

# **TFC: XML y Web Semántica**

**Estudio del Impacto de las aplicaciones  
comerciales basadas en tecnologías de Web  
Semántica**

**Ingeniería Técnica Informática de Sistemas**

**TFC: XML y Web Semántica [UOC]**

**Curso 2007-2008**

**Ricardo García Ruiz  
Ingeniería Técnica Informática de Sistemas  
Consultor: Sinuhe Arroyo Gómez**



## INDICE

Descripción del TFC.....	8
1 Introducción.....	10
1.1 Objetivos específicos del TFC.....	10
1.2 Características específicas del TFC.....	11
1.3 Características secundarias del estudio comparativo a desarrollar.....	12
1.4 Cronología de desarrollo del TFC.....	13
1.5 Temporización.....	14
1.6 Diagrama de Gantt.....	17
2 Tecnologías relacionadas con la Web Semántica.....	18
2.1 XML.....	18
2.1.1 Los lenguajes de marcado.....	18
2.1.2 SGML (Standard Generalized Markup Language).....	18
2.1.3 HTML (HyperText Markup Language).....	19
2.1.4 XML (eXtensible Markup Language).....	19
2.1.5 Breve comparativa de HTML y XML.....	20
2.1.6 Características básicas de XML.....	21
2.1.7 Limitaciones de XML con los metadatos.....	22
2.2 RDF (Resource Description Framework).....	23
2.2.1 Comprensión de qué RDF.....	23
2.2.2 El Modelo Básico RDF.....	24
2.2.3 XML como Lenguaje de Especificación de la Sintaxis RDF.....	25
2.2.4 Contenedores en RDF.....	26
2.2.5 Reificación en RDF (sentencias sobre sentencias).....	27
2.3 RDF Shema.....	28
2.3.1 De RDF a RDF Schema.....	28
2.3.2 Clases, propiedades y restricciones.....	30
2.4 OWL (Web Ontology Language).....	31
2.5 WSML (Web Services Modeling Language).....	33
3 La Web Semántica.....	34
3.1 Introducción a la Web Semántica.....	34
3.2 Arquitectura de la Web Semántica.....	36
3.3 Ventajas e inconvenientes de la Web Semántica.....	37
3.4 Servicios Web.....	38
3.4.1 Introducción e historia.....	38
3.4.2 Breve descripción de la Arquitectura.....	40
3.4.3 Ventajas e inconvenientes de los Servicios Web.....	41
3.5 Servicios Web Semánticos.....	43
3.5.1 Web Ontology Language for Services (OWL-S).....	43
3.5.2 Web Services Modeling Ontology (WSMO).....	45
3.5.3 Semantic Web Services Framework (SWSF).....	46
3.5.4 Web Service Semantics (WSDL-S).....	47
3.5.5 Semantic Annotations for Web Services Description Language (SAWSDL).....	48
3.5.6 Ventajas e inconvenientes de los SWS.....	49

4	Ontologías.....	50
4.1	Definición de Ontología.....	50
4.2	Tipos de Ontologías.....	52
4.2.1	Clasificación por el conocimiento que contienen.....	52
4.2.2	Clasificación por motivación.....	52
5	Tecnología de Agentes.....	53
5.1	¿Qué son los Agentes?.....	53
5.2	Agentes Inteligentes.....	55
5.3	Sistemas Multi-Agente.....	56
6	Comparación entre Web Actual y Web Semántica.....	57
6.1	Orígenes de la Web Actual.....	57
6.2	Evolución de la Web Actual.....	59
6.3	Problemas y dificultades en la Web Actual.....	60
6.4	El nuevo paradigma: La Web Semántica.....	62
6.5	La transición hacia la Web Semántica.....	64
6.6	Conclusiones.....	65
7	Uso de la Web Semántica.....	66
7.1	Herramientas para la Web Semántica.....	68
7.1.1	Editores de Metadatos u Ontológicos.....	68
7.1.2	Herramientas de anotación externas.....	69
7.1.3	Herramientas de autor.....	70
7.1.4	Lenguajes de codificación de Ontologías.....	71
8	Análisis de las herramientas para la construcción de la Web Semántica.....	73
8.1	Metodología para análisis de herramientas.....	73
8.1.1	Esquema de tabla para el análisis las herramientas de anotación externas y herramientas de anotación de autor.....	73
8.1.2	Esquema de tabla para el análisis las herramientas de creación de ontologías.....	75
8.2	Análisis individualizado de las principales herramientas.....	77
8.2.1	Cohse.....	77
8.2.2	Amaya.....	80
8.2.3	Annozilla.....	83
8.2.4	Yawas.....	86
8.2.5	Annotation System.....	89
8.2.6	Trellis Web.....	92
8.2.7	MnM.....	96
8.2.8	OntoMat-Annotizer.....	100
8.2.9	SHOE Knowledge Annotator.....	103
8.2.10	SMORE.....	107
8.2.11	Melita.....	110
8.2.12	GATE.....	114
8.2.13	LinkFactory.....	117
8.2.14	OILEd.....	120
8.2.15	OntoEdit.....	123
8.2.16	Ontolingua Server.....	126
8.2.17	Protégé.....	130



8.2.18	WebODE.....	134
8.3	Comparación de herramientas .....	138
8.3.1	Comparación de herramientas de anotación externas .....	138
8.3.2	Comparación de herramientas de anotación de autor .....	140
8.3.3	Comparación de herramientas de creación de ontologías.....	142
9	Estudio de las Aplicaciones en uso de la Web Semántica .....	145
9.1	FOAF (Friend-of-a-Friend).....	145
9.1.1	¿Qué es FOAF? .....	145
9.1.2	Identificación de personas.....	146
9.1.3	Extendiendo FOAF.....	147
9.2	Sindicación de los contenidos Web .....	148
9.2.1	¿Qué son los Syndications Feeds? .....	148
9.2.2	El apoyo de la industria.....	149
9.2.3	Descubriendo los feeds.....	150
9.2.4	Feeds: cómo suscribirse y leerlos .....	151
9.2.5	Nuevos retos para la sindicación Web.....	152
9.3	Análisis de RSS y Atom .....	153
9.3.1	RSS.....	153
9.3.2	Atom.....	157
9.4	La Web Semántica y las Bases de Datos .....	159
9.4.1	¿Para qué sirven las SGBD?.....	159
9.4.2	SGBD existentes .....	160
9.4.3	Lenguajes de consulta para RDF en relación con los SGBD .....	163
9.4.4	Análisis de los lenguajes RDF concretos .....	164
9.4.5	Análisis de soluciones SGBD que implementan soporten RDF.....	167
9.5	Portales Web.....	177
9.6	Colecciones multimedia .....	178
9.7	Gestión de sitios Web corporativos.....	179
9.8	Agentes Inteligentes .....	180
9.9	Agentes y Servicios Web .....	182
9.10	Informática presente en la vida diaria .....	184
9.11	Buscadores semánticos.....	185
9.12	Firmas digitales y redes de confianza .....	186
9.13	Topic Maps.....	188
9.14	Otro tipo de aplicaciones .....	192
10	Análisis de las tecnologías de Web Semántica .....	195
10.1	Búsquedas en la Web Semántica.....	195
10.1.1	La cuestión de la búsqueda en la Web Semántica.....	195
10.1.2	Métrica definible para la Web Semántica .....	196
10.1.3	Resumen del estado del arte del análisis de la Web Semántica .....	197
10.1.4	Diversos Análisis Realizados sobre la Web Semántica .....	198
10.1.5	Estadísticas de la Web Semántica en www.redwebsemantica.es.....	204
10.2	Algunos usos actuales representativos de la Web Semántica .....	211
	Índice de Figuras .....	214
	Índice de Tablas .....	216
	Glosario .....	218

A	218
Administrator	218
Anotación	218
Anotación semántica	218
Apache	218
API	218
ASP	218
B	218
Benchmark	218
Bugs	218
C	218
CDA	218
CGI	219
D	219
DAML	219
DAML+OIL	219
DOM	219
DTD	219
E	219
ebXML	219
F	219
FLASH	219
FpXML	219
G	220
GPL	220
Grafo	220
H	220
Hipermedia	220
HTML	220
HTTP	220
I	220
IEEE	220
IEEE LOM	220
IsaViz	220
J	221
J2EE	221
Java	221
JavaBeans	221
JavaScript	221
Jena	221
Joseki	221
JSP	221
K	221
Kaon	221
L	221
M	221



Metadato .....	221
MySQL .....	222
N .....	222
N3 .....	222
Netscape .....	222
NewsML .....	222
NLM Medline.....	222
O .....	222
OCML.....	222
Ontologia.....	222
Oracle .....	222
OWL.....	222
P .....	222
Parser .....	222
PHP .....	222
Plugin.....	223
PostgreSQL.....	223
PRISM .....	223
R .....	223
RDF .....	223
RDF Schema .....	223
RDFS .....	223
RDQL.....	223
Recurso.....	223
Reificacion .....	223
RIXML .....	223
Rollback.....	223
RQL .....	224
S.....	224
SAIL.....	224
SCIPHOX .....	224
SCORM .....	224
SeRQL .....	224
Servlet .....	224
Sesame .....	224
SGBD.....	224
Shell.....	224
SPARQL.....	224
SQL .....	224
T.....	225
Tomcat .....	225
TopicMaps.....	225
Triple RDF.....	225
U .....	225
URI.....	225
URL .....	225

V .....	225
Versa .....	225
W .....	225
W3C.....	225
WebODE.....	225
WWW .....	225
X .....	226
XBRL .....	226
XHTML.....	226
XML .....	226
XMLNews .....	226
Bibliografía .....	227
Páginas Web de referencia .....	231





© RICARDO GARCIA RUIZ

Reservats tots els drets. Està prohibida la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor.



## Descripción del TFC

---

*"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 cooperation."* [Tim Berners-Lee, James Hendler, Ora Lassila - The Semantic Web, Scientific American, Mayo 2001]

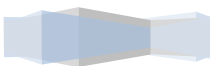
Este es el tipo de cooperación al que aspira conducirnos el concepto de Web Semántica.

Pero ¿qué es la Web Semántica realmente? Básicamente, es un área pujante en la confluencia de la Inteligencia Artificial y las tecnologías Web que propone introducir descripciones explícitas sobre el significado de los recursos, para permitir que las propias máquinas tengan un nivel de comprensión de la Web suficiente como para hacerse cargo de una parte, la más costosa, rutinaria, o físicamente inabarcable, del trabajo que actualmente realizan manualmente los usuarios que navegan e interactúan con la Web.

Mucho se lleva escrito sobre la Web Semántica, sobre la teoría, sus posibles aplicaciones, como instrumentarla, como hacerla eficaz y comercialmente atractiva... Sin embargo, en este trabajo se pretende ser algo más específico. Se intentará abordar un análisis detallado de las tecnologías que se valen de los conceptos e instrumentos de la Web Semántica, verificando su utilidad real, cuantificando en la medida de lo posible su impacto e intentando extrapolar datos sobre el futuro de la misma.

Así, se realizará un análisis sobre lo que es la Web Semántica, cómo se define, qué lenguajes son más apropiados para su desarrollo, las aplicaciones comerciales que se pueden desarrollar con la tecnología de Web Semántica, las aplicaciones que actualmente están en funcionamiento con esta tecnología, qué tipo de despliegue de la Web Semántica existe actualmente, utilización real de la tecnología de Web Semántica,...





# 1 Introducción

---

## 1.1 *Objetivos específicos del TFC*

---

El proyecto de desarrollo del TFC pretende abarcar en la medida de lo posible los siguientes aspectos:

- Historia y orígenes de la Web Semántica.
- Características de la Web Semántica.
- Comparación de la Web Semántica con la Web en uso actualmente.
- Características estructurales de la Web Semántica.
- Búsqueda y análisis de las aplicaciones empresariales basadas en Web Semántica (soluciones propietarias y de software libre).
- Análisis comparativo de las herramientas seleccionadas en contraposición a las mismas herramientas de la Web tradicional.
- Análisis de Impacto en el funcionamiento de la Red, a nivel comercial, de implantación o de uso, que tienen realmente las herramientas de la Web Semántica frente al uso de las tradicionales.



## **1.2 Características específicas del TFC**

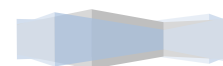
---

Este TFC es fundamentalmente de carácter investigador. No pretende en modo alguno desarrollar ninguna aplicación o función relacionada con la Web Semántica, XML o semejante.

No obstante lo anterior, si pretende que pueda convertirse en un referente general para otros proyectos de investigación posteriores. De esta forma sus objetivos son:

1. Sistematizar la búsqueda de datos referentes a la Web Semántica, tanto en libros o investigaciones publicadas como en la propia red.
2. Clasificar los datos encontrados de forma clara y coherente para que los mismos estén a disposición de los que deseen consultar la memoria final del TFC.
3. Sistematizar los datos relevantes encontrados en la documentación para su análisis posterior.
4. Realizar un análisis preciso de la Web Semántica de forma que queden claros los elementos básicos constitutivos del paradigma Web Semántica.
5. Analizar los elementos diferenciadores de la Web Semántica en relación a la Web Tradicional.
6. Analizar las soluciones implantadas en la Web y que tengan cierta relevancia por el uso que se haga de ellas o por sus características de desarrollo innovador.
7. Reflexionar sobre el tipo de soluciones Web Semántica, es decir, si son soluciones Open Source o propietarias.
8. Verificar cuál de las opciones (open source o propietarias) obtiene mejores resultados, tanto a nivel de preferencia por los usuarios como a nivel de resultados económicos.
9. Análisis del tráfico real que mantienen las soluciones Web Semántica escogidas previamente y verificar el impacto real que tienen en las preferencias de los desarrolladores.

Evidentemente el abanico de interés de este TFC es muy amplio, por lo que en el desarrollo del mismo se pretende priorizar en algunos aspectos sustanciales, como es el análisis de impacto real de las aplicaciones comerciales en el desarrollo de soluciones comerciales viables en la Red.



### ***1.3 Características secundarias del estudio comparativo a desarrollar***

En este caso, y dado el carácter fundamentalmente investigador y no desarrollador, el objetivo del TFC no está en el desarrollo de aplicaciones sino en el análisis de las soluciones ya implantadas.

Dentro de estos condicionantes a la investigación, y tomando en consideración que no son aspectos relevantes de primer orden para el presente TFC, se considera también dar respuesta a las siguientes cuestiones:

1. Nivel de coherencia con las reglas básicas de Web Semántica en los desarrollos implantados: en este caso se pretende verificar si las soluciones implantadas son puras de tipo Web Semántica o bien mantienen una cierta mezcla con las actuales prácticas de desarrollo Web.
2. Análisis de las ontologías y de los lenguajes de desarrollo de las ontologías.
3. Verificar la implantación de los sistemas basados en Web Semántica en los diversos sustratos de la generación de aplicaciones: SGBD, frameworks, IDE, ...
4. Limitaciones del lenguaje XML en el desarrollo de la Web Semántica: verificar el uso de la sintaxis XML en el desarrollo de soluciones Web Semántica.
5. También verificar y analizar los lenguajes de uso más comunes para la definición de las aplicaciones:
  - i. están basados en XML
  - ii. usan extensiones
  - iii. son estándares,
  - iv. si son soluciones portables,...
6. Análisis de los modelos transicionales:
  - i. Soluciones mixtas que usan tecnología no al 100% Web Semántica.
  - ii. Soluciones compartidas, que usan SGBD con estructura XML o derivada basada en ontología paradigmática de la Web Semántica, y usan modelos de aplicación no totalmente Web Semántica.
  - iii. Sistemas conversores: aplicaciones que acceden a datos no estructurados pero que son transicionales hacia un modelo Web Semántica (framework).

En cualquier caso, y dado el amplio espectro de trabajo para este TFC, se toma en consideración que estos puntos pueden ser tratados, si es posible, dentro del marco de trabajo de desarrollo de los puntos principales.



