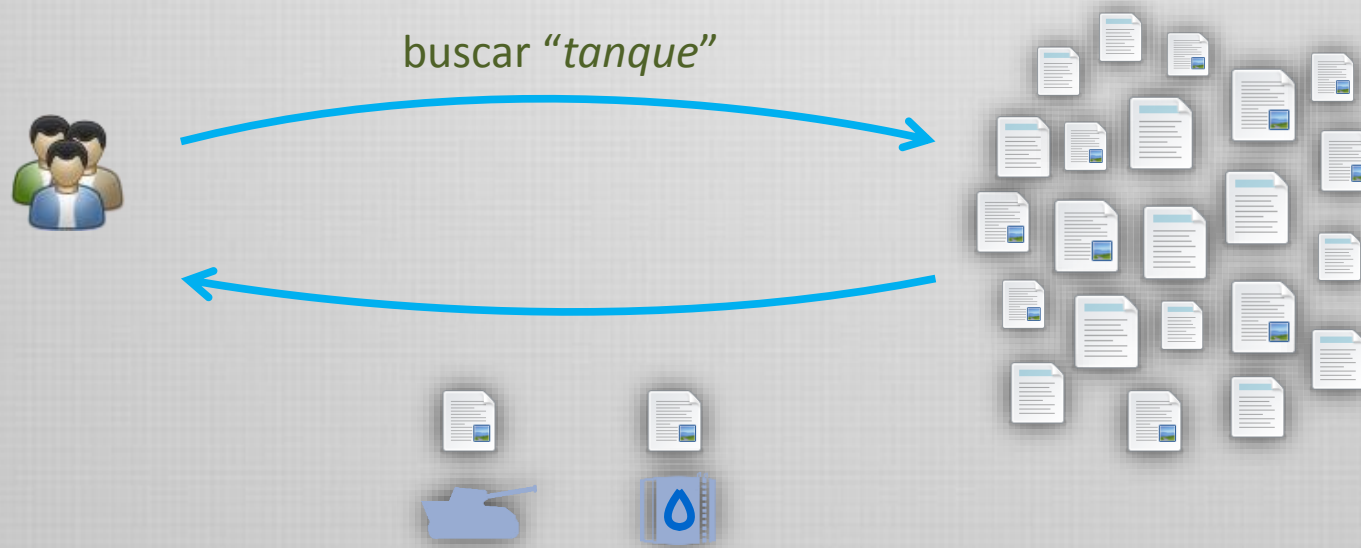
# Web 3.0

Un problema (previsible) es el constante crecimiento de la información en la red.

*Esto es acentuado en parte por el surgimiento de la Web 2.0*

El principal inconveniente es la falta de estructuración de la información.

Es fácil de obtener, pero no fácil de interpretar.



Una solución es descubrir el contexto de cada página según la búsqueda deseada.

*Este es el arte de la búsqueda en la web, la ingeniería de textos y data mining.*

Otra solución es representar contenido web incluyendo *información semántica*.

# Web semántica

Representar contenido web incluyendo *información semántica es, en parte, una iniciativa de Tim Berners Lee.*

*De hecho, se corresponde más con la propuesta original de la World Wide Web.*

*“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in co-operation.”*

Berners-Lee et al., 2001

La Web Semántica puede verse como

- Una evolución de la web actual
- Una tecnología de metadatos orientada al software empresarial
- Una nueva área de la Inteligencia Artificial

Son un conjunto de tecnologías para que la interacción con la Web sea

*comprensible, en sus términos y contexto  
interoperable, facilitando la comunicación con otros sistemas  
programable, permitiendo la intervención de agentes inteligentes  
con contenido filtrable, limitando la intervención humana  
proveedora de servicios, para ser consumido por otras aplicaciones*



# Web semántica

Limitaciones de la tecnología actual, débilmente estructurada:

- Búsqueda de información: realizada por motores basados en palabras claves.
- Extracción de información: requiere supervisión humana ¿que información es relevante?
- Mantenimiento de información: existen muchas inconsistencias y falta de depuración de información desactualizada.
- Descubrimiento de nueva información: requiere técnicas algo complejas y no muy adecuadas para información débilmente estructurada

La web semántica se orienta a

- Organizar el conocimiento de acuerdo a su significado.
- Permitir la automatización del control de inconsistencias y extracción de conocimiento.
- Reemplazar la búsqueda basada en palabras claves por el cálculo de consultas complejas (condiciones mas ricas, busquedas en varios documentos, etc).