## **1.4 Cronología de desarrollo del TFC**

---

1. Tareas correspondientes a la fase de toma de datos y análisis de las características de la Web Semántica:
  - Estudio y definición de Web Semántica.
  - Análisis de los lenguajes de consulta en la Web Semántica
  - Sistematización de lenguajes orientados a Web Semántica y su impacto en las aplicaciones basadas en esta tecnología.
  - El concepto de Ontología en Web Semántica
2. Análisis comparativo de la Web Actual y la Web Semántica:
  - Búsqueda de definiciones generales sobre la Web Actual.
  - Buscar correspondencias entre la Web Actual y la Web Semántica.
  - Analizar los aspectos diferenciadores de la Web Semántica sobre la Web Actual
  - Definir los aspectos por los que se puede considerar superior la Web Semántica.
  - Definir los aspectos en los que es inferior la Web Semántica.
  - Realizar tabla comparativa entre ambas estructuras Web.
3. Definición estructural sobre qué es la Web Semántica.
  - Buscar y sistematizar una definición coherente y consistente sobre que características básicas definen la terminología Web Semántica.
  - Elaborar una tabla de características definitorias claras sobre Web Semántica.
4. Búsqueda de las aplicaciones o herramientas que actualmente hacen uso de las técnicas de Web Semántica.
  - Búsqueda de información en la Red.
  - Análisis de la información.
  - Estructuración de la información en torno a:
    - open source
    - propietarias
    - grado de implantación de las técnicas Web Semántica
    - sistemas mixtos transicionales
    - cualquier otro que según el análisis sea relevante
5. Análisis de las herramientas.
  - Análisis estadístico de uso de las mismas.
  - Análisis de grado de satisfacción de uso de dichas herramientas.
  - Análisis comparativo de las herramientas Web Semántica frente a similares de la Web Tradicional.
6. Análisis de impacto que estas tecnologías están teniendo en el desarrollo y tendencias actuales en la Web.
7. Confección de la documentación final del proyecto
8. Elaboración de la memoria del TFC
9. Elaboración de la presentación del TFC



## 1.5 Temporización

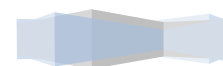
Id	Nombre	Días	Inicio	Fin	ID previo
1	Toma datos y análisis WS	16	16/03/2008	01/04/2008	
2	Estudio y definición de Web Semántica	4	16/03/2008	20/03/2008	
3	Análisis de los lenguajes de consulta en la Web Semántica	6	20/03/2008	26/03/2008	2
4	Sistematización e impacto lenguajes orientados a Web Semántica	6	26/03/2008	01/04/2008	3
5	El concepto de Ontología en Web Semántica	9	20/03/2008	29/03/2008	2
6	Análisis comparativo de la Web Actual y la Web Semántica	26	01/04/2008	27/04/2008	1
7	Búsqueda de definiciones generales sobre la Web Actual	3	01/04/2008	04/04/2008	1
8	Buscar correspondencias entre la Web Actual y la Web Semántica	10	04/04/2008	14/04/2008	7
9	Analizar los aspectos diferenciadores de la Web Semántica sobre la Web Actual	5	14/04/2008	19/04/2008	8
10	Definir los aspectos por los que se puede considerar superior la Web Semántica	3	19/04/2008	22/04/2008	9
11	Definir los aspectos en los que es inferior la Web Semántica	3	19/04/2008	22/04/2008	9
12	Realizar tabla comparativa entre ambas estructuras Web	5	22/04/2008	27/04/2008	10;11
13	Definición estructural sobre qué es la Web Semántica	20	04/04/2008	24/04/2008	7
14	Buscar y sistematizar una definición coherente y consistente sobre WS	10	04/04/2008	14/04/2008	7
15	Elaborar una tabla de características definitorias claras sobre Web Semántica	10	14/04/2008	24/04/2008	14
16	Elaboración PEC2	18	01/04/2008	19/04/2008	1
17	Entrega PEC2	1	20/04/2008	21/04/2008	16
18	Búsqueda de las aplicaciones o herramientas que actualmente hacen uso de las técnicas de Web Semántica	25	27/04/2008	22/05/2008	6;13
19	Búsqueda de información en la Red	5	27/04/2008	02/05/2008	
20	Análisis de la información	10	02/05/2008	12/05/2008	19
21	Estructuración de la información resultante según parámetros definidos	10	12/05/2008	22/05/2008	20
22	Análisis de las herramientas	15	22/05/2008	06/06/2008	18
23	Análisis estadístico de uso de las mismas	5	22/05/2008	27/05/2008	





24	Análisis de grado de satisfacción de uso de dichas Herramientas	5	27/05/2008	01/06/2008	23
25	Análisis comparativo de las herramientas Web Semántica frente a similares de la Web Tradicional	5	01/06/2008	06/06/2008	24
26	Elaboración PEC3	18	29/04/2008	17/05/2008	17
27	Entrega PEC3	1	18/05/2008	19/05/2008	26
28	Análisis de impacto que estas tecnologías están teniendo en el desarrollo y tendencias actuales en la Web	16	22/05/2008	07/06/2008	22
29	Elaboración de la memoria del TFC	70	29/03/2008	07/06/2008	5
30	Elaboración de la presentación del TFC	6	02/06/2008	08/06/2008	
31	Entrega Final TFC	1	08/06/2008	09/06/2008	
32	Entrega Final		08/06/2008	08/06/2008	29;30

**Tabla 1: Cronología de desarrollo del TFC**



**Contenido de las entregas parciales:**

**PEC2:**

Entrega de la memoria con los resultados de los puntos 1 al 9. y el 14 (como mínimo).

**PEC3:**

Entrega de la memoria con los resultados de los puntos hasta el 21 (como mínimo).

**Entrega final:**

Entrega de la memoria.

Entrega de la presentación.

**Evaluación del material necesario**

Se prevé que la principal fuente de información para el desarrollo de la aplicación sea Internet.

**Riesgos y planes de contingencia**

Los posibles riesgos vendrán derivados de las siguientes incidencias:

- Dificultades técnicas.
- Dificultad de utilización de las herramientas de software en la etapa de análisis.
- Imprevistos de índole personal-familiar.
- Aumentos de la carga y horario laboral.

En previsión a los riesgos relacionados con la etapa de implementación, el plan de contingencia consiste en poder destinar un mayor plazo de ejecución para dicha etapa en contraste con el resto de etapas (ver Temporización).

Para el resto de riesgos no previsibles, el plan de contingencia principal consiste en ampliar los plazos, disponiendo de los días de reserva. En ausencia de incidencias, dichos días serán destinados al repaso de la presentación y de la memoria.



## 1.6 Diagrama de Gantt

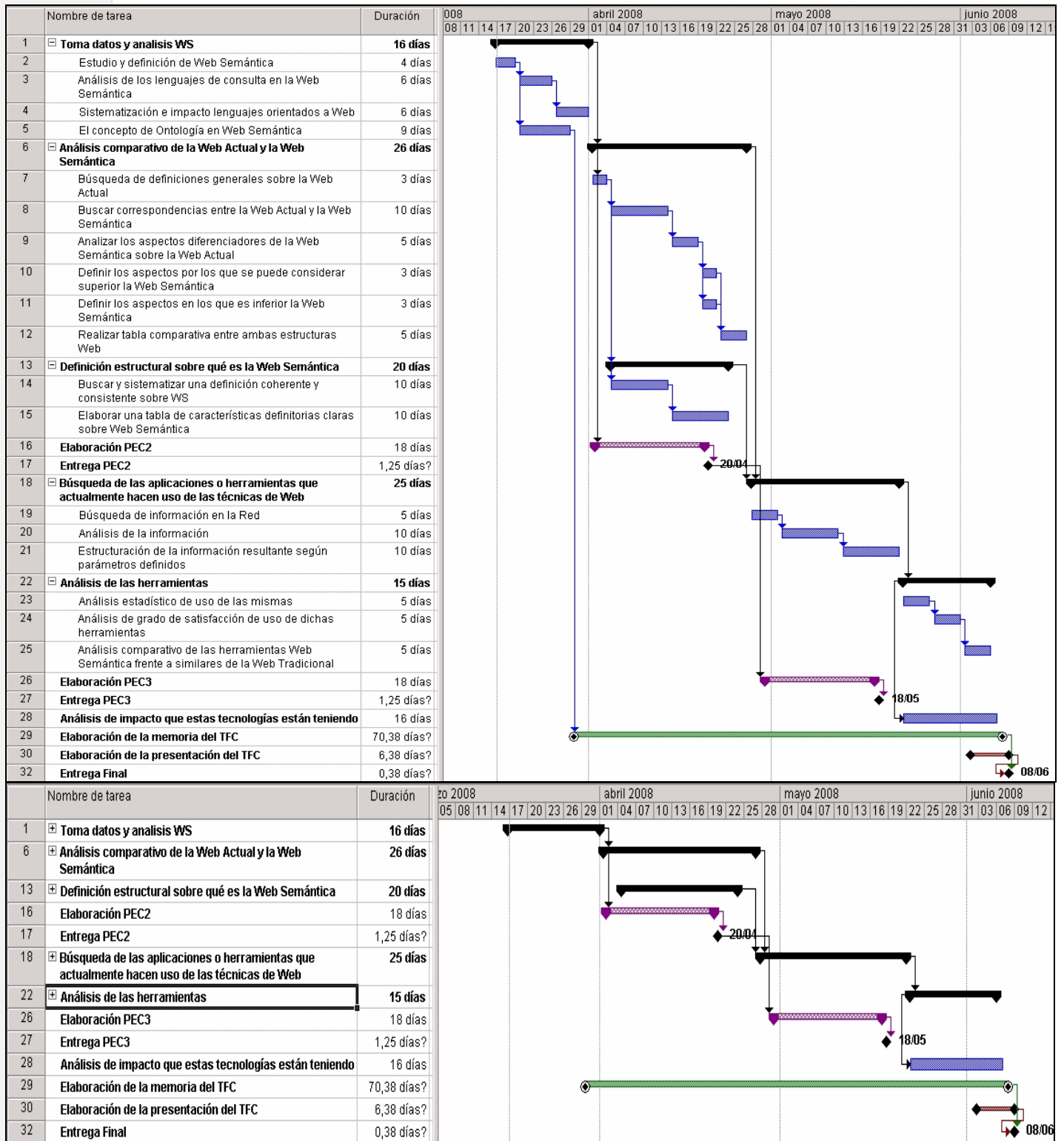


Fig. 1: Diagramas de Gantt con la programación temporal del desarrollo del TFC

## 2 Tecnologías relacionadas con la Web Semántica

---

### 2.1 XML

---

#### 2.1.1 Los lenguajes de marcado

---

Los lenguajes de marcado surgen a partir del nacimiento de SGML (Standard Generalized Markup Language) que es una norma de estandarización derivada de GML (Generalized Markup Language), la cual define que el etiquetado debe describir la estructura del documento electrónico y otros atributos de forma que puedan ser interpretados y procesados por una máquina. Así mediante marcas de delimitan y describen los elementos que componen el documento electrónico. Los lenguajes de marcado pueden utilizar de forma distintas las marcas, algunos las utilizan para estructurar los elementos del documento mientras otros las usan para definir el formato de presentación estos elementos.

#### 2.1.2 SGML (Standard Generalized Markup Language)

---

En 1969 tres empleados de IBM Research: Charles F. Goldfarb, Ed Mosher y Ray Lorie, inventaron el primer lenguaje moderno de marcado GML (Generalized Markup Language). Este describía cómo acotar mediante marcas la estructura arbitraria de un conjunto de datos, y llegó a convertirse en un metalenguaje, Un lenguaje que puede ser usado para describir otros lenguajes, sus gramáticas y sus vocabularios. GML se convirtió en SGML, y en 1986 fue adoptado por la ISO como un estándar internacional de almacenamiento e intercambio de datos.

SGML proporciona un modo consistente y preciso de aplicar etiquetas para describir las partes que componen un documento, permitiendo además el intercambio de documentos entre diferentes plataformas gracias a que está codificado únicamente con caracteres ASCII. Las marcas determinan la estructura no la presentación. Los elementos que pueden aparecer en un tipo de documento, sus características y el orden en el que deben escribirse se definen en un documento aparte llamado DTD (Document type definition), de esta forma los documentos se consideran instancias de un tipo de documento específico que define su estructura válida. El documento debe incluir una referencia a la DTD a partir de la cual se ha definido y cuyas restricciones debe cumplir.

SGML también tiene capacidad de hipertexto donde los enlaces unen un elemento origen con un elemento destino. Para poder imprimir o visualizar por pantalla un documento SGML se le debe aplicar un formato mediante una hoja de estilos donde se indica como formatear cada elemento del documento. Estas características hicieron que SGML se convirtiera en norma de facto para el intercambio de documentos extensos, complejos y de tipos muy diferentes. Sin embargo tiene un gran inconveniente: su complejidad.



### 2.1.3 HTML (HyperText Markup Language)

---

Cuando Tim Berners-Lee desarrolló el World Wide Web a finales de los 80 necesitaba un lenguaje para expresar los contenidos que iban a circular por la red y presentarse a través del navegador. Entonces utilizó el metalenguaje SGML como base para definir el lenguaje HTML original. Así se concibe HTML como una aplicación SGML para codificar documentos y distribuirlos en el World Wide Web. Su sencillez le proporciona éxito inmediato (revolución en Internet) pese a ello presenta algunas limitaciones de las que destacan las siguientes:

- Limitación para describir documentos complejos o datos.
- Define más la presentación que el contenido.
- No es fácilmente procesable por máquinas.
- Problemas de internacionalización.
- Falta estructuración y validación.
- No es reutilizable ni extensible.
- Interpretación ambigua según software utilizado.
- Sólo usable para Web.
- No dispone de un mecanismo de enlaces robusto.
- Tecnologías (Java, lenguajes de script, etc.) muy potentes cuya capacidad está siendo infrautilizada por las limitaciones del formato HTML.

Algunas de estas limitaciones se han querido subsanar con la incrustación de scripts, javascripts, Active X, HTML dinámico, hojas de estilo en cascada (CSS). Todo esto es insuficiente para crear una arquitectura abierta de tipo cliente/servidor, con lo que el W3C (World Wide Web Consortium), organismo que vela por el desarrollo de la World Wide Web, se ha replanteó crear un nuevo estándar llamado XML (eXtensible Markup Language) que ofreciese las ventajas que aportaba SGML pero evitando sus complejidades.

### 2.1.4 XML (eXtensible Markup Language)

---

La versión 1.0 del lenguaje XML es una recomendación del W3C desde Febrero de 1998, pero se ha trabajado en ella desde un par de años antes. Está basado en el anterior estándar SGML, de él se derivó XML como subconjunto simplificado.

HTML es simplemente un lenguaje, una aplicación SGML, mientras que XML como SGML es un metalenguaje, esto es, un lenguaje para definir lenguajes.

XML no se presenta como sustituto de HTML, XML no ha nacido sólo para su aplicación en Internet, sino que se propone como lenguaje para intercambio de información estructurada entre diferentes plataformas.

Mientras que HTML especifica lo que cada etiqueta y atributo significan (y frecuentemente la apariencia que presentará en un navegador el texto que hay entre ellos) XML usa las etiquetas sólo para delimitar piezas de datos, y deja la interpretación de los datos, completamente, a la aplicación que los lee, pero las reglas para los archivos XML son más estrictas que para los archivos HTML.

Algunos de los objetivos planteados por el Grupo de Trabajo XML y el W3C son:

- XML debe ser directamente utilizable sobre Internet.
- XML debe soportar una amplia variedad de aplicaciones.

- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- El número de características opcionales en XML debe ser absolutamente mínimo, idealmente cero.
- Los documentos XML deben ser legibles por los usuarios de este lenguaje y razonablemente claros.
- El diseño de XML debe ser formal, conciso y preparado rápidamente.
- Los documentos XML deben ser fácilmente creables.
- La brevedad en las marcas XML es de mínima importancia.

### 2.1.5 Breve comparativa de HTML y XML

---

El siguiente cuadro muestra algunas de las características que diferencian XML y HTML.

Característica	1ª Generación	2ª Generación
<b>Lenguaje principal</b>	HTML	XML
<b>Forma y estructura</b>	documentos no estructurados	documentos estructurados
<b>Semántica</b>	semántica implícita	etiquetado explícito (metadatos, Web semántica)
<b>Relación entre contenido y forma</b>	HTML = fusión de forma y contenido	estructura en capas de forma y contenido: XML + transformación (p.e., XSL) a HTML, WML, PDF, u otros formatos
<b>Editabilidad</b>	documentos estáticos	documentos dinámicos
<b>Descomponibilidad y recomponibilidad</b>	sitios Web monolíticos, independientes	bricolaje (agregación), sindicación, reasignación de contenido
<b>Interactividad</b>	medio de difusión unidireccional	Web editable, bidireccional
<b>Audiencias</b>	para consumo humano	para humanos y ordenadores (p.e., servicios Web)
<b>Control de producción</b>	centralizado	descentralizado (peer-to-peer P2P)

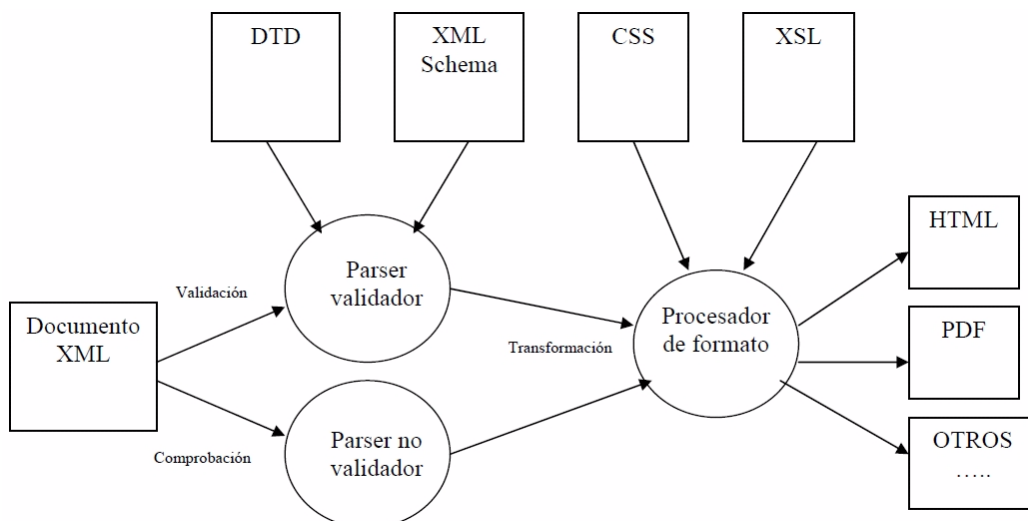
Tabla 2: Diferencias entre HTML y XML



## 2.1.6 Características básicas de XML

XML fue desarrollado por un grupo de trabajo bajo los auspicios del consorcio World Wide Web (W3C) a partir de 1996. Este fue constituido en 1994 con el objetivo de desarrollar protocolos comunes para la evolución de Internet. Al comenzar el proyecto, los objetivos planteados por el grupo de desarrollo del XML fueron diez puntos:

1. XML debe ser directamente utilizable sobre Internet.
2. XML debe soportar una amplia variedad de aplicaciones.
3. XML debe ser compatible con SGML.
4. Debe ser fácil la escritura de programas que procesen documentos XML.
5. El número de características opcionales en XML debe ser absolutamente mínimo, idealmente cero.
6. Los documentos XML deben ser legibles por los usuarios de este lenguaje y razonablemente claros.
7. El diseño de XML debe ser formal, conciso y preparado rápidamente.
8. XML debería ser simple pero perfectamente formalizado.
9. Los documentos XML deben ser fáciles de crear.
10. La brevedad en las marcas XML es de mínima importancia.



**Fig. 2: Diagrama de análisis, proceso y formateo de un documento XML**

Un análisis detallado de las características de XML nos lleva a condensarlas en la siguiente tabla:

Es un dialecto simplificado de SGML pero con las propiedades más importantes y deseables de éste
XML es un Metalenguaje: es un lenguaje de marcas permite la creación de otros lenguajes de marcas
XML ofrece una representación estructural de los datos que se puede implementar ampliamente y es fácil de distribuir
XML garantiza que los datos estructurados sean uniformes e independientes de aplicaciones o fabricantes. Interoperabilidad
Intercambio de datos e información estructurada a través de Internet y el WWW (Prácticamente se ha convertido en el estándar de facto para el intercambio de información en Internet)
Facilita la Integración de datos procedentes de fuentes heterogéneas
Fácilmente procesable por humanos y máquinas

Separa información o contenido de presentación y formato
Utilizable con cualquier lenguaje o alfabeto
Codifica datos de acuerdo a un esquema semántico
Basado en marcas sintácticas que cumplen reglas léxicas y sintácticas. Se etiquetan los componentes del documento de acuerdo con su semántica
Se le puede asociar una gramática para definir un modelo de estructura de documento (DTD - Esquema)
Extensible, etiquetas no predefinidas
Pretende ser razonablemente simple

**Tabla 3: Características fundamentales del lenguaje XML**

### **2.1.7 Limitaciones de XML con los metadatos**

---

Los metadatos se pueden definir como datos sobre datos, es decir, descripciones de características y propiedades sobre datos. Estas descripciones son de gran ayuda en la catalogación y recuperación de datos. XML presenta algunas deficiencias para la creación de metadatos. Básicamente, la falta de flexibilidad y escalabilidad, ya que el orden en el cual los elementos aparecen en un documento XML es significativo y muchas veces necesario.

Esto es altamente antinatural en el mundo de los metadatos, no es relevante si en un listado de propiedades de un libro se pone el ISBN antes que la fecha de publicación, lo importante es poder acceder a los datos. Además es difícil y costoso el mantenimiento del orden sobretodo cuanto mayor son los elementos a tratar. La limitación está en la estructura jerárquica y en forma de árbol del modelo de datos XML. Los árboles XML son complicados de combinar entre sí. Además los accesos en estos árboles es problemático debido a los múltiples caminos que puede haber para llegar hasta un dato determinado. Este problema se acentúa cuanto mayor es el árbol, y por ello no ofrece buena escalabilidad en sus accesos.

Por otro lado las gramáticas XML permiten validar la estructura de los documentos XML pero no permiten interpretar de forma general los datos que contienen. Para ello se debe implementar una aplicación que proporcione los significados de los datos. Ante estas limitaciones aparece RDF con un modelo de datos más flexible basados en triples (sujeto + predicado + objeto) y cuyos elementos están identificados con URIs lo que permite el acceso a su semántica y a la interpretación de los datos.





## **2.2 RDF (Resource Description Framework)**

---

### **2.2.1 Comprensión de qué RDF**

---

Actualmente, el lenguaje recomendado por el W3C para representar semántica en la nueva generación de la Web es en este momento el RDF (Resource Description Framework). Mientras que XML permite definir datos de una forma estructurada, RDF permite definir la semántica de esos datos.

RDF fue creado en agosto de 1997 bajo los auspicios del World Wide Web Consortium (W3C) con el fin de crear un formato que permitiera alcanzar la compatibilidad entre los diversos sistemas de metadatos, suministrando para ello una arquitectura genérica de metainformación.

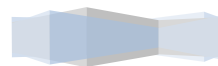
El objetivo general de RDF es definir un mecanismo para describir recursos que no cree ninguna asunción sobre un dominio de aplicación particular, ni defina (a priori) la semántica de algún dominio de aplicación.

La definición del mecanismo debe ser neutral con respecto al dominio, sin embargo el mecanismo debe ser adecuado para describir información sobre cualquier dominio. La capacidad que tiene RDF para procesar metadatos facilita la interoperabilidad entre diversas aplicaciones, proporcionando un mecanismo perfecto de intercambio de información a través del Web.

Existen varios conceptos que pueden definir el modelo RDF, entre los cuales caben mencionar:

- Sistema que permite la interoperabilidad entre aplicaciones mediante el intercambio de información legible por ordenador a través del Web.
- Mecanismo que facilita la automatización de procesos susceptibles de ser realizados con recursos Web.
- Infraestructura que permite la codificación, intercambio y reutilización de metadatos estructurados (Miller, Eric, 1998). Es capaz, además, de fusionar diferentes sistemas de metadatos utilizados para la descripción de recursos Web.
- Es una forma de expresar relaciones entre objetos.

En general RDF se enfoca en establecer un mecanismo que permita describir recursos, entendidos estos como objetos, que tengan como principios la multiplataforma (es decir, independencia de software y/o sistema operativo) y la interoperatividad de metadatos (que posibilite fusionar diferentes descripciones de recursos realizadas con distintos conjuntos de metadatos). RDF posee semánticas que generan una base para razonar sobre el significado de una expresión RDF. Además posee un vocabulario extensible, basado en URIs.



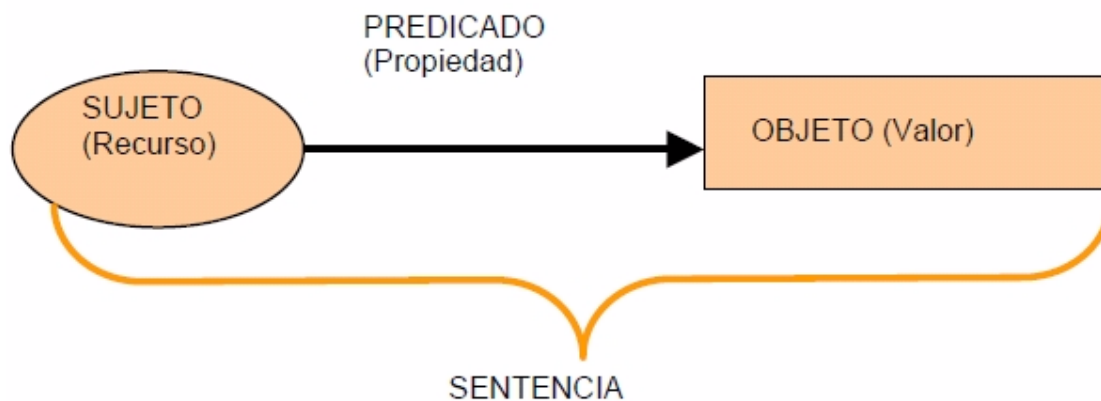
## 2.2.2 El Modelo Básico RDF

---

El modelo RDF se compone de tres conceptos principales:

- **Recursos:** Todas las cosas descritas por expresiones RDF se denominan recursos. Los recursos, representan cualquier entidad (lugares, personas, objetos) del mundo real y están identificados por un URI (Universal Resource Identifier). Un recurso puede ser una página Web, una parte de una página Web; p. ej. un elemento HTML o XML específico dentro del documento fuente, o un objeto que no sea directamente accesible vía Web, p. ej. un libro impreso.
- **Propiedades:** Una propiedad es un aspecto específico, característica, atributo, o relación utilizado para describir un recurso. Cada propiedad tiene un significado específico, define sus valores permitidos, los tipos de recursos que puede describir, y sus relaciones con otras propiedades. Las características de las propiedades se definen con RDFS (RDF Schema ) que explicaré más adelante.
- **Sentencias (declaraciones, enunciados):** Un recurso específico junto con una propiedad denominada, más el valor de dicha propiedad para ese recurso es una sentencia RDF. Estas tres partes individuales de una sentencia se denominan, respectivamente, sujeto, predicado y objeto. Una propiedad es a su vez un recurso. El objeto es el valor asignado a dicha propiedad, y puede ser una cadena simple de caracteres (string), u otro recurso, es decir, un recurso especificado por un URI.

Las sentencias constituyen la construcción básica que establece el modelo de datos RDF. Es decir, el significado de los datos se expresa mediante un conjunto de sentencias RDF que formalmente son representadas por tri-uplas (sujeto, predicado, objeto). Otra forma de representar las declaraciones (sentencias) RDF es a través de un grafo rotulado como se muestra en la **Fig. 3**.



**Fig. 3: Grafo de declaración RDF**



### 2.2.3 XML como Lenguaje de Especificación de la Sintaxis RDF

Uno de los principales aspectos que han contribuido al éxito de la tecnología RDF en el contexto de la Web, es la posibilidad de poder representar e intercambiar datos (sentencias) RDF usando XML. Además toda la sintaxis RDF está definida en XMLS en el documento <http://www.w3.org/1999/02/22-rdf-syntax-ns>.

Sin embargo, es importante remarcar que el modelo de datos RDF no es serializado en un árbol sintáctico XML, donde se debe respetar la jerarquía especificada por el esquema XML relacionado. El modelo de datos RDF (a diferencia del modelo de árbol XML) es un grafo dirigido, lo que nos permite una mayor potencia para representar sentencias (podemos aseverar cualquier cosa sobre cualquier objeto), y esto es totalmente independiente de su representación serializada XML.

```
<?xml version="1.0" >
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:m="http://gidis.edu.ar/Esquema#">
  <rdf:Description about="http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo">
    <m:catalogada>20/04/2004</m:catalogada>
    <m:cuantifica
  rdf:resource="http://gidis.edu.ar/DatosMetricas#DensidadDeErrores"/>
  </rdf:Description>
</rdf:RDF>
```

Fig. 4: Serialización en XML de sentencias RDF

La Fig. 4 muestra cómo ciertas sentencias pueden ser serializadas usando XML. La segunda línea del código indica el trozo que usa sintaxis RDF, e identifica con los prefijos “rdf” y “m” la ubicación de los documentos que contienen las definiciones de los elementos usados, es decir los espacios de nombres. Las demás líneas representan una descripción RDF del recurso identificado por el URI <http://gidis.edu.ar/DatosMetricas#ErroresPorMódulo>, que en este caso representa una métrica, la marca <m:catalogada> indica que la métrica que se está describiendo tiene una propiedad llamada “catalogada” cuyo valor es “20/04/2004” y cuya semántica está definida en el vocabulario asociado al prefijo m:. Además, la marca <m:cuantifica> indica que la métrica también tiene una propiedad llamada “cuantifica” cuyo valor es el objeto (que en este caso es un recurso, no un literal) identificado por el URI “<http://gidis.edu.ar/DatosMetricas#DensidadDeErrores>”.

Cuando escribimos una frase en lenguaje natural utilizamos palabras que, en lo posible, encierran la intención de transmitir un significado inequívoco. En el caso de descripciones RDF se debe establecer el significado preciso de cada propiedad para que el procesamiento automático de las sentencias dé el resultado esperado. Es muy importante que tanto el escritor como el lector de una sentencia (declaración) entiendan el mismo significado para los términos utilizados, tales como catalogada, cuantifica, etc.

En RDF el significado se expresa a través de un esquema. Podemos pensar en un esquema como en una especie de diccionario. Un esquema define los términos que se utilizarán en una declaración RDF y le otorgará significados específicos. Para evitar confusiones entre definiciones independientes (y posiblemente conflictivas) del mismo término, RDF utiliza la facilidad de los namespace de XML. Los namespaces ("espacios de nombre") son simplemente una forma de asociar el uso específico de una palabra en el contexto del diccionario (esquema) en que se puede encontrar la definición. En RDF, cada predicado utilizado en una sentencia debe ser identificado con un sólo namespace, o esquema.

Sin embargo una descripción de recurso puede contener sentencias con predicados de varios esquemas.

## 2.2.4 Contenedores en RDF

Muchas veces es necesario expresar que un recurso está relacionado con una colección de objetos, por ejemplo, para expresar que una métrica fue desarrollada por más de una persona, o para enumerar las distintas métricas que cuantifican un atributo de una entidad. Los contenedores RDF se usan para mantener tales listas de recursos o literales. RDF define tres tipos de objetos contenedores:

- **Bag:** Una lista desordenada de recursos o literales. Los Bags (bolsa en español), se utilizan para indicar que una propiedad tiene múltiples valores y que no es significativo el orden en que se den tales valores.
- **Sequence:** Una lista ordenada de recursos o literales. Sequence se usa para manifestar que una propiedad tiene múltiples valores y que el orden de los valores es significativo.
- **Alternative:** Una lista de recursos o literales que representan alternativas para un valor (individual) de una propiedad. Una aplicación que utiliza una propiedad cuyo valor es una colección alternativa, sabe que se puede elegir como correcto uno cualquiera de los ítems en la lista.

Las definiciones de Bag y Sequence permiten explícitamente duplicar valores. RDF no define un concepto puntual de Set (conjunto), que podría ser un Bag sin duplicados, porque la esencia de RDF no impone un mecanismo de validación en el caso de que se violen estas restricciones.

Para representar colecciones de recursos, RDF utiliza un recurso adicional que identifica la colección específica (una instancia de uno de los tipos objetos contenedores definidos arriba). La propiedad type, se utiliza para hacer esta declaración. La relación de pertenencia entre estos recursos contenedores y los recursos que pertenecen a la colección se establece a través de un conjunto de propiedades de filiación que se denominan simplemente "\_1", "\_2", "\_3", etc. Los recursos contenedores pueden tener otras propiedades añadidas a las propiedades de pertenencia, y la propiedad type. Cualquiera de tales sentencias adicionales describe al contenedor.

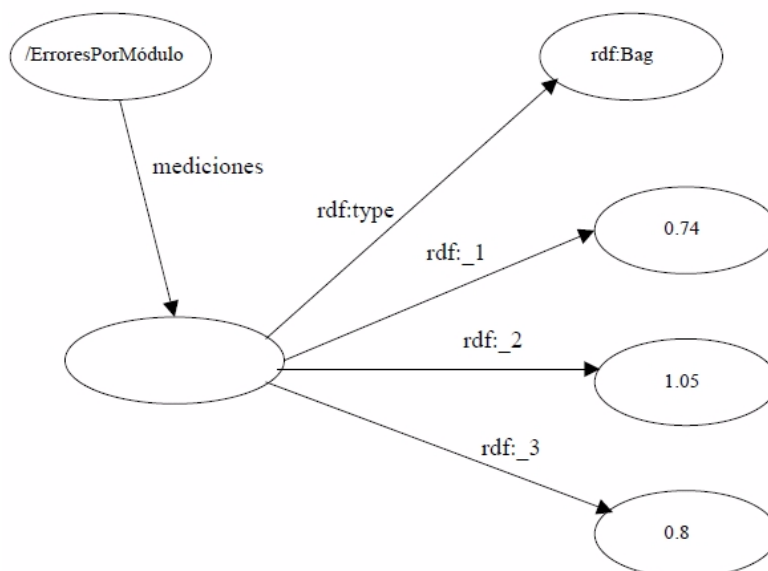


Fig. 5: Ejemplo de uso de un contenedor Bag

Por ejemplo en el modelo RDF de la Fig. 5, se representa la sentencia: "Los valores de mediciones tomadas para la métrica ErroresPorMódulo son 0.74, 1.05, y 0.8".



### 2.2.5 Reificación en RDF (sentencias sobre sentencias)

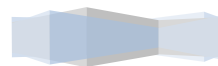
---

Además de permitir sentencias sobre los recursos Web, RDF puede usarse para crear sentencias sobre otras declaraciones RDF; nos referiremos a éstas como sentencias de alto nivel. Para realizar declaraciones sobre otras declaraciones, tenemos que construir un modelo de la sentencia original; este modelo es un nuevo recurso al que podemos anexar propiedades adicionales. Esta característica de RDF de poder considerar una sentencia como un recurso permite anidar declaraciones, y a este proceso se lo denomina formalmente "reificación" (transformación de algo abstracto en concreto) en el contexto de la representación del conocimiento.

Formalmente, una reificación en RDF significa expresar una sentencia como un recurso con cuatro propiedades:

- **subject:** La propiedad subject identifica el recurso que describe la sentencia modelada; es decir, el valor de la propiedad subject es el recurso sobre el cual se hace la sentencia original.
- **predicate:** La propiedad predicate identifica la propiedad original en la declaración modelada. El valor de la propiedad predicate representa la propiedad específica en la sentencia original.
- **object:** La propiedad object identifica el valor de la propiedad en la sentencia modelada. El valor de la propiedad object es el objeto en la sentencia original.
- **type:** El valor de la propiedad type describe el tipo del nuevo recurso. Todas las sentencias transformadas reificadas son instancias de `rdf:statement`; es decir tienen una propiedad type cuyo objeto es `rdf:statement`.

Un nuevo recurso con las cuatro propiedades anteriores representa la sentencia original y puede usarse como objeto de otras declaraciones.



## 2.3 RDF Shema

---

### 2.3.1 De RDF a RDF Schema

---

RDF no ofrece ningún mecanismo para especificar el vocabulario usado en las descripciones (sentencias) tal como el término “documentada” usado en el ejemplo anterior. RDF Schema (RDFS), es el lenguaje que completa el modelo RDF básico en la función de alcanzar interoperabilidad semántica en el contexto de la Web.

Esta propuesta del W3C provee un sistema de tipos básicos conjuntamente con un mecanismo para definir nuevos tipos, conformando distintos espacios de nombres, permitiendo de esta manera, que comunidades de usuarios de distintos dominios puedan compartir sus propios vocabularios. RDFS es una extensión de RDF que permite:

- Describir los conceptos usados en aplicaciones RDF
- Especificar restricciones de tipos para los sujetos y objetos de las triplas.

RDFS se puede ver como una forma de modelado orientado a objetos (OO) en la Web. Con los atributos predefinidos `rdfs:Class` y `rdfs:subClass` es posible definir una jerarquía de clases, y las instancias pueden referirse a ellas a través del atributo `rdf:type` de RDF.

El mecanismo para la definición de tipos en un esquema RDF es ligeramente diferente a la definición de tipos de las metodologías tradicionales de modelado OO. En ésta última, la principal preocupación es la identificación de las entidades que serán representadas como clases, subclases y sus correspondientes atributos; en RDFS, en cambio, las propiedades (atributos) son definidas en términos de las clases de recursos a las cuales ellas se aplican. Este enfoque centrado en las propiedades facilita la descripción semántica de los recursos existentes en la Web (enumerando sus propiedades y valores asociados), principal objetivo de la arquitectura RDF.

Así, el modelo de datos de RDF define un modelo simple para describir las relaciones entre recursos en términos de propiedades y valores designados. Las propiedades RDF pueden entenderse como atributos de un recurso y en este sentido corresponden con los tradicionales pares atributo-valor. Las propiedades RDF también representan relaciones entre recursos. De esta forma, el modelo de datos RDF puede parecer un diagrama entidad-relación. El modelo de datos RDF, sin embargo, no proporciona mecanismos para declarar estas propiedades, ni proporcionan ningún mecanismo para definir las relaciones entre estas propiedades y otros recursos. Este es el papel o la función que desempeña el Esquema RDF.

Resumiendo, RDFS proporciona información sobre la interpretación de una sentencia dada en un modelo de datos RDF. Mientras una DTD o un Esquema XML pueden utilizarse para validar la sintaxis de una expresión XML, un esquema sintáctico sólo no es suficiente para los objetivos de RDF. Los Esquemas RDF pueden también especificar restricciones que deben seguirse por estos modelos de datos. El trabajo futuro en torno al Esquema RDF y al Esquema XML podría facilitar la sencilla combinación de reglas sintácticas y semánticas para ambos.