# Datos interpretables

빌 게이츠

اسم

1955년 10월 28일

عيد ميلاد

는 미국의 기업인이다. 어렸을 때부터 컴퓨터 프로그램을 만드는 것을 좋아했던 그는 하버드 대학교를 다니다가 자퇴하고 폴 앨런과 함께 마이크로소프트를 공동 설립했다. 두 사람은 1970년대 중반 앨테어 8800에서 동작하는 Altair Basic 인터프리터를 고안했다. Altair는 상업적인 성공을 거둔 최초의 개인용 컴퓨터로 평가받는다. 다트머스 대학교에서 학습용으로 개발된 배우기 쉬운 컴퓨터 프로그래밍 언어 베이식에서 영감을 얻어, 폴 앨런과 함께 새로운 BASIC 버전을 개발하여 MS-DOS의 핵심적 프로그램 언어로 채택했다. 이후 1990년대에 개인용 컴퓨터의 보급이 급속히 증가하기 시작하는데 힘입어 MS-DOS의 지위는 공고해졌고 마이크로소프트는 개인용 컴퓨터 소프트웨어 시장의 주도권을 얻게 되었다. 이후 개인용 컴퓨터를 위한 운영 체제인 윈도우 95를 발표하여 대성공을 거두며 세계 최고의 부호로 등극하였다. 지금은 엄청난 돈과 명예로 남을 돕는 사업도 하고 있다. 2007년 6월 7일 명예졸업장을 받았다. 2008년 1월 24일에는 스위스 다보스 세계경제포럼 기조연설에서 기업에게 복지의 의무를 주장하는 창조적 자본주의를 주장하였다.

빌 게이츠는 2000년 1월 마이크로소프트의 최고 경영자 직책에서 물러났으며, 이후 회장직과 더불어 최고 소프트웨어 아키텍트(chief software architect) 직책을 신설하여 맡았다. 2008년 6월 27일에 그는 공식적으로 마이크로소프트에서 퇴임하고 빌 & 멜린다 게이츠 재단에서 무급으로 폴

بيان السيرة

# Datos interpretables

빌 게이츠

nombre

1955년 10월 28일

fecha de nacimiento

는 미국의 기업인이다. 어렸을 때부터 컴퓨터 프로그램을 만드는 것을 좋아했던 그는 하버드 대학교를 다니다가 자퇴하고 폴 앨런과 함께 마이크로소프트를 공동 설립했다. 두 사람은 1970년대 중반 앨테어 8800에서 동작하는 Altair Basic 인터프리터를 고안했다. Altair는 상업적인 성공을 거둔 최초의 개인용 컴퓨터로 평가받는다. 다트머스 대학교에서 학습용으로 개발된 배우기 쉬운 컴퓨터 프로그래밍 언어 베이식에서 영감을 얻어, 폴 앨런과 함께 새로운 BASIC 버전을 개발하여 MS-DOS의 핵심적 프로그램 언어로 채택했다. 이후 1990년대에 개인용 컴퓨터의 보급이 급속히 증가하기 시작하는데 힘입어 MS-DOS의 지위는 공고해졌고 마이크로소프트는 개인용 컴퓨터 소프트웨어 시장의 주도권을 얻게 되었다. 이후 개인용 컴퓨터를 위한 운영 체제인 윈도우 95를 발표하여 대성공을 거두며 세계 최고의 부호로 등극하였다. 지금은 엄청난 돈과 명예로 남을 돕는 사업도 하고 있다. 2007년 6월 7일 명예졸업장을 받았다. 2008년 1월 24일에는 스위스 다보스 세계경제포럼 기조연설에서 기업에게 복지의 의무를 주장하는 창조적 자본주의를 주장하였다.