La RDF Schema es una extensión de RDF que proporciona primitivas adicionales. Enriquece el modelo básico, proporcionando un vocabulario para RDF, que se asume tiene una cierta semántica. Permite a los diseñadores especificar una jerarquía explícita de clases de recursos y propiedades que describen estas clases, junto con las restricciones sobre las combinaciones permitidas de clases, propiedades y valores.



A continuación se describen las principales primitivas de modelado RDFS que están agrupadas en clases, propiedades y restricciones para facilitar su explicación. La Fig. 6 (extraída del reporte técnico del W3C ), muestra la relación existente entre estas primitivas y las primitivas del modelo RDF básico, sobre el cual está construido RDFS.

En dicha figura, los rectángulos con puntas redondeadas indican clases y las flechas señalan cuál es la clase que define al recurso. La relación subclase se indica con un rectángulo (la superclase) que contiene a otro rectángulo (la subclase). Por último, se muestra que todo objeto en RDF es un recurso (Resource).

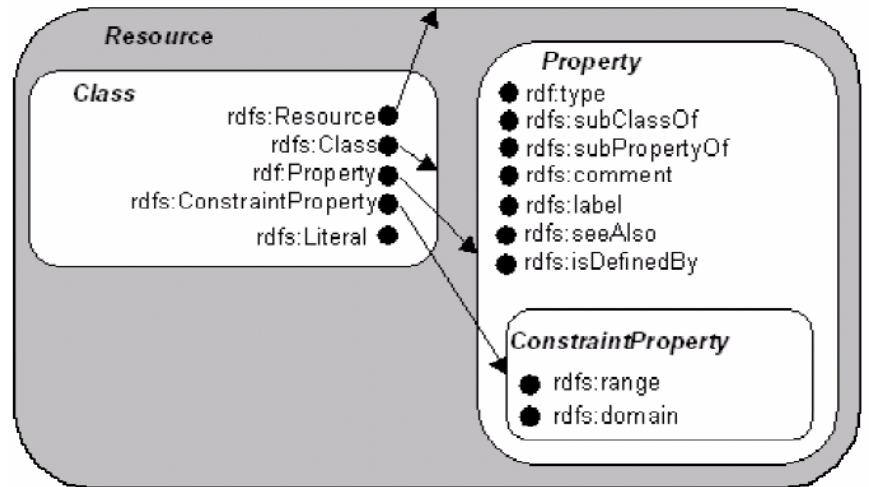


Fig. 6: Jerarquía de clases del modelo RDFS

## 2.3.2 Clases, propiedades y restricciones

---

### Clases

---

RDFS permite la definición de subclases que heredan las definiciones de una o más clases ascendentes, permitiendo la implementación de herencia múltiple. Esta característica incorpora gran flexibilidad al modelo RDF, porque las clases que se están definiendo pueden heredar de clases ya definidas en esquemas previos, especializando los metadatos, promoviendo de esta manera compartir y reusar estos esquemas. Para este propósito, las principales primitivas que ofrece RDFS son:

- **rdfs:Resource** - representa la clase genérica en el modelo RDF Schema. Todo objeto descrito por expresiones RDF es un recurso.
- **rdfs:Class** – es subclase de `rdfs:Resource` y representa el concepto genérico de tipo o categoría, similar a la noción de clase en orientación a objetos.
- **rdf:Property** – es subclase de `rdfs:Resource` y representa un aspecto del recurso que se está describiendo, similar a la noción de atributo en orientación a objetos.

### Propiedades

---

Las propiedades posibilitan expresar relaciones entre instancias y sus clases o clases y sus superclases. Relaciones entre propiedades también son permitidas, obteniéndose así una jerarquía de propiedades. Las principales propiedades disponibles en RDFS son:

- **rdf:type** – es subclase de `rdf:Property` y denota que un recurso es instancia de una clase, poseyendo todas sus características. Un recurso puede ser instancia de más de una clase.
- **rdfs:subClassOf** - es subclase de `rdf:Property` y denota la relación de subclase/superclase entre dos clases. Esta propiedad es la base para la especificación de la herencia múltiple, y tiene característica de transitividad.
- **rdfs:subPropertyOf** - es subclase de `rdf:Property` y denota la relación de especialización entre dos propiedades, posibilitando la definición de una jerarquía de propiedades.

### Restricciones

---

Este mecanismo permite asociar restricciones a las propiedades de un recurso. Las principales primitivas de restricciones son:

- **rdfs:domain** – es una instancia de la clase `rdfs:ConstraintProperty` y especifica a qué clase se aplica una propiedad.
- **rdfs:range** – es una instancia de la clase `rdfs:ConstraintProperty` y restringe los valores (u objetos) que una propiedad puede asumir.

Finalmente, es importante observar que RDFS solamente permite modelar relaciones binarias entre recursos, además, no permite especificar restricciones de cardinalidad sobre las relaciones ni sus propiedades (simétrica, transitiva), tampoco se pueden especificar condiciones necesarias y suficientes para definir pertenencia a una clase. En resumen, RDFS carece de poder expresivo para representar restricciones lógicas; de ahí que sea necesario en la Web semántica, una capa lógica por encima de RDFS.





## 2.4 OWL (Web Ontology Language)

Debido a las limitaciones de RDF/RDFS, principalmente la carencia del suficiente poder expresivo, se comenzaron a definir tanto en Estados Unidos como en Europa nuevos lenguajes ontológicos con mayor capacidad expresiva. Así surgieron SHOE, DAML-ONT, OIL, y DAML+OIL, que sería el precursor de OWL, la actual recomendación del W3C (World Wide Web Consortium). A continuación, se enumeran algunas de las posibilidades adicionales que proporciona el vocabulario de OWL sobre el de sus predecesores:

- Definición de clases mediante restricciones sobre propiedades, valores o cardinalidad.
- Definición de clases mediante operaciones booleanas sobre otras clases: intersección, unión y complemento.
- Relaciones entre clases (p.ej. inclusión, disyunción, equivalencia).
- Propiedades de las relaciones (p.ej. inversa, simétrica, transitiva).
- Cardinalidad (p.ej. “únicamente una”).
- Igualdad y desigualdad de clases.
- Igualdad y desigualdad de instancias.
- Clases enumeradas.

Estas características introducidas en el lenguaje, si bien mejoran el poder expresivo del lenguaje, también limitan la capacidad de los razonadores para inferir nuevo conocimiento a partir de unas sentencias dadas. En particular, no se pueden garantizar ni (1) la completitud computacional, esto es, que el razonador encuentre todas las conclusiones válidas, ni (2) la “decidibilidad” computacional, esto es, que se obtenga la respuesta para cualquier entrada en un periodo de tiempo finito. Esto llevó a la creación de tres variantes de OWL que presentan diferentes estados en la relación expresividad/decidibilidad. De menor capacidad expresividad/mayor eficiencia de razonamiento a mayor capacidad expresiva/menor eficiencia de razonamiento nos encontramos las siguientes:

- **OWL Lite:** permite una jerarquía de clasificación y restricciones simples (restricciones de cardinalidad, pero sólo valores de cardinalidad de 0 o 1). Entre las ventajas, se encuentran la facilidad de entender (por parte de los usuarios) y la facilidad de implementar (para constructores de herramientas). La mayor desventaja es, por supuesto, la restringida expresividad.
- **OWL DL (‘Description Logic’, lógica descriptiva):** es el lenguaje indicado para aquellos usuarios que requieren el máximo de expresividad, mientras conservan completamente la propiedad de ser computable (se garantiza que todas las conclusiones son computables) y resoluble (todos los cálculos terminarán en tiempo finito). Incluye todos los constructores del lenguaje OWL, pero solamente se pueden utilizar bajo ciertas restricciones.
- **OWL Full:** permite la máxima expresividad y ofrece la libertad sintáctica de RDF pero carece de garantías en cuanto a la propiedad de ser computable.

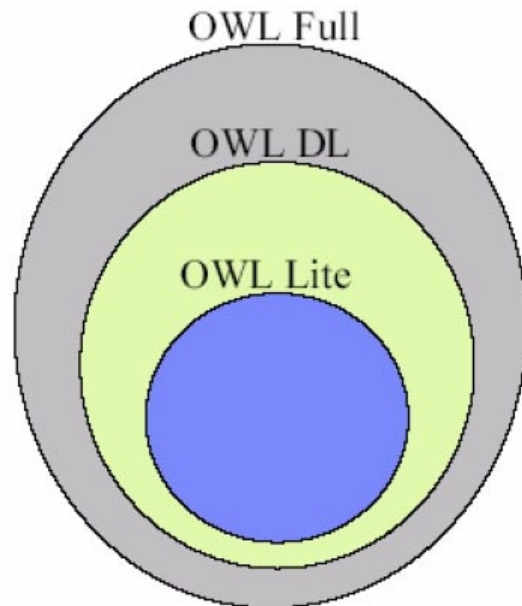


Fig. 7: Versiones del Lenguaje OWL

Cada uno de estos sub-lenguajes es una extensión de su predecesor más simple, en los que ambos pueden ser expresados legalmente (esto es, las fórmulas bien formadas en OWL Lite son válidas en OWL DL y las bien formadas en OWL DL válidas en OWL Full) y en los que pueden ser válidamente concluidos (esto es, toda fórmula inferida en OWL Lite puede ser válidamente concluida en OWL DL partiendo del mismo conocimiento, y, con la misma asunción, las inferidas en OWL DL válidamente concluidas en OWL Full) (Fig. 7).

El propósito de OWL es similar al de RDFS, esto es, servir de vocabulario XML para definir clases, sus propiedades y las relaciones entre clases. En comparación con RDFS, OWL permite expresar relaciones mucho más ricas y, por tanto, disponer de capacidades de inferencia mejoradas. La semántica formal y el soporte al razonamiento en OWL se garantizan a través del mapeo de OWL en formalismos lógicos utilizando la lógica de predicados (lógica de primer orden) y la lógica descriptiva.





## 3 La Web Semántica

---

### 3.1 Introducción a la Web Semántica

---

La Web Semántica, una visión de los creadores de la Word Wide Web (WWW), tiene por objetivo la creación una nueva red, similar a Internet, pero destinada para aplicaciones, en lugar de para usuarios humanos. Esta nueva Web es una red que ofrece datos y contenido semánticamente anotado para permitir que los programas Informáticos puedan procesar y entender su significado. El paradigma de utilización de la Web cambia de ser un repositorio de documentos a ser una gran base de conocimiento para avanzados sistemas capaces de ejecutar tareas complejas.

Internet, junto con su cara más visible -la Web- se ha convertido en el almacén más grande de información que jamás existió, una especie de biblioteca de Alejandría de la edad moderna, con la particularidad que cada uno de nosotros (ciudadanos del primer mundo) podemos contribuir a su crecimiento. Hoy en día Internet constituye la Memoria de la Humanidad, que es mucho más de lo que fueron las aspiraciones de sus creadores y patrocinadores en sus comienzos, hace 30 años. La Web ha revolucionado nuestra forma de trabajar, dándonos acceso a contenido en cualquier lugar del mundo a cualquier hora. No obstante, este crecimiento desmesurado también nos trae problemas: al igual que sucede con las publicaciones tradicionales en papel del mundo, somos incapaces de leerlas todas, seguir las novedades, filtrar las interesantes, resumirlas o hacer uso de ellas. Hoy en día la explotación de este inmenso potencial está muy limitada, y el mayor freno a su explosión en términos de utilidad somos nosotros, nuestra capacidad de asimilarlo sin ayuda automática.

En los comienzos la WWW se ceñía al ámbito científico, ofreciendo un número abordable de páginas. El crecimiento del contenido en los años sucesivos dio lugar a los buscadores Internet que nos permiten elegir de toda la inmensa cantidad de documentos aquellos que más nos puedan interesar. Estos buscadores (Yahoo, Lycos, Altavista o Google, que es el más usado, con el 40% de las 375 millones de búsquedas diarias en la red), localizan palabras en documentos mediante algoritmos sofisticados de búsquedas y con una gran inversión en hardware. Y el contenido crece de manera exponencial.

En cuanto al número de usuarios, pasamos de los 1000 del 1990 a los 200 millones del año 2000. Mientras en el mundo del papel nos hemos resignado a este desbordamiento, en la era de los documentos en formato electrónico, como miembros de la sociedad de la información, nos resistimos.

Uno de los padres de Internet y gran visionario, Tim Berners-Lee, premio Príncipe de Asturias del año 2002, y director de consorcio WWW (W3C), está promoviendo una solución para abordar esta explosión de contenido: dejar que los programas informáticos busquen y manejen la información de Internet por nosotros. Es decir, cambiar el paradigma actual de recuperación de documentos para y por humanos hacia la delegación de tareas a software. Los agentes software, como se llaman estos programas, son mucho más rápidos y eficientes para cierto tipo de tareas que el más hábil de los humanos. El problema que se plantea es que la Web, tal y como la conocemos hoy, fue concebida para humanos, contando con nuestra capacidad que entender el lenguaje natural, imágenes y sonidos. Los agentes software van a tardar muchos años, si es que lo consiguen alguna vez, en llegar a este punto. Es por eso que el contenido de la Web debe ser adaptado para ellos. Esto son las bases de la futura Web, concebida tanto para humanos como para agentes software, la llamada: Web Semántica. En otras palabras: la Web actual se basa en un lenguaje que define el 'cómo' visualizar texto e imágenes en una pantalla, mientras la Web Semántica se basa en el 'qué': el significado de los textos e imágenes que vemos en las pantallas, permitiendo así su comprensión por agentes de software.



Uno de los retos que más importancia puede alcanzar es la presencia de una masa crítica de contenido semántico para el desarrollo de aplicaciones sobre dominios reales. En la WWW actual, dónde las primeras páginas fueron diseñadas y creadas con procedimientos manuales dentro de pequeños grupos de investigación, hoy en día, existe una gran proporción del contenido creado de forma automática. Algo similar debe suceder en el proceso de creación del contenido para la Web Semántica. El coste de la creación de un documento procesable por software dentro de los estándares propuestos puede ser mucho más costoso que el mismo contenido orientado para consumo humano. Es por eso que la existencia de procesos automáticos y semiautomáticos de creación de contenidos semánticos es uno de los puntos críticos de su éxito.

La disponibilidad de contenido formal, o conocimiento, para sistemas software no es un problema nuevo. Los sistemas basados en conocimiento y más tarde, los agentes inteligentes, entre otros, se han enfrentado ya a este reto. El éxito alcanzado en la provisión de contenido dependía en cada caso de los sistemas particulares en concreto y del dominio sobre el cual operaban. Cada sistema poseía su propio formato de almacenamiento del conocimiento adaptado a sus propósitos y funcionalidades concretas. El paradigma de la Web Semántica propone un formato común estandarizado, con sintaxis controlada (basada en XML y promovida por el W3C) y con semántica consensuada expresada en modelos formales. Estos lenguajes, a diferencia de los lenguajes actuales de la Web donde se expresa el 'cómo' se debe presentar un contenido, permiten describir la semántica, el 'qué', o cual es el significado del contenido publicado. Estos lenguajes semánticos se organizan en capas permitiendo así diferentes grados de comprensión por parte del software. Además, la Web Semántica está siendo una iniciativa que a nivel mundial engloba a un gran número de centros de investigación, gobiernos, organismos de estandarización y cada vez más empresas de diferentes sectores.

Existen varias formas de abordar el problema de disponibilidad de contenido formal para el software. Una de las posibles soluciones consiste en procesar documentos ya existentes y adaptarlos a los formatos necesarios para que el software los entienda.

Esta línea tiene especial atención en la iniciativa de la Web Semántica, ya que permitiría reutilizar y hacer disponible parte del contenido actual de la WWW. En la parte del estado del arte del presente trabajo se enumeran algunas áreas de investigación, tecnologías y sistemas existentes que permiten procesar documentos online y extraer conocimiento de ellos para propósitos de procesamiento automático. Así mismo se establece una relación entre las distintas clases de fuentes o documentos procesados, desde el punto de vista estructural y la eficiencia de las tecnologías estudiadas.



## 3.2 Arquitectura de la Web Semántica

La Web Semántica está surgiendo como una evolución de la Web actual a la que se agrega una estructura para captar el significado de los contenidos de las páginas y proporcionar un ambiente donde las aplicaciones puedan procesar y relacionar contenidos provenientes de distintas fuentes.

En la propuesta de desarrollo de la Web Semántica del consorcio W3C se sugiere una arquitectura básica en capas (ver Fig. 3.1), comenzando por la capa de nivel inferior XML (eXtended Markup Language) que permite estructurar sintácticamente los datos, siguiendo por una capa RDF que define la semántica de dichos datos, luego por la capa ontológica que define consensuadamente conceptos y relaciones para distintos dominios y por último, la capa lógica, que define las reglas lógicas y mecanismos para hacer inferencias.

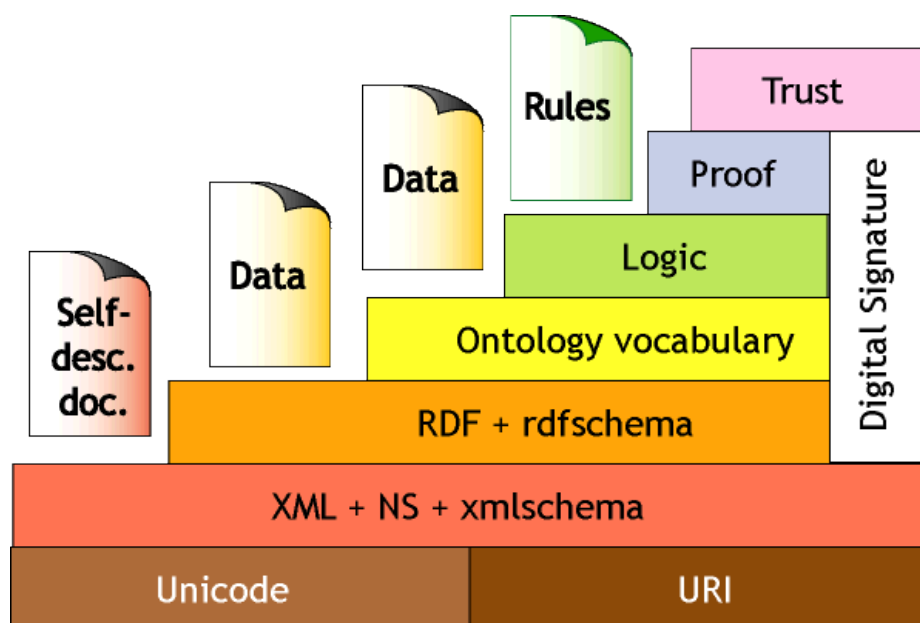


Fig. 9: Estructura en capas de la Web Semántica

De forma resumida, a continuación se presentan las características de las primeras capas de esta estructura:

- XML aporta la sintaxis superficial para los documentos estructurados, pero sin dotarles de ninguna restricción sobre el significado.
- XML Schema es un lenguaje para definir la estructura de los documentos XML.
- RDF es un modelo de datos para los recursos y las relaciones que se puedan establecer entre ellos. Aporta una semántica básica para este modelo de datos que puede representarse mediante XML.
- RDF Schema es un vocabulario para describir las propiedades y las clases de los recursos RDF, con una semántica para establecer jerarquías de generalización entre dichas propiedades y clases.
- OWL, finalmente, añade más vocabulario para describir propiedades y clases. Entre otras mejoras en expresividad, se encuentra la posibilidad de definir relaciones entre clases, cardinalidad, igualdad, tipologías de propiedades más complejas, caracterización de propiedades o clases enumeradas.



### **3.3 Ventajas e inconvenientes de la Web Semántica**

---

Entre las ventajas que pretende aportar la Web Semántica podemos destacar las siguientes:

- Ahorro de tiempo en el procesado de los datos (tiempo de búsqueda, gestión de la información, etc.): la mayor parte de las tareas relativas al procesado de datos podrán realizarse por parte de componentes software automatizados y sin requerir, en muchos casos, intervención humana.
- Resultados más adecuados de búsqueda: se mejorará la precisión de las búsquedas en Web, debido a que el contenido de las páginas Web estará anotado semánticamente y, por tanto, los motores de búsqueda serán capaces de obtener aquellos resultados que más se adecuen a la consulta del usuario. En particular, los motores de búsqueda podrán buscar páginas que se refieran a un determinado concepto de una ontología, al contrario de lo que ocurre ahora, devolviendo todas aquellas páginas que contienen una palabra clave que, en muchos casos, es ambigua.
- Mejora de la comunicación entre servicios Web: la comunicación entre distintos componentes y servicios, sobre todo en aquellos casos en los que los componentes no han sido diseñados para trabajar conjuntamente, ha sido siempre fuente de problemas en cuanto a la interoperabilidad debido, principalmente, a la ambigüedad del lenguaje. El uso de ontologías compartidas (y, posiblemente, mapeadas con otras ontologías) propuesto por la Web Semántica, solventa el problema en la comunicación entre servicios Web gracias a que esta comunicación se produce utilizando conceptos pertenecientes a una ontología.

La Web Semántica mantiene los principios que han contribuido al éxito de la Web actual, como son los principios de descentralización, compartición, compatibilidad, máxima facilidad de acceso y contribución, o la apertura al crecimiento y uso no previstos de antemano (Castells, 2003).

Sin embargo, existen ciertos problemas que deben solucionarse para alcanzar todo el potencial que se le supone a la Web Semántica. En primer lugar, la aparición de la Web Semántica supondrá mayor trabajo para los creadores de páginas Web ya que éstas deberán estar anotadas semánticamente. Por tanto, es crítico el desarrollo de herramientas que permitan a usuarios no experimentados la creación de páginas para la nueva Web Semántica con la misma facilidad con la que éstos lo pueden hacer para la Web tradicional.

Otro de los inconvenientes de la Web Semántica son los efectos que puede producir en relación a la privacidad y la censura. En la actualidad, las técnicas de análisis de textos utilizadas por los gobiernos para controlar los contenidos de la Web son vulnerables a simples modificaciones de palabras (usando, por ejemplo, metáforas) o al uso de imágenes en vez de texto consiguiendo, de esta forma, limitar las posibilidades de censura. En la Web Semántica sería mucho más sencillo controlar la creación y visualización de contenidos Web. Además, la aplicación de propuestas como FOAF (Friend of a Friend), que permite la definición semántica de perfiles de usuario y redes sociales, supondría una limitación severa al anonimato en la Web. Si los investigadores en Web Semántica son capaces de eliminar esta problemática, no cabe duda de que esta tecnología supondrá una clara evolución con respecto a la Web tradicional, transformándose en la nueva Web 3.0.

## 3.4 Servicios Web

### 3.4.1 Introducción e historia

Los Servicios Web proporcionan un mecanismo estándar para que diferentes aplicaciones software que están siendo ejecutadas en diferentes plataformas y/o marcos de trabajo puedan interoperar.

La tecnología de los Servicios Web no es, sin embargo, una idea completamente novedosa. El origen de esta tecnología, en cuanto a principios se refiere, bien se podría remontar a los años 70 del siglo XX con el surgimiento de Internet y la computación distribuida.

La computación distribuida comenzó alrededor de 1970 con el origen de dos tecnologías:

- Minicomputadores (posteriormente denominadas estaciones de trabajo y después ordenadores personales o PCs).
- Redes de ordenadores (ethernet e Internet).

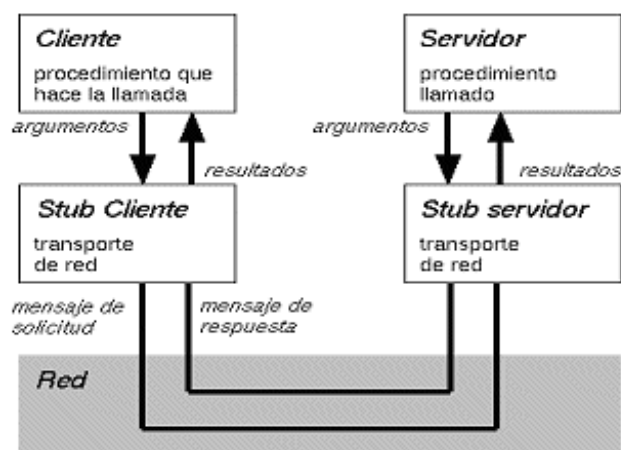
Un sistema distribuido consiste en entidades software heterogéneas y discretas que deben trabajar de forma conjunta para llevar a cabo algunas tareas (Booth et al., 2004).

Además, las entidades software en un sistema distribuido no operan en el mismo entorno de procesamiento, de forma que deben de comunicarse a través de pilas de protocolos hardware/software sobre una red de comunicaciones. Los sistemas de objetos distribuidos son sistemas en los que la semántica de la inicialización de un objeto y la de la invocación de los métodos se expone a sistemas remotos por medio de mecanismos estándares o propietarios que permiten intermediar en las peticiones que salen de los límites del sistema.

Numerosas son las aproximaciones que se han realizado que permiten la construcción de sistemas distribuidos. En este apartado, se destacan las más reconocidas y aquellas que más han influido para la posterior elaboración de la tecnología de los Servicios Web: RPC, DCOM, RMI y CORBA.

#### Llamada a Procedimiento Remoto (RPC).

Es un protocolo que permite a un programa ejecutar una subrutina o procedimiento en otra máquina remota sin necesidad, por parte del programador, de codificar los detalles de la interacción remota (ver Fig. 12). Esta tecnología, que data de mediados de los años 70, supuso un gran avance en el desarrollo de sistemas dentro del paradigma cliente-servidor. Existen numerosas implementaciones de este protocolo como el ONC RPC de Sun, el DCE/RPC de la 'Open Software Foundation' (OSF) y el Modelo de Objetos de Componentes Distribuidos (DCOM) de Microsoft. Estas implementaciones no son, en su mayoría, compatibles entre sí. Las soluciones más populares basadas en objetos (esto es, objetos distribuidos) previas a los Servicios Web se describen brevemente a continuación.



Llamada de procedimiento remoto

Fig. 10: Diagrama de llamada a Procedimiento remoto (RPC).





### **Modelo de Objetos de Componentes Distribuidos (DCOM).**

Es una tecnología propietaria de Microsoft para desarrollar componentes distribuidos sobre varios ordenadores y que se comunican entre sí. La base de DCOM es el Microsoft RPC (MSRPC), una extensión de DCE/RPC de la OSF. Si bien se ha trabajado en la implementación de DCOM para su compatibilidad con sistemas UNIX, esta tecnología fue ideada para su uso sobre el sistema operativo Microsoft. DCOM soporta varios lenguajes de programación, como Visual Basic, C y C++. En la actualidad, la tecnología DCOM ha sido abandonada en favor del framework .NET.

### **Invocación de Métodos Remotos (RMI).**

Es un mecanismo ofrecido en Java para invocar un método remotamente. Al estar integrado en Java, RMI es independiente del sistema operativo pero únicamente soporta el lenguaje de programación Java. La arquitectura de RMI se puede ver como un modelo de cuatro capas. La primera capa es la de aplicación, correspondiendo con la implementación de las aplicaciones cliente y servidor. La capa Proxy es la segunda, también llamada stub-skeleton porque es la que contiene estos componentes que permiten a los desarrolladores elaborar aplicaciones sin tener en cuenta los detalles referentes a la interacción. La tercera capa es la de referencia remota, responsable del manejo de la parte semántica de las invocaciones remotas. Finalmente, la última capa es la de transporte, encargada de realizar las conexiones necesarias y del manejo del transporte de los datos entre las máquinas.

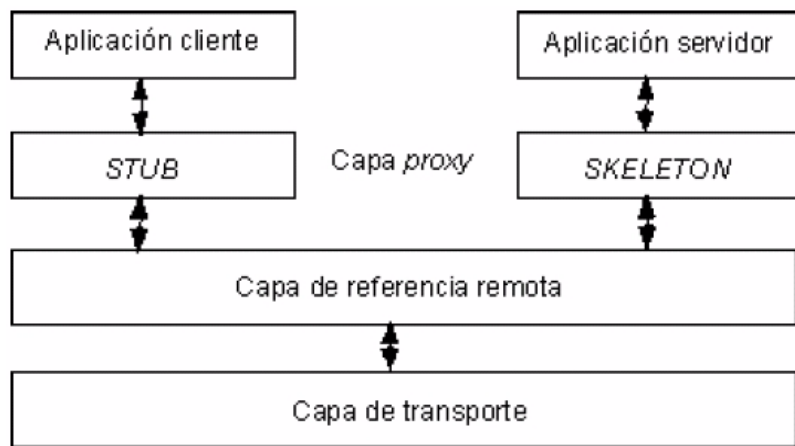


Fig. 11: Diagrama de invocación RMI

### **Arquitectura Común de Intermediarios en Peticiones a Objetos (CORBA).**

El problema de las arquitecturas mencionadas anteriormente, RMI de Sun y DCOM de Microsoft, junto con otras que se desarrollaron de la misma índole (SOM de IBM, OpenDoc de Apple) es que se basan en estándares propietarios, de forma que la interoperabilidad entre ellas es bastante reducida, cuando no nula. Este es el motivo por el cual se desarrolló CORBA, a saber, un estándar que establece una plataforma de desarrollo de sistemas distribuidos, facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos. Fue definido y está controlado por el 'Object Management Group' (OMG), un consorcio que incluye, entre sus compañías y organizaciones constituyentes más renombradas, a Hewlett-Packard, IBM, Sun Microsystems y Apple Computer. CORBA es independiente del sistema operativo y soporta varios lenguajes, entre ellos, Java, C, C++, Ada, Cobol y Visual Basic.

A pesar de propuestas consensuadas como CORBA, los desarrolladores de software no han sido capaces de alcanzar la interoperabilidad deseada. Esto ha sido debido a la falta de aceptación propiciada por la ausencia de Microsoft en el consorcio que elaboró CORBA. Se demostró, por tanto, que para desarrollar estándares aceptados globalmente para la tecnología de componentes distribuidos, era necesario encontrar un conjunto mínimo de elementos comunes. El lenguaje XML y el protocolo HTTP han sido capaces de ofrecer este conjunto mínimo tecnológico para el éxito de la computación distribuida. Estos elementos constituyen la base de los Servicios Web.

### 3.4.2 Breve descripción de la Arquitectura

Como se ha mencionado anteriormente, las soluciones previas a la tecnología de los Servicios Web tenían una serie de problemas, principalmente asociados con la limitación en cuanto a la interoperabilidad. Así, mientras unas se restringían a unas plataformas concretas, otras no eran flexibles en cuanto al lenguaje de programación. En general, el uso de soluciones propietarias suponía un problema en la mayoría de los casos porque, en primer lugar, imposibilitaba la interacción entre los elementos elaborados de acuerdo con las distintas soluciones y, lo que es peor, su uso global estaba



Fig. 12: Diagrama de interacción de Servicios Web

fuertemente restringido por los cortafuegos (firewalls) que bloqueaban los puertos por los que se comunicaban estos componentes. Por tanto, la solución radicaba en encontrar un conjunto de estándares que fueran aceptados mundialmente para construir una tecnología que fuese más acorde a la visión de Internet. XML y HTTP constituyen la base sobre la que se ha desarrollado la tecnología de los Servicios Web.

La tecnología de los Servicios Web se fundamenta en tres estándares principales (ver Fig. 12): el "Web Services Description Language" (WSDL, Lenguaje de Descripción de Servicios Web), el "Simple Object Access Protocol" (SOAP, Protocolo Simple de Acceso a Objetos) y el "Universal Description, Discovery, and Integration" (UDDI, Descripción, Descubrimiento e Integración Universales). De forma resumida, WSDL es un lenguaje basado en XML que permite describir las características de los Servicios Web; SOAP es un protocolo estándar que define cómo dos objetos localizados remotamente pueden comunicarse por medio de intercambio de datos XML y que funciona sobre HTTP (posee también enlaces con otros protocolos de Internet); y UDDI es un registro que proporciona mecanismos

estándares para publicar documentos WSDL que contienen descripciones de Servicios Web y realizar búsquedas sobre los mismos.

Son cinco los pasos principales que constituyen el modus operandi para la utilización de los Servicios Web (ver Fig. 14). En primer lugar, es necesario que la organización o entidad que desee ofrecer un servicio, añada una descripción del mismo (fichero WSDL del servicio) en un registro UDDI, esto es, registre el servicio para su posterior utilización. Esto debe ocurrir para todos los servicios, de todas las organizaciones, que se deseen que sean accesibles globalmente. El siguiente paso, una vez que la descripción de todos los servicios se encuentra almacenada en los registros UDDI accesibles por Internet, consiste en que los clientes consumidores de servicios busquen aquellos que les sean necesarios para alcanzar un objetivo. El proceso de búsqueda lo realiza el propio registro UDDI, que recibe como entrada la funcionalidad esperada del servicio y debe devolver todos aquellos servicios que cumplan con esa funcionalidad. El registro de servicios devuelve al cliente, como respuesta a su petición, una lista con los ficheros de descripción de los servicios (esto es, en WSDL) cuyas capacidades cubran las necesidades del cliente.

Una vez el cliente consumidor de servicio dispone de los ficheros WSDL de todos aquellos servicios a los que el cliente puede acceder para alcanzar un objetivo, aquel debe determinar qué servicio, de entre los de la lista, es el más apropiado de acuerdo con las preferencias del cliente. Cuando el cliente ha seleccionado el servicio más apropiado a ejecutar, realiza la petición al proveedor del servicio a través de un mensaje SOAP. El servicio es entonces ejecutando utilizando los parámetros de entrada indicados en el mensaje SOAP, y el resultado de la ejecución se devuelve al cliente en forma de otro mensaje SOAP, terminando así el proceso de invocación de un Servicio Web.

### 3.4.3 Ventajas e inconvenientes de los Servicios Web

---

La aplicación de arquitecturas orientadas a servicios y, más concretamente, su implementación a través de Servicios Web, conlleva numerosas ventajas de las que pueden beneficiarse los sistemas desarrollados. Una de las mayores ventajas de la tecnología de los Servicios Web es la utilización de protocolos de transporte estándar como HTTP. Esta medida permite el paso de mensajes entre servicios sin necesidad de preocuparse por los sistemas de seguridad y, en particular, los cortafuegos de las distintas organizaciones. En general, el uso de protocolos estándar facilita la interoperabilidad entre plataformas de distintos fabricantes.

Por otro lado, como ya se mencionó anteriormente, los Servicios Web aportan interoperabilidad entre aplicaciones software con independencia de su implementación, del lenguaje de programación utilizado y de la plataforma en que estén instaladas. Esto ha provocado que esta tecnología se utilice, cada vez con más frecuencia, para permitir el uso de sistemas heredados por parte de las nuevas aplicaciones que se desarrollan en el ámbito de una organización. Por su parte, la independencia con respecto a la implementación aporta flexibilidad al sistema, de forma que se puede modificar la implementación del servicio sin necesidad de actualizar su descripción. Además, los Servicios Web permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

Por otra parte, no son numerosos pero sí importantes los problemas que presentan los Servicios Web y el paradigma SOA. En general, el mayor limitante de la tecnología de los Servicios Web es su rendimiento. Comparado con otros modelos de computación distribuida como RMI, DCOM o CORBA, la prestación de esta solución es baja. Este problema viene ocasionado por el uso de XML como lenguaje para la codificación de mensajes. En particular, el problema radica en la necesidad de parsear y componer los ficheros XML, lo cual requiere mayor capacidad computacional y hace que las aplicaciones se ejecuten más lentamente. Como resultado del esfuerzo por resolver este limitante, la W3C creó en 2003 el grupo de trabajo para la caracterización binaria de XML ("XML Binary Characterization Working Group") que en 2005 se transformó en el grupo de trabajo para el intercambio eficiente de XML ("Efficient XML Interchange Working Group"). El objetivo final ha sido el

desarrollo de una especificación de un formato de codificación que permita el intercambio eficiente de documentos XML.

Otro de los inconvenientes de los Servicios Web es que el grado de madurez de algunas de sus especificaciones no puede compararse con el grado de desarrollo de estándares abiertos en computación distribuida como CORBA. Hasta ahora, los investigadores en Servicios Web se han preocupado más de cómo realizar tareas de alto nivel como el descubrimiento y la invocación de servicios, dejando sin cubrir aspectos más básicos como la seguridad, la gestión de transacciones, etc. Hoy en día existen numerosos grupos de trabajo para el desarrollo de este tipo de especificaciones (WS-\*: WS-Security, WS-Trust, WS-Policy, WS-Privacy, WS-ReliableMessaging, WS-Routing, WS-Addressing, etc.). Esto introduce un plus de riesgo cuando se trata del desarrollo de sistemas comerciales que deben ser aplicados en entornos reales.