빌 게이츠는 2000년 1월 마이크로소프트의 최고 경영자 직책에서 물러났으며, 이후 회장직과 더불어 최고 소프트웨어 아키텍트(chief software architect) 직책을 신설하여 맡았다. 2008년 6월 27일에 그는 공식적으로 마이크로소프트에서 퇴임하고 빌 & 멜린다 게이츠 재단에서 무급으로 폴

curriculum



# Web semántica

Algunos escenarios que se benefician con la web semántica

## Business to consumer:

*¿cuál es la mejor oferta para un determinado producto?*

Podemos dedicar horas a la navegación manual o utilizar alguna aplicación de ayuda (agentes de software).

- *Los agentes utilizan heurísticas para recuperar la información en cada página.*



### MercadoLibre

```
...<a href="http://articulo.mercadolibre.com.ar/..." class="list-view-
item-figure"></a>
<p class="price-info"><span class="price-info-cost"><strong
class="price">$&nbsp;1.000<sup>00</sup>    </strong></span>
<span class="price-info-installments">...
```



### eBay

```
...<td class="dt1"><div class="ttl">
<a href="http://cgi.ebay.com/Rossignol-165cm-Skis-Mint-Condition-No-Reserve-
/250841655905?pt=Skiing&hash=item3a6753ea61" class="vip">Rossignol 165cm
Skis</a></div><div class="dyn dynS"><div class="s1 empty"><div
class="mWSpc"></div>&nbsp;</div></div></td><td class="trs"></td><td
class="bids bin1">0 Bids</td><td class="prc g-b">$49.99</td>...
```

# Web semántica

Algunos escenarios que se benefician con la web semántica

## Wikis:

### *Creación y gestión colaborativa de conocimiento*

La información no está fuertemente estructurada semánticamente.

- Para buscar ciudades de Argentina, debe existir una página explícita que liste las ciudades.
- Si, por ejemplo, Ing. Jacobacci no está en la lista será imposible encontrarla, aún cuando la página de Jacobacci *indique* que está ubicada en Argentina.

## Sistemas de recomendación:

### *Búsqueda y gestión automatizada de información on-line*

- ¿Cuál es la mejor ruta para ir de Bahía Blanca a Chilecito?
- ¿Cuál es la ruta más corta?
- ¿Cuál es la ruta con más atractivos naturales?
- ¿Cuál es la de menor tráfico?
- ¿Cuál es la de mejores hoteles en paradas intermedias?

# Tecnologías de la web semántica

Las tecnologías asociadas a la iniciativa de la web semántica procuran enriquecer y/o estructurar la información ofrecida.

El objetivo esencial es **describir los datos**, para **comprenderlos**.

- **XML – eXtensible Markup Language**

Como lenguaje independiente de la plataforma para la descripción de metadatos

- **RDF – Resource Description Framework**

Como modelo de datos para objetos y sus relaciones entre sí.

- **OWL – Web Ontology Language**

Como lenguaje de representación de conocimiento general

Estos son especialmente importantes al describir conceptualizaciones.

## **ontología**

En filosofía, es el estudio de la naturaleza de la existencia.

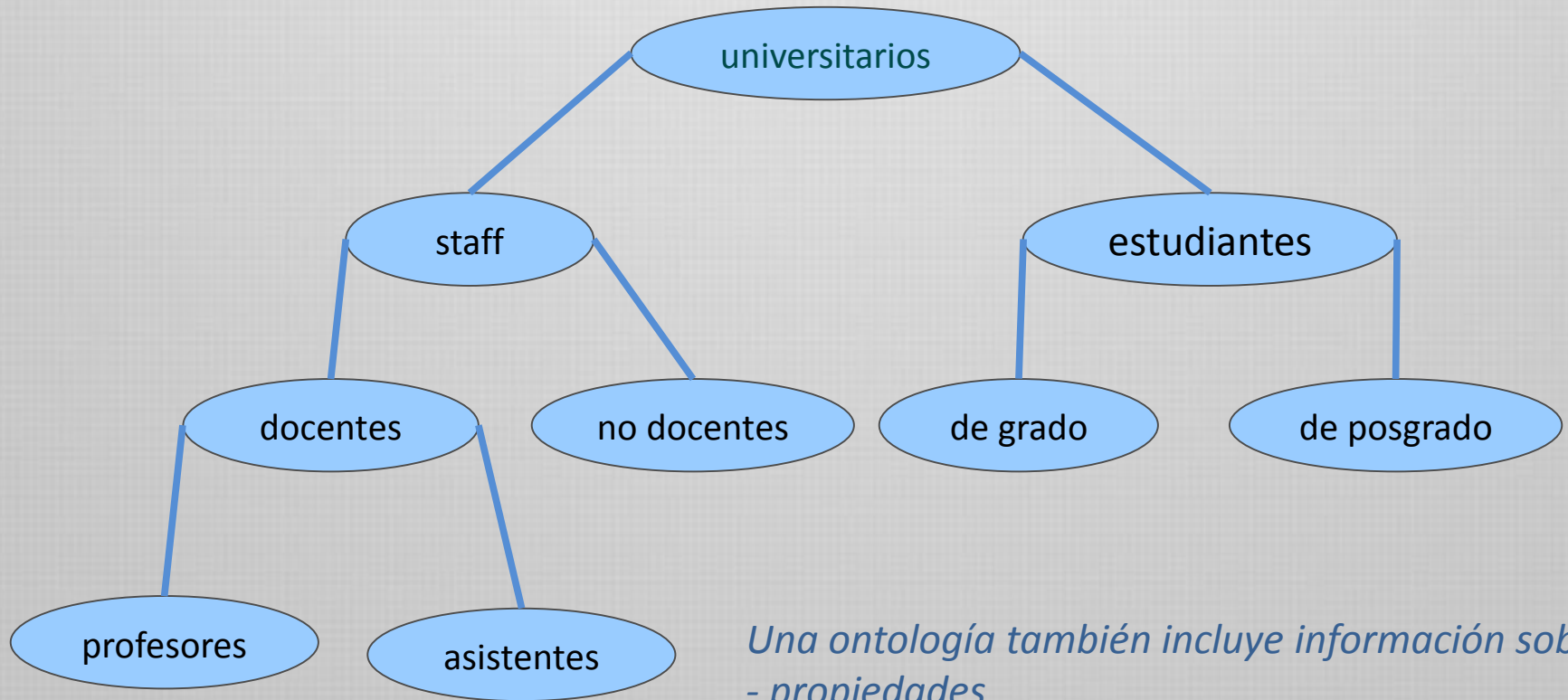
Se ocupa de identificar, en términos generales, las clases de cosas que existen y cómo describirlas.

En nuestra área, una ontología es una **descripción concreta y formal de una conceptualización completa**

# Ontologías

Una ontología consiste típicamente de

- una lista finita de términos, denotando conceptos  
*(universitarios, profesores, estudiantes, alumnos regulares, etc)*
- las relaciones entre esos términos, usualmente jerarquías entre clases  
*(todos los alumnos regulares son estudiantes)*



*Una ontología también incluye información sobre:*

- propiedades
- restricciones de valores
- relaciones lógicas entre objetos



# Utilidad de las ontologías

## Algunos casos de uso de ontologías para la web (según W3C)

- Portales Web  
*Reglas de categorización utilizadas para mejorar la búsqueda*
- Colecciones Multimedia  
*Búsquedas basadas en contenido para medios no textuales*
- Administración de Sitios Web Corporativos  
*Organización taxonómica automatizada de datos y documentos*  
*Asignación entre Sectores Corporativos*
- Documentación de Diseño  
*Explicación de partes "derivadas" (p.ej. el ala de un avión)*  
*Administración explícita de Restricciones*
- Agentes Inteligentes  
*Expresión de las Preferencias y/o Intereses de los usuarios*  
*Mapeo de contenidos entre sitios Web*
- Servicios Web y Computación Ubicua  
*Composición y Descubrimiento de Servicios Web*  
*Administración de Derechos y Control de Acceso*



# Ontologías - RDF



RDF es un lenguaje de especificación de datos.

Significa *Resource Description Framework* y es un estándar de la W3C.

*Provee metadatos sobre los recursos web.*

Consiste de ternas o sentencias de la forma

**<objeto, atributo, valor>**

***Tolkien escribió El Señor de los Anillos***

La sintaxis de RDF es la de XML.