Disponiendo de todos los componentes que constituyen la tecnología de Servicios Web, cualquiera podría hacer uso de los servicios ofrecidos por todos los Servicios Web disponibles en Internet. Sin embargo, y a medida que la Web crece en tamaño y diversidad, cada vez es más necesaria la automatización de aspectos relacionados con los Servicios Web como el descubrimiento, la selección, la composición y la invocación. De hecho, una de las principales ventajas de esta tecnología es la posibilidad de componer de forma dinámica servicios utilizando componentes software independientes y reutilizables. El problema, entonces, es que la tecnología actual (en torno a UDDI, WSDL y SOAP) no proporciona los medios para conseguir este dinamismo.



### 3.5 Servicios Web Semánticos

Para resolver los limitantes mencionados en el mundo de los Servicios Web, y asimilando conceptos surgidos con la aparición de la Web Semántica, se originó lo que se ha resuelto en denominar Servicios Web Semánticos. Éstos se pueden definir como la aplicación conjunta de conceptos de Servicios Web y de Web Semántica para crear Servicios Web inteligentes. Esta tecnología es un nuevo paso adelante en la evolución de la Web, que consiste en describir a los Servicios Web con contenido semántico de forma que el descubrimiento de servicios, su composición e invocación se pueda realizar de forma automática por parte de entidades software (p.ej., mediante agentes inteligentes) capaces de procesar la información semántica disponible. En la siguiente figura, se pueden distinguir las principales etapas en el desarrollo de la Web hasta el origen de los Servicios Web Semánticos:

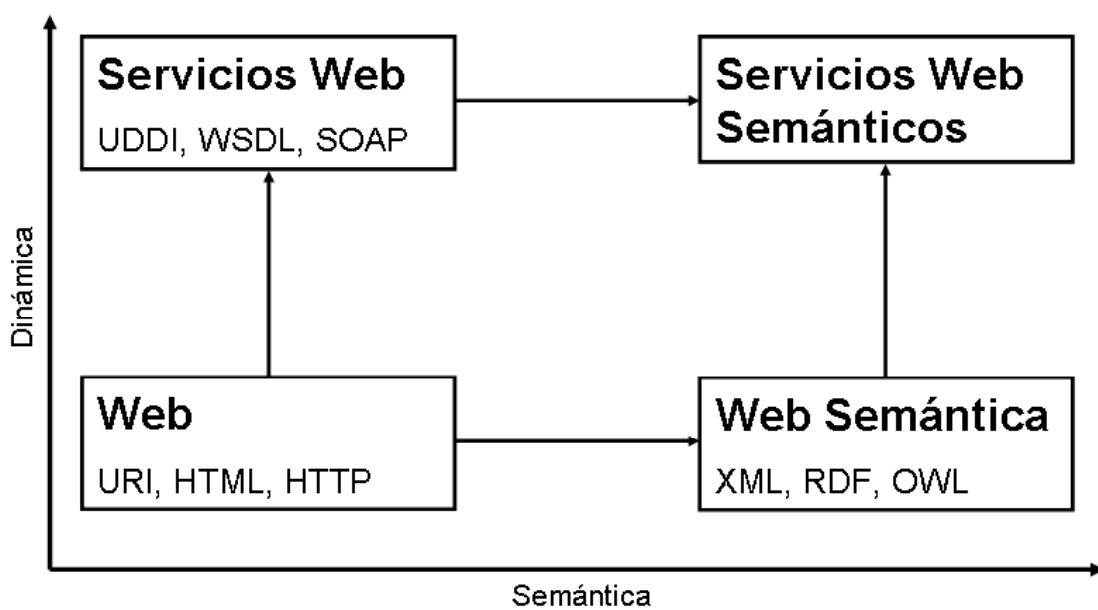


Fig. 13: Esquema de desarrollo e influencia hasta llegar a los Servicios Web Semánticos

En la actualidad, el W3C está examinando cinco aproximaciones a los Servicios Web Semánticos con el propósito de alcanzar un estándar para esta tecnología: OWL-S, WSMO, SWSF, WSDL-S y SAWSDL.

#### 3.5.1 Web Ontology Language for Services (OWL-S)

El origen de la Web Semántica permitió el acceso a funcionalidad en la Web nunca antes concebida. En este entorno, tanto usuarios como agentes software deben ser capaces de realizar ciertos procesos con un alto grado de automatización. Entre éstos, se pueden destacar los de descubrir, invocar, componer y monitorizar recursos Web que ofrecen determinados servicios y poseen propiedades particulares. OWL-S (anteriormente DAML-S) es una ontología de servicios que hace todo esto posible (OWL-S, 2004). En OWL-S, un servicio se entiende como “un sitio Web que no solamente provee información estática sino que permite efectuar alguna acción o cambio en el mundo, tal como la venta de un producto o el control de un dispositivo físico”. Así, para que un agente software pueda acceder a un Servicio Web, es necesaria una descripción de las capacidades del servicio y de cómo acceder al mismo que sea interpretable por componentes software. Con este propósito se hace uso del concepto de ontología, en particular del lenguaje ontológico OWL y de los mecanismos asociados al mismo.

Una característica a destacar en OWL-S es la distinción que hace entre servicios “atómicos” y servicios “compuestos”. Los servicios atómicos son aquellos donde un solo programa accesible por la Web, sensor o dispositivo es invocado por un mensaje de solicitud, realiza su tarea y, posiblemente, produce una única respuesta a la solicitud. Por su parte, los servicios compuestos (o complejos) son aquellos que están constituidos por múltiples servicios primitivos y pueden requerir una interacción extendida o conversación entre el solicitante, y que el conjunto de servicios que están siendo utilizados. El objetivo último de OWL-S es hacer posible la ejecución de las siguientes tareas sobre los servicios disponibles en la Web:

1. Descubrimiento automático de servicio Web. Se trata de automatizar el proceso de localización de los servicios Web que puedan proporcionar una clase particular de capacidades a la par que cumple unas determinadas restricciones especificadas por el cliente. Para esto, OWL-S permite anunciar de forma declarativa las propiedades y capacidades de los servicios.
2. Invocación automática de servicio Web. A través de esta tarea, se intenta realizar la ejecución automática de un servicio Web por parte de un programa computerizado o agente, dada únicamente una descripción declarativa de ese servicio. El marcado de servicios Web de OWL-S proporciona una API declarativa e interpretable por la máquina que incluye la semántica de los argumentos a especificar cuando se ejecutan las llamadas, así como la semántica de los mensajes que se producen como resultado de la ejecución del servicio, tanto cuando éste falla, como cuando tiene éxito.
3. Composición e interoperación automática de servicio Web. Dada la descripción de alto nivel de un objetivo, se permite realizar una tarea compleja a partir de la selección automática, composición e interoperación de diversos servicios Web. Para dar soporte a estas tareas, OWL-S proporciona especificaciones declarativas de los prerequisites y consecuencias de la aplicación de servicios individuales y un lenguaje para describir composiciones de servicio e interacciones de flujos de datos.

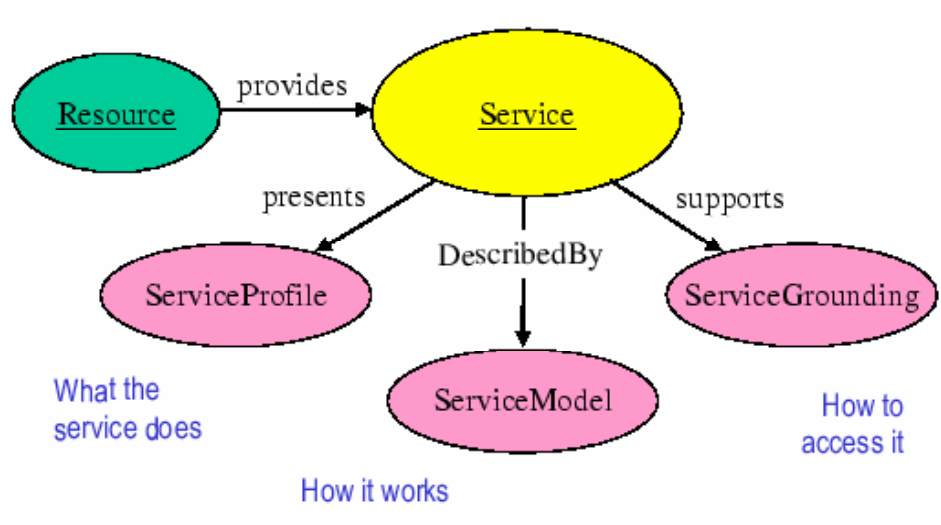


Fig. 14: Ontología de alto nivel OWL-S

En esencia, OWL-S es una ontología que contiene los elementos fundamentales que caracterizan un servicio y que permite describir las capacidades que sustenta un servicio. Esta ontología contiene tres tipos de conocimiento fundamentales:

1. qué es lo que un servicio hace (ServiceProfile, utilizado para publicitar el servicio),
2. cómo se usa el servicio (ServiceModel), y
3. cómo interactuar con el servicio (ServiceGrounding, detalles sobre los protocolos de transporte).



### 3.5.2 Web Services Modeling Ontology (WSMO)

La propuesta relacionada con la descripción semántica de los Servicios Web que cronológicamente siguió a OWL-S fue WSMO. Con una visión similar a la de los autores de OWL-S, WSMO proporciona un marco de trabajo conceptual y un lenguaje formal para describir de forma semántica todos y cada uno de los aspectos relevantes relacionados con los servicios Web para, así, facilitar la automatización de tareas tales como el descubrimiento, la combinación y la invocación de servicios electrónicos sobre la Web. En WSMO, un servicio Web se define como una entidad computacional capaz, una vez invocada, de satisfacer el objetivo de un usuario.

WSMO está basado en el modelo conceptual propuesto en el “Web Service Modeling Framework” (WSMF), que identifica cuatro elementos fundamentales para describir servicios Web semánticos:

1. Ontologías: proporcionan la terminología que será usada por los restantes elementos.
2. Objetivos: representan los deseos de los usuarios o intenciones que deben ser satisfechas por algún servicio Web.
3. Descripciones de servicios Web: define los aspectos funcionales y de comportamiento de un servicio Web.
4. Mediadores: tienen el propósito de gestionar de forma automática los problemas de interoperabilidad que surjan entre los restantes elementos.

Teniendo como base los conceptos identificados en WSMF, WSMO proporciona una especificación ontológica para los elementos que conforman el núcleo de los servicios Web semánticos. A diferencia de OWL-S, el lenguaje de ontologías utilizado con este propósito en WSMO es WSML. En la Fig. 15 se muestran los conceptos de alto nivel que constituyen la ontología de WSMO.

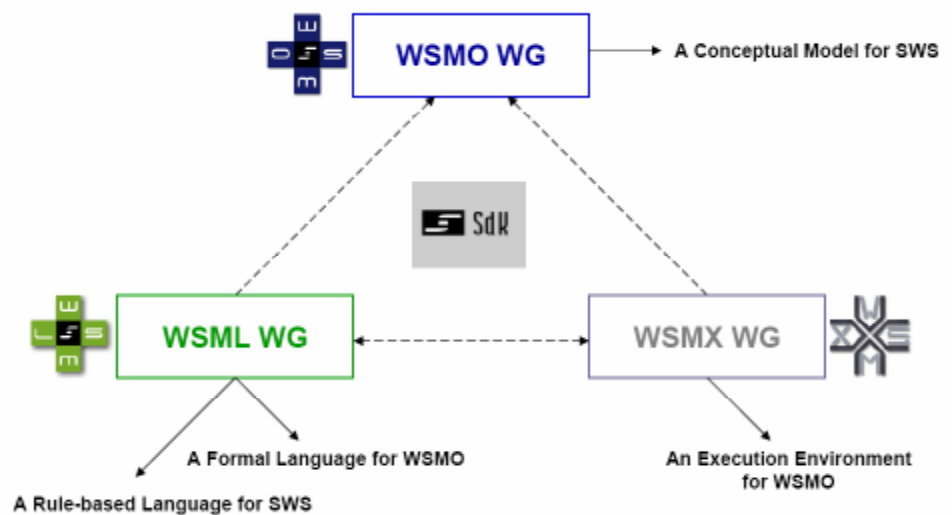


Fig. 15: Visión general de funcionamiento de WSMO

### 3.5.3 Semantic Web Services Framework (SWSF)

---

El “Semantic Web Services Framework” (SWSF) es una nueva iniciativa para la definición de una especificación semántica más rica sobre la actual tecnología de los Servicios Web que permita una mayor automatización y flexibilidad en la provisión y uso de servicios. El marco de trabajo está diseñado a su vez con el propósito de soportar la construcción de herramientas y metodologías más potentes en el entorno de los Servicios Web, así como promocionar el uso de procesos de razonamiento sobre servicios fundamentados en semántica. En este sentido, y al igual que la aproximaciones anteriores, SWSF pretende incorporar una semántica más rica que dé soporte a una mayor automatización de tareas como la selección e invocación de servicios, traducción del contenido de los mensajes entre servicios heterogéneos que operan entre sí, composición de servicios, etc., además de permitir aproximaciones integrales para la monitorización de servicios y la recuperación de errores.

Esta propuesta está compuesta de dos componentes principales: el “Semantic Web Services Language” (SWSL) y el “Semantic Web Services Ontology” (SWSO). El SWSL es un lenguaje lógico de propósito general que incluye ciertas características para hacerlo más apropiado para las necesidades de la Web y los Servicios Web Semánticos. Entre estas peculiaridades, se incluye el uso de URIs, la integración de los tipos que forman parte de XML y el uso de mecanismos de importación y espacios de nombres compatibles con XML. Este lenguaje es el utilizado para especificar las caracterizaciones formales sobre los conceptos relacionados con los Servicios Web y sus descripciones. Incluye dos sub-lenguajes: SWSL-FOL, basado en lógica de primer orden con extensiones de HiLog y sintaxis de marcos de F-Logic, y se utiliza para expresar la caracterización formal de los conceptos relacionados con los Servicios Web (lenguaje ontológico), y SWSL-Rules, basado en el paradigma de programación lógica, y utilizado para dar soporte al uso de la ontología de servicios en procesos de razonamiento y en entornos de ejecución basados en ese paradigma.

La SWSO define un modelo conceptual por el cual los Servicios Web pueden ser descritos, y una representación formal o axiomatización de dicho modelo. La axiomatización completa se implementa en lógica de primer orden utilizando SWSL-FOL, con una semántica de Teoría de Modelos que especifica un significado preciso de los conceptos de la ontología. Esta forma de la ontología que utiliza lógica de primer orden se denomina “First-Order Logic Ontology for Web Services” (FLOWS). Adicionalmente, los axiomas de FLOWS se han traducido al lenguaje de reglas SWSL-Rules, obteniéndose, de este modo, una ontología basada en semántica de programación lógica denominada “Rules Ontology for Web Services” (ROWS). Por tanto, FLOWS, al igual que WSMO y OWL-S (sobre el cual se fundamenta y extiende), es una ontología formal sobre conceptos de servicios, que proporciona un marco de trabajo conceptual para describir y razonar sobre servicios.

En FLOWS, un servicio es un objeto conceptual que corresponde a un servicio Web (u otro servicio accesible de forma electrónica). Una contribución clave de la ontología FLOWS sobre OWL-S es el desarrollo de un modelo de procesos de comportamiento rico, basado en el lenguaje de especificación de procesos PSL (“Process Specification Language”), que es un estándar ISO que proporciona una ontología extensible, con diferentes capas, para especificar las propiedades de los procesos. Este modelo de procesos permite la interoperabilidad entre los distintos modelos de procesos sobre Servicios Web que están apareciendo en la actualidad, y hace posible la automatización de tareas en línea con la perspectiva de los servicios Web semánticos.





### 3.5.4 Web Service Semantics (WSDL-S)

WSDL-S (WSDL-S, 2005) cambia radicalmente con respecto a las perspectivas “tradicionales” para la incorporación de semántica a los Servicios Web. WSDL-S define un mecanismo para asociar anotaciones semánticas con Servicios Web que han sido descritos utilizando WSDL. A diferencia de los lenguajes descritos anteriormente, WSDL-S asume la existencia de modelos semánticos del dominio relevante para cada servicio, mantenido fuera del ámbito de los documentos WSDL y que pueden ser referenciados desde un documento WSDL a través de elementos de extensibilidad ideados como parte de la propuesta WSDL-S.

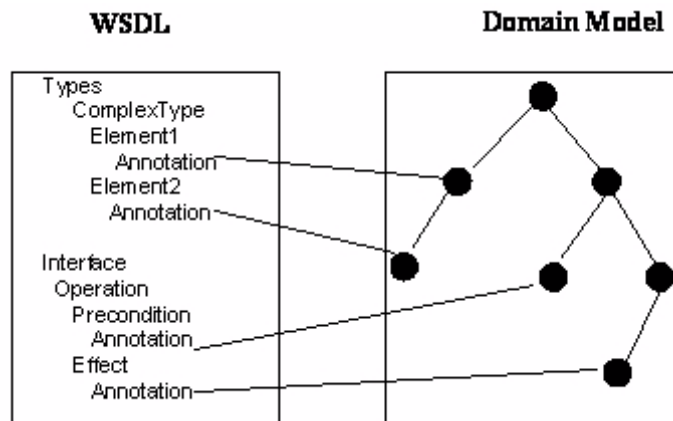


Fig. 16: Anotación semántica de los elementos de WSDL

El fundamento conceptual de esta aproximación se basa, al igual que en los casos anteriores, en el hecho de que el estándar WSDL actual opera a un nivel sintáctico que carece de la expresividad semántica necesaria para representar los requisitos y capacidades de los servicios. De este modo, la incorporación de semántica puede mejorar la reutilización de software y el descubrimiento, facilitar la composición de servicios Web y permitir la integración de aplicaciones heredadas como parte de la integración de procesos de negocio. La información semántica que se considera por parte de WSDL-S incluye las definiciones de precondiciones, entradas, salidas y efectos de las operaciones de servicios Web.