Los conceptos fundamentales de RDF son

- **Recursos (resources)** : objetos, cosas, tales como *autores, libros, empleados, autos*, etc.  
Cada recurso es asociado a un URI – Uniform Resource Identifier  
Puede ser una URL u otra forma unívoca de identificar un recurso.  
*El URL no necesariamente permite acceder al recurso.*
- **Propiedades (properties)**: una clase especial de recursos.  
Identifica relaciones entre recursos (“escrito por”, “trabaja en”, “es un”, etc)  
*También son identificadas por un URI, proveyendo un esquema de nombres único y global.*
- **Sentencias (statements)**: ternas que formalizan las propiedades de los recursos.

# Ontologías – RDF - ejemplo

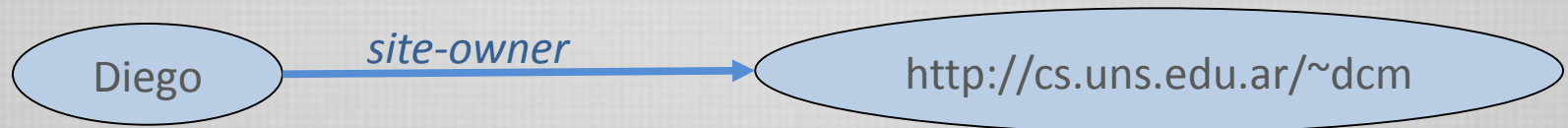
Un ejemplo de una sentencia puede ser:

*Diego es el dueño de <http://cs.uns.edu.ar/~dcm>*

Que puede interpretarse como un predicado lógico  $P(x,y)$

*SiteOwner(Diego, <http://cs.uns.edu.ar/~dcm>)*

O un grafo denotando una *red semántica*



O una terna *objeto-atributo-valor*

```
( http://www.cs.uns.edu.ar/~dcm,  
  http://www.mydomain.org/site-owner, #Diego )
```

# Ontologías – RDF - ejemplo

Una posible descripción de esta sentencia en XML es la siguiente:

```
<?xml version="1.0" encoding="UTF-16"?>
<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.cs.uns.edu.ar/~dcm">
    <mydomain:site-owner rdf:resource="#Diego"/>
  </rdf:Description>
</rdf:RDF>
```

Pueden existir más de un elemento **rdf:Description**

*el atributo about referencia un recurso existente*  
*un atributo ID especifica un nuevo recurso*

La propiedad es un tag dentro del elemento **rdf:Description**

## Ontologías – RDF - ejemplo

```
<rdf:Description rdf:about="77777">
  <rdf:type rdf:resource="&uni;curso"/>
  <uni:courseName>
    Ingenieria de Aplicaciones Web
  </uni:courseName>
  <uni:isTaughtBy rdf:resource="12345"/>
</rdf:Description>
```

```
<rdf:Description rdf:about="12345">
  <rdf:type rdf:resource="&uni;profesor"/>
  <uni:name>Diego Martinez</uni:name>
  <uni:title>Profesor Adjunto</uni:title>
</rdf:Description>
```

```
<rdf:Description rdf:about="77765">
  <uni:courseName>Inteligencia Artificial</uni:courseName>
  <uni:isTaughtBy>
    <rdf:Description rdf:about="56789">
      <uni:name>Guillermo Simari</uni:name>
      <uni:title>Profesor Titular</uni:title>
    </rdf:Description>
  </uni:isTaughtBy>
</rdf:Description>
```

# Ontologías – RDF Schema

RDF es un lenguaje universal que permite a los usuarios describir recursos utilizando su propio vocabulario.

*No asume nada de ningún dominio de aplicación ni define su semántica*

*Para eso se utiliza RDF Schema*

*Es un lenguaje para la representación de conocimiento.*

*Define los vocabularios que se utilizan en un RDF, principalmente indicando clases y sus relaciones. Funciona como una definición de tipo de datos.*

*Evita definir cosas como*

*Lenguajes de Programación es dictada por Informática I*

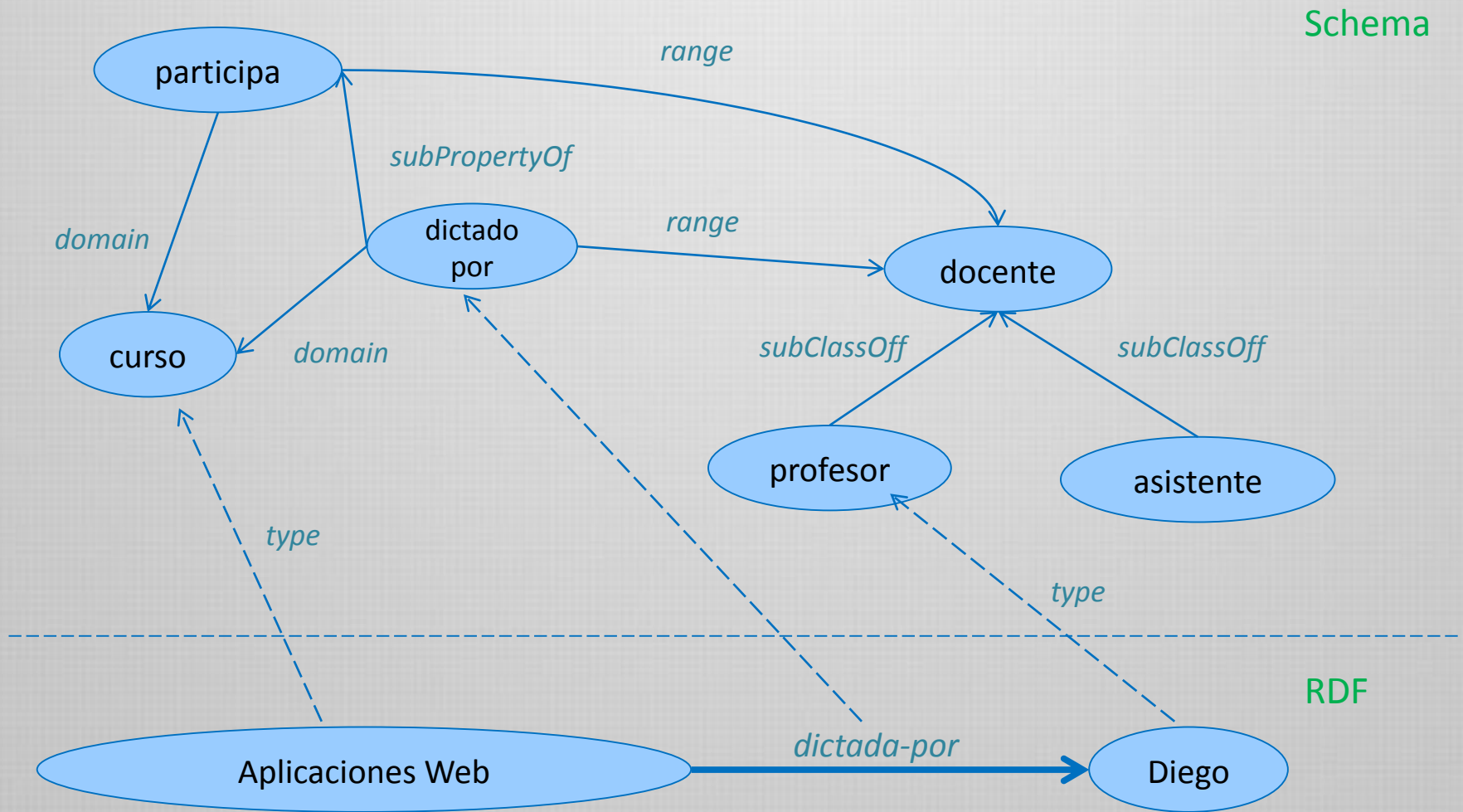
## RDF Schema

- permite definir a los recursos como instancias de una o más clases.
- permite que las clases puedan ser organizadas en forma jerárquica.

*RDF schema extiende a RDF para incluir un vocabulario con significado adicional.*



# Ontologías – RDF Schema



# Ontologías – RDF Schema

RDF incluye elementos que definen estas propiedades:

- **rdfs:Class** es la clase de todas las clases
- **rdfs:Resource** es la clase de todos los recursos
- **rdfs:Literal** es la clase de todos los literales (strings)
- **rdfs:Property** es la clase de todas las propiedades
- **rdfs:Statement** es la clase de todas las sentencias

Las propiedades básicas son:

- **rdfs:type** relaciona un recurso con su clase
- **rdfs:subClassOf** relaciona una clase con una de sus superclases.
- **rdfs:subPropertyOf** relaciona una propiedad con una de sus superpropiedades

```
<rdfs:Class rdf:about="profesor">  
  <rdfs:subClassOf rdf:resource="docente"/>  
</rdfs:Class>
```