Entre las ventajas de esta novedosa aproximación sobre otras, como OWL-S y WSMO, se pueden destacar las siguientes:

1. Los usuarios pueden describir, de forma incremental, todos los detalles, tanto semánticos como a nivel de operaciones en WSDL, un lenguaje que es familiar para la comunidad de desarrolladores.
2. Al externalizar los modelos semánticos del dominio, WSDL-S permanece independiente del lenguaje de representación de ontologías que se desee utilizar. Esto permite a los desarrolladores de Servicios Web anotar sus servicios con el lenguaje ontológico de su elección (p.ej., UML, OWL, etc.). Esto supone una ventaja adicional, porque reutilizar modelos del dominio existentes expresados con lenguajes de modelado como UML puede acelerar la incorporación de anotaciones semánticas.
3. Es relativamente fácil modificar las herramientas existentes actualmente alrededor de la especificación WSDL para incorporar los elementos propuestos por esta aproximación. En todo caso, es más rápido y fiable que el desarrollo de herramientas implementadas desde cero.

### 3.5.5 Semantic Annotations for Web Services Description Language (SAWSDL)

---

SAWSDL es una extensión, desarrollada por un grupo de trabajo del W3C, del lenguaje de descripción de servicios Web (WSDL). Está constituido por dos tipos básicos de anotaciones: la referencia al modelo y el mapeo del esquema.

Las anotaciones de referencia al modelo son las mismas que las empleadas en el modelo WSDL-S8, se utilizan para asociar interfaces, puertos, operaciones, entradas, salidas, elementos xml y atributos con conceptos semánticos.

En cuanto al mapeo del esquema, son atributos añadidos a la declaración del esquema XML de los elementos, para especificar mapeos entre la información semántica y XML. Se utilizan durante la invocación de los servicios, para formatear la información del cliente a XML, de forma que pueda ser entendida por el servicio Web, solucionándose el problema de la estructura de las entradas y salidas del servicio. Normalmente suele ir asociado a una hoja de transformación XSLT.

A diferencia de WSDL-S, permite especificar semánticamente el comportamiento de los servicios Web, mediante la utilización de lenguajes de descripción de ontologías como OWL, siendo útil para la realización de coreografías de servicios.

Permite el desarrollo de las siguientes funcionalidades:

- Clasificación de servicios
- Descubrimiento
- Matching
- Composición
- Invocación dinámica



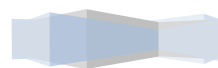
### 3.5.6 Ventajas e inconvenientes de los SWS

---

Desde su concepción, la Web Semántica en general y los Servicios Web Semánticos en particular han estado asociados al concepto de agente. Refiriéndose a agente, en este ámbito, como aquella entidad software autónoma (o sea, sin necesidad de intervención humana) que es capaz de sacar provecho de los datos que han sido anotados semánticamente como lo podría hacer un usuario humano.

Por tanto, centrándose en la tecnología de Servicios Web Semánticos, el mercado semántico de la funcionalidad ofrecida por los servicios, permite que agentes software con las habilidades adecuadas sean capaces de hacer uso de estos servicios para satisfacer objetivos de alto nivel. Desde un punto de vista más integral, y teniendo en cuenta las posibilidades ofrecidas por todas y cada una de las aproximaciones existentes en la actualidad para la provisión de Servicios Web Semánticos, los agentes deben ser capaces de realizar de forma autónoma las tareas de descubrimiento, selección, composición, ejecución y monitorización de los servicios, adaptándose, en cada momento, a los cambios que se pueden producir en el entorno debido al dinamismo propio de este tipo de sistemas.

En resumen, el “problema” de la tecnología de Servicios Web Semánticos, desde su propia concepción, es su dependencia de entidades software de nivel superior con capacidades cognitivas que posean la capacidad de acceder al contenido semántico de las descripciones de los servicios, procesarlo y entenderlo de modo que puedan hacer un uso efectivo y apropiado de la funcionalidad que estos servicios ofrecen.



## 4 Ontologías

---

### 4.1 Definición de Ontología

---

En el diccionario de la Real Academia de la Lengua Española se define el concepto ‘ontología’ como la “parte de la metafísica que trata del ser en general y de sus propiedades transcendentales”. Y es que fue la filosofía el primer campo de conocimiento donde se utilizó este término. Fue a partir de los primeros años de la década de los noventa cuando las ontologías comienzan a tener mayor protagonismo entre la ciencia de la computación y, más concretamente, en Inteligencia Artificial (en particular en Ingeniería del Conocimiento) y Procesamiento de Lenguaje Natural.

Sin embargo, la definición más citada del término Ontología en el área de Inteligencia Artificial es la de Gruber “Una ontología es la especificación explícita de una conceptualización.”

Para la Inteligencia Artificial lo que existe es exactamente aquello que puede ser representado computacionalmente. Cuando el conocimiento de un dominio es implementado en un formalismo declarativo, el conjunto de objetos que pueden ser representados es llamado el universo de discurso. Este conjunto de objetos, y las relaciones descriptibles entre ellos, se refleja en el vocabulario con el cual se representa el conocimiento.

En la especificación de una ontología se asocian mediante definiciones, los nombres de las entidades del ámbito en cuestión, (por ejemplo clases, relaciones, funciones u otros objetos), con cierto texto legible por el ser humano que las describe y con axiomas que restringen su interpretación.

Borst modificó ligeramente la definición de Gruber: “Las ontologías se definen como la especificación formal de una conceptualización compartida”.

Studer y sus colegas agregaron expresividad a las definiciones de Gruber y Borst explicitando:

- **Conceptualización** se refiere a un modelo abstracto de algún fenómeno en el mundo, proveniente de haber identificado los conceptos relevantes de dicho fenómeno.
- **Explícita** se refiere a que los conceptos usados y las restricciones para su uso se definen explícitamente.
- **Formal** se refiere al hecho de que la ontología debería ser legible o interpretable por una computadora.
- **Compartida** refleja la noción de que una ontología captura conocimiento consensuado, es decir, no es conocimiento privado de un individuo, sino aceptado por un grupo o comunidad.

Por tanto, se puede comprobar que, a día de hoy, no existe una definición consensuada del término ontología. Muy al contrario, incluso los autores de las definiciones más reconocidas y aceptadas se replantean las mismas.

Finalmente, se puede destacar una definición extensiva de ontología indicando los componentes que la forman. En general, las ontologías proporcionan un vocabulario común de un área y definen, a diferentes niveles de formalismo, el significado de los términos y relaciones entre ellos. El conocimiento en ontologías se formaliza principalmente usando seis tipos de componentes: clases, atributos, relaciones, funciones, axiomas e instancias:

- Una clase puede ser algo sobre lo que se dice algo, como por ejemplo un tipo de objeto, la descripción de una tarea, función, acción, estrategia, proceso de razonamiento, etc. Las clases en la ontología se suelen organizar en taxonomías. En todo caso, cabe destacar que ontología



y taxonomía son dos elementos diferentes, aunque algunas veces la noción de ontología se diluye en el sentido que las taxonomías se consideran ontologías completas (Studer et al., 1998). Se suelen usar indistintamente los términos 'clase' y 'concepto'.

- Los atributos representan la estructura interna de los conceptos. Atendiendo a su origen, los atributos se clasifican en específicos y heredados. Los atributos específicos son aquellos que son propios del concepto al que pertenecen, mientras que los heredados vienen dados por las relaciones taxonómicas en las que el concepto desempeña el rol de hijo y, por tanto, hereda los atributos del padre. Los atributos vienen caracterizados por el dominio en el cual pueden tomar valor.
- Las relaciones representan un tipo de interacción entre los conceptos del dominio. Se definen formalmente como cualquier subconjunto de un producto de  $n$  conjuntos, esto es:  $R: C_1 \times C_2 \times \dots \times C_n$ . Entre los distintos tipos de relaciones posibles, se encuentran las relaciones taxonómicas ("es\_un") y las mereológicas o partonómicas ("parte\_de") como relaciones binarias más destacadas.
- Las funciones son un tipo especial de relaciones en las que el  $n$ -ésimo elemento de la relación es único para los  $n-1$  precedentes. Formalmente, definimos las funciones ( $F$ ) como:  $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$ . Como ejemplos, se pueden mencionar las funciones "madre de" y "precio de un coche usado".
- Los axiomas son expresiones que son siempre ciertas. Pueden ser incluidas en una ontología con muchos propósitos, tales como definir el significado de los componentes ontológicos, definir restricciones complejas sobre los valores de los atributos, argumentos de relaciones, etc., verificando la corrección de la información especificada en la ontología o deduciendo nueva información.
- Por último, las instancias son las ocurrencias en el mundo real de los conceptos. En una instancia todos los atributos del concepto tienen asignado un valor concreto.

## 4.2 Tipos de Ontologías

---

Existen diferentes clasificaciones de tipos de ontologías. Se siguen principalmente dos criterios para estas clasificaciones: (a) el tipo de conocimiento que contienen; y (b) la motivación de la ontología. A continuación, se enumeran los distintos tipos de ontologías para cada una de estas posibles clasificaciones.

### 4.2.1 Clasificación por el conocimiento que contienen

---

Gertjan van Heijst y sus colegas propusieron los siguientes tres tipos de ontologías atendiendo al conocimiento que contienen:

- Ontologías terminológicas, lingüísticas: especifican los términos para representar conocimiento en un dominio determinado.
- Ontologías de información: especifican la estructura de los registros de la base de datos. Los esquemas de bases de datos son un ejemplo.
- Ontologías para modelar conocimiento: especifican conceptualizaciones de conocimiento. Este tipo de ontologías tienen una estructura interna mucho más rica que los anteriores y son las que interesan a los desarrolladores de sistemas basados en conocimiento.

### 4.2.2 Clasificación por motivación

---

La segunda clasificación se hace dependiendo del motivo por el que se elabora de la ontología. Tradicionalmente se distinguen cuatro tipos principales de acuerdo con su motivación:

- Ontologías para la representación de conocimiento: permiten especificar las conceptualizaciones que subyacen a los formalismos de representación del conocimiento, por lo que también se denominan meta-ontologías (meta-level o top-level ontologies).
- Ontologías genéricas: definen conceptos generales y fundacionales del conocimiento como las estructuras parte/todo, la cuantificación, los procesos o los tipos de objetos. Estas ontologías son reutilizables en diferentes dominios.
- Ontologías del dominio: permiten representar el conocimiento especializado pertinente de un dominio o subdominio, como la medicina, las aplicaciones militares, la cardiología, etc.
- Ontologías de aplicación: están ligadas al desarrollo de una aplicación concreta. Tales ontologías cubren los aspectos relacionados con aplicaciones particulares. Típicamente estas ontologías toman conceptos de ontologías del dominio y genéricas, así como métodos específicos para realizar la tarea, por lo que frecuentemente no pueden ser reutilizadas.



## 5 Tecnología de Agentes

---

### 5.1 ¿Qué son los Agentes?

---

La auténtica potencia de la Web semántica no se hará efectiva hasta que se creen muchos programas que recopilen contenidos de la Web tomados de diversas fuentes, procesen la información e intercambien sus resultados con otros programas. La eficacia de estos agentes informáticos crecerá exponencialmente conforme vayan estando disponibles en la Web más contenidos entendibles por computadoras y más servicios automáticos (entre ellos, otros agentes). La Web semántica promueve esta sinergia: incluso agentes no expresamente diseñados para trabajar en colaboración pueden transferirse datos uno al otro si éstos se hallan provistos de semántica.

Las firmas digitales constituirán otra característica esencial. Las firmas digitales son bloques encriptados de datos que las computadoras y los agentes pueden utilizar para comprobar que la información agregada procede de una fuente identificada y digna de confianza. Parece obvio que los agentes deberían desconfiar de las aserciones que leyeran en la Web semántica si no han verificado las fuentes de información.

Aunque existen ya en la Web muchos servicios automáticos, por carecer de semántica los demás programas y los agentes en particular no tienen forma de localizar uno que realice una tarea específica. Este proceso, la búsqueda de servicios, sólo podrá producirse cuando exista un lenguaje común (ontología) para describirlos de modo tal que otros agentes puedan "comprender" tanto la función ofrecida como la manera de aprovecharla. Servicios y agentes pueden dar a conocer su función depositando, por ejemplo, sus descripciones en anuarios semejantes a las Páginas Amarillas.

En la actualidad se dispone de unos pocos sistemas de búsqueda de servicios, pero son de bajo nivel. Estas iniciativas, se basan en la estandarización de un conjunto predeterminado de descripciones de funciones. Pero la estandarización no puede ir demasiado lejos, pues no hay forma de prever todas las necesidades futuras.

La Web semántica es, por el contrario, más flexible. Los agentes productor y consumidor pueden llegar a entenderse mediante el intercambio de ontologías, que proporcionan el vocabulario requerido para "dialogar". Los agentes podrían incluso crear por sí mismos nuevas capacidades de razonamiento cuando descubrieran nuevas ontologías.

Un proceso típico consistirá en la creación de una "cadena de valores" donde se van transfiriendo subcadenas de información de un agente a otro, cada uno de los cuales "añade un valor", tratando de construir el resultado definitivo solicitado por el usuario final. Por ejemplo, si hubiera agentes que brindaran servicios automáticos de búsqueda de métricas adecuadamente definidas para una necesidad específica, éstos podrían comunicar los resultados a herramientas automáticas de medición para obtener los valores de las mediciones, que a su vez podrían ser pasados a otro agente para su análisis y así siguiendo.

Cabe destacar que para crear automáticamente bajo demanda cadenas de valor complejas, algunos agentes se valdrán no sólo de la Web semántica, sino también de técnicas de inteligencia artificial. Pero la Web semántica proporcionará los cimientos y la estructura necesarios para hacer más realizables tales técnicas.

En la siguiente etapa, la Web semántica desbordará el mundo virtual y se extenderá hasta nuestro mundo físico. Las URI pueden apuntar a cualquier cosa, sin exceptuar a las entidades materiales.

Significa esto que podemos servirnos de ontologías para describir objetos, como teléfonos móviles o aparatos de televisión. Tales equipos pueden anunciar, al estilo de los agentes informáticos, las funciones que pueden realizar y el modo en que pueden ser controlados.

Ya se han dado los primeros pasos en esta dirección. Se ha trabajado en el desarrollo de una norma para la descripción de las capacidades funcionales de los equipos (por ejemplo, los tamaños de pantalla) y las preferencias del usuario. Este estándar, construido sobre el RDF, ha sido llamado CC/PP (Composite Capability/Preference Profile). En un principio, permitirá que los teléfonos móviles y otros clientes no estándar de la Web describan sus características de modo que el contenido de la Web pueda ajustarse para ellos sobre la marcha.





## **5.2 Agentes Inteligentes**

---

No existe una definición consensuada del término 'agente' ni acuerdo en cuanto a las propiedades que este tipo de entidades debe de presentar. El diseño de agentes inteligentes es una rama del mundo de la Inteligencia Artificial. En este dominio, una de las definiciones de agente más citadas es la establecida por Russell y Norvig (2004):

“Un agente es cualquier cosa capaz de percibir su medioambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores.”

Esta definición se centra en el componente físico del término y en su interacción con el mundo que le rodea. Acercándonos más a la parte funcional del concepto, una definición comúnmente aceptada es la propuesta por Wooldridge y Jennings (1995), posteriormente adaptada por Wooldridge (2000):

“Un agente es un sistema computerizado que está situado en algún entorno, y que es capaz de actuar de forma autónoma en este entorno para satisfacer sus objetivos de diseño.”



### **5.3 Sistemas Multi-Agente**

---

Los agentes pueden ser útiles como entidades independientes en entornos aislados a las que se les delegan ciertas tareas repetitivas y que se pueden automatizar en representación de unos determinados usuarios. Sin embargo, en la mayoría de los casos, los agentes se encuentran en entornos que contienen otros agentes constituyendo de este modo un Sistema Multi-Agente (SMA), esto es, un sistema constituido por un grupo de agentes que pueden interactuar (Vlassis, 2003). Una definición de SMA más elaborada es la propuesta por el Laboratorio de Agentes Software Inteligentes (2001):

“Un Sistema Multi-Agente (SMA) es una red poco acoplada de agentes software que interactúan para resolver problemas que van más allá de las capacidades o conocimiento individual de cada uno de los componentes que resuelven problemas.”

Cuando un grupo de agentes individuales forma un SMA, surge la necesidad de disponer de un mecanismo para coordinar dicho grupo de agentes y de un lenguaje para permitir la comunicación entre ellos. Entre los mecanismos de coordinación, se pueden distinguir los casos en los que los agentes tienen objetivos comunes y, por tanto, cooperan, de los casos en los que los agentes son “auto-interesados” y tienen objetivos conflictivos con los demás, para lo cual precisaremos de mecanismos de negociación. De forma análoga a esta clasificación, Wooldridge (2002) distingue entre sistemas distribuidos de resolución de problemas (constituidos por agentes diseñados explícitamente para conseguir de forma cooperativa un objetivo dado) y sistemas abiertos (donde confluyen agentes elaborados por distintos desarrolladores y que poseen posiblemente objetivos diferentes).

En el caso de agentes cooperativos, los mecanismos de cooperación más comunes son las estructuras organizacionales, la planificación multi-agente (centralizada y distribuida), redes de contratos y cooperación funcionalmente exacta. Por otro lado, para el caso de agentes competitivos, se hace necesario un mecanismo de negociación. Entre los tipos de mecanismos de negociación más utilizados en la literatura destaca la formación de coaliciones, los mecanismos de mercados, la teoría del regateo, la votación, las subastas y la asignación de tareas entre dos agentes.



## 6 Comparación entre Web Actual y Web Semántica

---

Para comprender que es la Web Semántica conviene comprender claramente como funciona la Web actual de hoy en día.

Es importante saber como surge y cuales fueron las motivaciones que la originaron, y como a lo largo del tiempo ha evolucionado y se ha adaptado a las necesidades del momento, sus características, y finalmente cuales son las limitaciones que presenta y que hace precisa la aparición de la Web Semántica.

### 6.1 Orígenes de la Web Actual

---

La World Wide Web, o simplemente Web, tiene su origen haya por el año 1989 en los laboratorios del CERN. Al contrario de lo que generalmente se cree, la Web no fue un invento americano, sino europeo. El padre de la Web fue Tim Berners-Lee quien en 1990 completó el primer servidor Web y el primer cliente, publicando el primer borrador de las especificaciones del HTML y del protocolo HTTP.