# Ontologías – RDF Schema

## Propiedades restrictivas:

- **rdfs:domain** especifica el dominio de una propiedad P e indica que cualquier recurso al que se le dá determinada propiedad es instancia de las clases dominio.
- **rdfs:range** especifica el rango de una propiedad P e indica que los valores de la propiedad son instancia de las clases rango

```
<rdf:Property rdf:ID="participa">  
  <rdfs:domain rdf:resource="#curso"/>  
  <rdfs:range rdf:resource="#docente"/>  
</rdf:Property>
```

```
<rdf:Property rdf:ID="dictado-por">  
  <rdfs:subPropertyOf rdf:resource="#participa"/>  
</rdf:Property>
```

# FOAF

FOAF es un vocabulario definido en RDF.

Significa *Friend of a Friend* y procura crear una red de datos sobre personas, comprensible por las máquinas.

```
<foaf:Person rdf:about="#danbri" xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <foaf:name>Dan Brickley</foaf:name>
  <foaf:homepage rdf:resource="http://danbri.org/" />
  <foaf:openid rdf:resource="http://danbri.org/" />
  <foaf:img rdf:resource="/images/me.jpg" />
</foaf:Person>
```

Existe una [foaf:Person](#) con un [foaf:name](#) 'Dan Brickley'; cuya página es [foaf:homepage](#) y posee una relación [foaf:openid](#) con algo llamado [http://danbri.org/](#) y una relación [foaf:img](#) con algo llamado [/images/me.jpg](#)

Puede agregarse esta información en nuestra web:

```
<link rel="meta" type="application/rdf+xml" title="FOAF"
      href="http://sitio.com/~yo/foaf.rdf"/>
```

# FOAF

FOAF define clases y propiedades centradas en las personas, sus características y relaciones sociales.

## FOAF CORE

Agent  
Person  
name  
title  
img  
depiction (depicts)  
familyName  
givenName  
knows  
based\_near  
age  
made (maker)  
primaryTopic (primaryTopicOf)  
Project  
Organization  
Group  
...

## SOCIAL WEB

nick  
mbox  
homepage  
weblog  
openid  
jabberID  
interest  
topic\_interest  
topic (page)  
workplaceHomepage  
workInfoHomepage  
schoolHomepage  
publications  
currentProject  
pastProject  
account  
OnlineAccount  
accountName  
accountServiceHomepage  
PersonalProfileDocument  
tipjar  
sha1  
thumbnail  
logo



# FOAF

```
<rdf:RDF xmlns:rdf="...">
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:resource="#me"/>
  <foaf:primaryTopic rdf:resource="#me"/>
</foaf:PersonalProfileDocument>
  <foaf:Person rdf:ID="me">
    <foaf:name>Diego Martinez</foaf:name>
    <foaf:title>Dr</foaf:title>
    <foaf:givenname>Diego</foaf:givenname>
    <foaf:family_name>Martinez</foaf:family_name>
    <foaf:mbox rdf:resource="mailto:dcm@cs.uns.edu.ar"/>
    <foaf:homepage rdf:resource="cs.uns.edu.ar/~dcm"/>
    <foaf:phone rdf:resource="tel:4595135"/>
    <foaf:workplaceHomepage rdf:resource="cs.uns.edu.ar/~dcm"/>
    <foaf:workInfoHomepage rdf:resource="cs.uns.edu.ar"/>
    <foaf:knows>
      <foaf:Person>
        <foaf:name>Leo De Matteis</foaf:name>
        <foaf:mbox rdf:resource="mailto:ldmatte@cs.uns.edu.ar"/>
        <rdfs:seeAlso rdf:resource="cs.uns.edu.ar/~ldmatte"/>
      </foaf:Person>
    </foaf:knows>
  </foaf:Person>
</rdf:RDF>
```

# SPARQL

SPARQL es un lenguaje de consulta para datos en RDF

*Es un acrónimo recursivo: **SPARQL Protocol And RDF Query Language**.*

Dado que RDF es básicamente un grafo dirigido, SPARQL define sentencias de consultas de datos basadas en patrones de ternas.

*select-from-where*

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    ?person foaf:name ?name .
}
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT *
WHERE {
    ?person foaf:name ?name .
    ?person foaf:mbox ?email .
}
```

# Ontologías - OWL

RDF y RDF Schema son naturalmente de poder expresivo limitado.

- *RDF se basa en predicados binarios*
- *RDF Schema se basa en jerarquías de clases y propiedades con dominios y rangos definidos*

OWL es un lenguaje de representación de conocimiento para el modelado de ontologías

*Al igual que RDF, se utiliza cuando la información contenida en los documentos necesita ser procesada por aplicaciones, no presentada a humanos*

OWL extiende RDFS para permitir

- la expresión de relaciones complejas entre diferentes clases RDFS
- mayor precisión en las restricciones de clases y propiedades específicas.

Existen tres versiones de OWL:

- OWL Full: es la unión de la sintaxis OWL con RDF.
- OWL DL: es la restricción de OWL correspondiente a lógicas descriptivas.
- OWL Lite: un subconjunto de OWL, fácil de implementar.

# Ontologías - OWL

La sintaxis de OWL es la misma que RDF utilizando XML.

*Incluye elementos de RDF y agrega propios, enriqueciendo la descripción semántica*

- Aspectos de RDF Schema

*Class, rdfs:subClassOf, Individual  
rdf:Property, rdfs:subPropertyOf  
rdfs:domain, rdfs:range*

- Igualdades

*sameClassAs, samePropertyAs, sameIndividualAs  
differentIndividualFrom*

- Cardinalidad restringida

*minCardinality, maxCardinality  
cardinality*

- Características de propiedades

*inverseOf, TransitiveProperty, SymmetricProperty  
FunctionalProperty, InverseFunctionalProperty  
allValuesFrom, someValuesFrom*

# Ontologías – OWL - ejemplos

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

```
<owl:Class rdf:ID="vegetal">
  <owl:disjointWith rdf:resource="#animal"/>
</owl:Class>
```

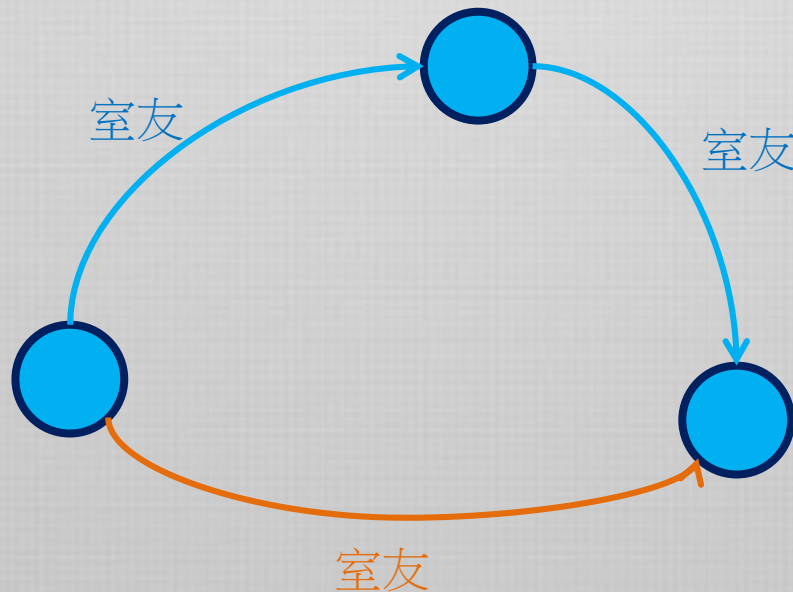
```
<owl:Class rdf:ID="arbol">
  <rdfs:subClassOf rdf:resource="#vegetal"/>
</owl:Class>
```

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



# Ontologías – OWL - ejemplos

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



# Ontologías – OWL - ejemplos

```
<owl:Class rdf:ID="herbivoro">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#animal"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#come"/>
      <owl:allValuesFrom>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#vegetal"/>
            <owl:Restriction>
              <owl:onProperty rdf:resource="#is_part_of"/>
              <owl:allValuesFrom rdf:resource="#vegetal"/>
            </owl:Restriction>
          </owl:unionOf>
        </owl:Class>
      </owl:allValuesFrom>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```