La idea de la Web era disponer de un sistema que permitiera servir como base para el intercambio de información entre investigadores, de forma que resultase cómodo y sencillo el proceso. No obstante, no es hasta la publicación del navegador Mosaic en el año 1993 por parte de la NCSA, cuando la Web se conoce a nivel mundial, extendiéndose primero por universidades y laboratorios y seguidamente al gran público.

La idea original de Tim Berners-Lee, era crear un sistema que sirviese de base para el intercambio de información entre investigadores, así como permitir la revisión de referencias que existiesen en el documento original mientras este se leía y siempre y cuando el usuario quisiera hacerlo. Es así como surge la que sea probablemente la primera aplicación real de los conceptos de hipertexto.

Técnicamente, un documento de hipertexto consta de los elementos siguientes:

- los denominados nodos o secciones, que son las partes del texto que contiene información accesible para el usuario.
- los enlaces o hipervínculos, que se establecen entre los nodos y permiten que el usuario realice una lectura secuencial o no de los mismos.
- los anclajes, o puntos de activación de los enlaces.
- aparte de estos tres elementos pueden aparecer otros (por ejemplo un resumen o un índice del documento), en función de los cuales los hipertexto tienen un grado (grado 1, 2, etc.).

Su crecimiento fue imparable, de manera que enseguida las posibilidades del hipertexto se quedaron cortas. Los usuarios empezaron a demandar cada vez más la posibilidad de interactuar con las páginas, de manera que éstas pudiesen responder a las características particulares de cada uno de ellos. Es así como aparecen la creación de páginas Web de manera automática (o dinámica), que permiten al usuario modificar el contenido de las páginas que visualiza en base a los datos que introduce en formularios y a una serie de procesos implementados, siguiendo alguna de las tecnologías disponibles, que en general, e interactuando con bases de datos generan la respuesta pedida. Las tecnologías que posibilitaron esto han sido de lo más variadas:

1. CGIs (Common Gateway Interface): o Interface de Pasarela Común, es un mecanismo que permite establecer una comunicación entre un servidor Web y una aplicación, permitiendo de esta manera que la aplicación pueda interactuar con Internet. El fundamento de la programación CGI es muy simple y puede resumirse en lo siguiente: cuando un navegador solicita una petición a un servidor Web, éste genera un proceso (tantos como peticiones reciba) al que pasa los datos que el usuario envíe (si los hay), el proceso realiza su misión en función de estos datos (por ejemplo consulta a una base de datos) más otros denominados de ambiente y representados por lo que se conoce como variables CGI, el servidor Web captura la salida del CGI y la envía de vuelta al navegador del usuario en donde se muestra la salida correspondiente. El lenguaje más popular en el que se solían escribir este tipo de aplicaciones era el lenguaje C o el lenguaje Perl.
2. ASP y ASP .NET (Active Server Pages): Ambas tecnologías son la respuesta de Microsoft a la creación de páginas dinámicas. ASP apareció junto con el servidor Web de Microsoft Internet Information Server (o IIS), y ha pasado por cuatro generaciones diferentes que son: la versión 1.0 (sobre IIS 3.0), la versión 2.0 (sobre IIS 4.0), la versión 3.0 (sobre IIS 5.0) y la última generación que funciona sobre la plataforma .NET. La ventaja fundamental que tenemos con ASP y ASP.NET respecto a los CGIs es que cuando varios usuarios realizan peticiones sobre una misma página, no se crean n-instancias de la misma para dar respuesta a cada cliente (como sucede con los CGIs en donde hay n aplicaciones corriendo, una por cada cliente). Por otro lado, la programación de este tipo de aplicaciones, resulta mucho más fácil e intuitiva ya que siguen una sintaxis muy similar a la del HTML estándar, diferenciándose de éste en la existencia de etiquetas especiales con comportamientos específicos ya establecidos. ASP, tiene una limitación importante ya que solamente funciona en servidores Windows.
3. JSP (Java Server Pages): Es un desarrollo de la compañía Sun Microsystems y funciona de manera similar a las páginas ASP pero en este caso basándose fundamentalmente en el uso del lenguaje Java en donde reside su mayor potencial, ya que permite la integración con clases de este lenguaje de una manera natural. Gracias a esto se consigue una gran independencia entre la capa de presentación y la de negocio. Cuando se realiza una petición a una página JSP, ésta se compila y se transforma en un servlet que es el que se encargará de realizar el trabajo y generar la salida que el servidor Web enviará al cliente. JSP es multiplataforma y puede ejecutarse en multitud de servidores Web.
4. PHP (Hypertext Preprocessor Pages): Originalmente escrito en el lenguaje Perl, su autor original fue el programador Rasmus Lerdorf. Además de la creación de aplicaciones Web dinámicas PHP permite la programación en consola, al estilo del shell scripting, así como la creación de aplicaciones gráficas independientes del navegador mediante la combinación de PHP y GTK (GIMP Tool Kit). Entre las características más importantes de PHP radican la de ser un lenguaje multiplataforma, ser completamente gratuito, poseer grandes posibilidades de conexión a múltiples bases de datos y poderse expandir mediante la programación de sus propias extensiones.

Hoy día, la mayor parte de los sitios Web disponibles en Internet se encuentran desarrollados siguiendo alguno de los tres últimos lenguajes siendo la combinación JSP + Java, el paradigma más impuesto. La aparición de Java a principios de los años 90 supuso una gran revolución por su carácter multiplataforma, y la rápida expansión del uso de la tecnología JSP junto con el desarrollo por parte de Sun de la especificación J2EE, han sido sin duda determinantes para tener la posición dominante que ostentan.



## **6.2 Evolución de la Web Actual**

---

Analizando los contenidos y modelos en la historia de la Web podemos ver claramente que ha pasado por tres etapas o generaciones muy claras, a saber:

1. Generación o Generación de Contenido Estático
2. Generación o Generación de Contenido Dinámico o Interactivo
3. Generación o Generación de Contenido Colaborativo

La evolución continua de la tecnología hace un poco complicado establecer los límites exactos de fecha en los cuales podemos decir que la Web es de una generación u otra, ya que en la realidad lo que vemos es una superposición de cada una de estas generaciones según el sitio Web que examinemos. Sin embargo trataremos de hacer una aproximación.

### **Primera Generación o Generación de Contenido Estático**

La 1ª Generación de la Web o Generación de Contenido Estático, se corresponde con la Web que va desde que se crea por Tim Berners-Lee hasta la aparición de la 2ª Generación a mediados de los años 90.

Esta generación se caracteriza porque las páginas Web son completamente estáticas, es decir, el contenido que presentan es el que es y no permiten al usuario realizar ningún tipo de interacción con las mismas, salvo la manipulación propia del manejo del hipertexto y los hiperenlaces, esto es, saltar de una página Web a otra.

### **Segunda Generación o Generación de Contenido Dinámico**

La 2ª Generación de la Web o Generación de Contenido Dinámico, se corresponde con la Web en la que aparecen las primeras técnicas para permitir la inclusión de contenido dinámico, se puede decir que es la Web que hoy por hoy está más extendida y es la que se utiliza más comúnmente.

Esta generación se caracteriza porque las páginas Web son generadas por alguna de las tecnologías vistas para la generación de contenido dinámico (CGIs, ASP, ASP.NET, JSP o PHP, entre otros), permiten la interacción con el usuario en un nivel en donde éste, puede hacer preguntas y el sistema presenta las respuestas en función de los criterios introducidos en formularios. La experiencia del usuario queda limitada a él y a la aplicación que utiliza.

### **Tercera Generación o Generación de Contenido Colaborativo**

De la 3ª Generación de la Web, podemos decir que es el modelo que se está imponiendo poco a poco, y que haciendo uso de las capacidades adquiridas en la generación anterior, permite que la experiencia del usuario con la Web mejore espectacularmente.

En esta generación, las aplicaciones van más allá de la mera interacción entre aplicación-usuario-aplicación, ahora ellas son el mecanismo que permiten que se produzca interacción entre usuario-usuario llegando a crearse un entorno de contenido colaborativo, en donde, el usuario es un participante más en la creación del contenido que aparece en la Web. Ideas como los Wiki, los blogs, etc. han cambiado la forma en la que el usuario interactúa con la Web haciendo de esta un lugar mucho más rico e interesante para trabajar.

### 6.3 Problemas y dificultades en la Web Actual

---

No obstante, en este panorama existen ciertos puntos oscuros que sugieren la necesidad de acometer ciertas mejoras en el modelo. Por ejemplo, el enorme tamaño que ha alcanzado la Web, una de las claves de su éxito, hace que algunas tareas requieran un tiempo excesivo para una persona o resulten sencillamente inabarcables. Crear aplicaciones que realicen estas tareas en nuestro lugar es muy complicado, ya que es difícil reproducir y mantener en una máquina la capacidad de una persona para comprender los contenidos de la Web tal y como están codificados actualmente.

La eficacia de los buscadores actuales tiene también sus límites. Actualmente, se calcula que existen más de 3.000 millones de páginas en Internet, que crecen a un ritmo aproximado de 7 millones de páginas nuevas cada día. Una gran cantidad de dichas páginas quedan fuera del alcance de los buscadores.

Los buscadores tradicionales apenas abarcan el 1% de toda las páginas contenidas en la Red, por lo que su utilización puede resultar infructuosa cuando se pretende localizar una determinada información. Se estima que la información disponible en la Red es 500 veces superior a la accesible a través de dichos buscadores. Algunas importantes limitaciones de los buscadores son:

- Muchos buscadores sólo localizan las páginas que tienen una antigüedad limitada a un número determinado de semanas.
- Además, aunque localicen un determinado sitio, omiten la información contenida en documentos pertenecientes a ese sitio que existen en formato audiovisual, pdf o están incluidos en bases de datos.

**Ejemplo 1:** Búsquese en Google información sobre una palabra de significado ambiguo, como por ejemplo banco

**Deficiencia:** No existe referente semántico en las páginas, y es difícil deducir el significado de una palabra a través del contexto

**Ejemplo 2:** Necesito buscar un banco que me ofrezca un préstamo en unas ciertas condiciones, sabiendo que mi situación personal es X

**Deficiencia:** No existe un agente inteligente que sea capaz de deducir, de la información expuesta en la página del banco, cuáles serían las características del préstamo

Necesitamos:

- Un lenguaje común para la comunicación de la pregunta
- Que la información no sea ambigua
- Razonamiento automático
- No sólo demostración automática; se necesita razonamiento con conocimiento común

**Tabla 4: Ejemplos de deficiencias en la búsqueda dentro de la Web Actual**

Todos estos ejemplos son el síntoma de una causa común: la falta de capacidad de las representaciones en que se basa la Web actual para expresar significados. Los contenidos y servicios en la Web se presentan en formatos (p.e. HTML) e interfaces (p.e. formularios) comprensibles por personas, pero no por máquinas.

En estas condiciones es poco viable automatizar tareas mediante software en substitución del humano. Un programa puede llevar al usuario hasta lugares en la Web, generar, transportar,



transformar y ofrecer la información a las personas, pero la máquina sencillamente no sabe lo que esta información significa, y por tanto su capacidad de actuación autónoma es muy limitada. Esta misma limitación expresiva hace que la noción de semántica que manejan los buscadores Web se limite a palabras clave con pesos, pero planas e inconexas, lo que no permite reconocer ni solicitar significados más elaborados.



## 6.4 El nuevo paradigma: La Web Semántica

La Web semántica, tal y como ya hemos analizado antes en este TFC, propone superar las limitaciones de la Web actual mediante la introducción de descripciones explícitas del significado, la estructura interna y la estructura global de los contenidos y servicios disponibles en la WWW. En oposición a la semántica implícita, el crecimiento caótico de recursos, y la ausencia de una organización clara, características omnipresentes en la Web actual, la Web semántica propugna por clasificar, dotar de estructura y anotar los recursos con semántica explícita procesable por máquinas. En la actualidad la Web se asemeja a un grafo formado por nodos del mismo tipo, y arcos (hiperenlaces) igualmente indiferenciados. Por ejemplo, no se hace distinción entre la página personal de un profesor y el portal de una tienda on-line, como tampoco se distinguen explícitamente los enlaces a las asignaturas que imparte un profesor de los enlaces a sus publicaciones. Por el contrario en la Web semántica cada nodo (recurso) tiene un tipo (profesor, tienda, pintor, libro), y los arcos representan relaciones explícitamente diferenciadas (pintor – obra, profesor – departamento, libro – editorial).

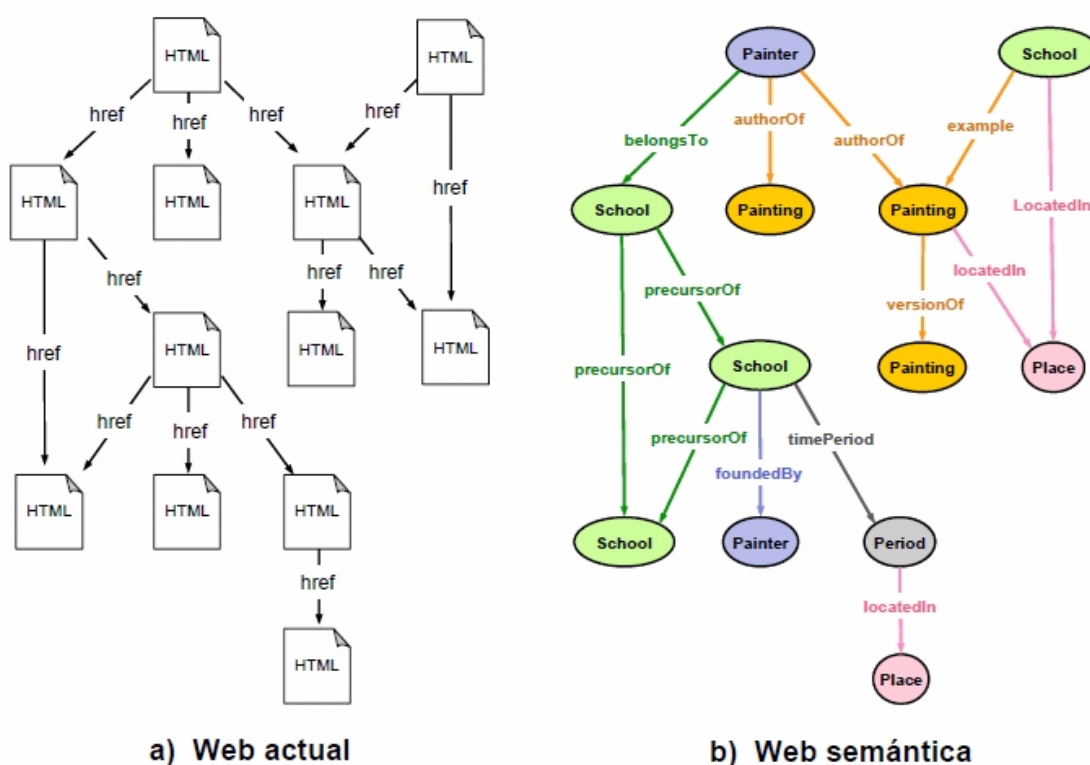


Fig. 17: La Web Actual frente a la Web Semántica

La Web semántica mantiene los principios que han hecho un éxito de la Web actual, como son los principios de descentralización, compartición, compatibilidad, máxima facilidad de acceso y contribución, o la apertura al crecimiento y uso no previstos de antemano. En este contexto un problema clave es alcanzar un entendimiento entre las partes que han de intervenir en la construcción y explotación de la Web: usuarios, desarrolladores y programas de muy diverso perfil. La Web semántica rescata la noción de ontología del campo de la Inteligencia Artificial como vehículo para cumplir este objetivo.

La idea es que la Web semántica esté formada (al menos en parte) por una red de nodos tipificados e interconectados mediante clases y relaciones definidas por una ontología compartida por sus distintos autores. Por ejemplo, una vez establecida una ontología sobre cuadros y pintura, un museo virtual puede organizar sus contenidos definiendo instancias de pintores, cuadros, etc., interrelacionándolas y





publicándolas en la Web semántica. La adopción de ontologías comunes es clave para que todos los que participen de la Web semántica, contribuyendo o consumiendo recursos, puedan trabajar de forma autónoma con la garantía de que las piezas encajen. Así por ejemplo varios museos podrían colaborar para dar lugar a un gran meta-museo que integre los contenidos de todos ellos. Un programa que navegue por una red como ésta puede reconocer las distintas unidades de información, obtener datos específicos o razonar sobre relaciones complejas. A partir de aquí sí podemos distinguir entre un cuadro pintado por un artista y un retrato de un artista.

Por último, la Web no solamente proporciona acceso a contenidos sino que también ofrece interacción y servicios (comprar un libro, reservar una plaza en un vuelo, hacer una transferencia bancaria, simular una hipoteca). Los servicios Web semánticos son una línea importante de la Web semántica, que propone describir no sólo información sino definir ontologías de funcionalidad y procedimientos para describir servicios Web: sus entradas y salidas, las condiciones necesarias para que se puedan ejecutar, los efectos que producen, o los pasos a seguir cuando se trata de un servicio compuesto. Estas descripciones procesables por máquinas permitirían automatizar el descubrimiento, la composición, y la ejecución de servicios, así como la comunicación entre unos y otros.



## 6.5 La transición hacia la Web Semántica

Una pregunta importante a responder es cómo encaja la Web Semántica con la actual, es decir

- cómo accederá el usuario a la Web Semántica
- cómo hacer la transición de la Web actual a la Web Semántica.

Para que la Web Semántica pueda realizarse es importante que guarde, al menos al principio, una compatibilidad con la tecnología actual. Es deseable por ejemplo mantener HTML (u otros lenguajes compatibles con los navegadores actuales) como vehículo de comunicación con el usuario. La asociación entre las instancias de la Web Semántica y el código HTML se puede establecer de distintas maneras (ver Fig. 18). Una consiste en conservar los documentos actuales, y crear las instancias asociadas anotando su correspondencia con los documentos (imagen izquierda en la figura 3). Esta posibilidad es la más viable cuando se parte de un gran volumen de material antiguo. Otra es generar dinámicamente páginas Web a partir de las ontologías y sus instancias (imagen derecha).

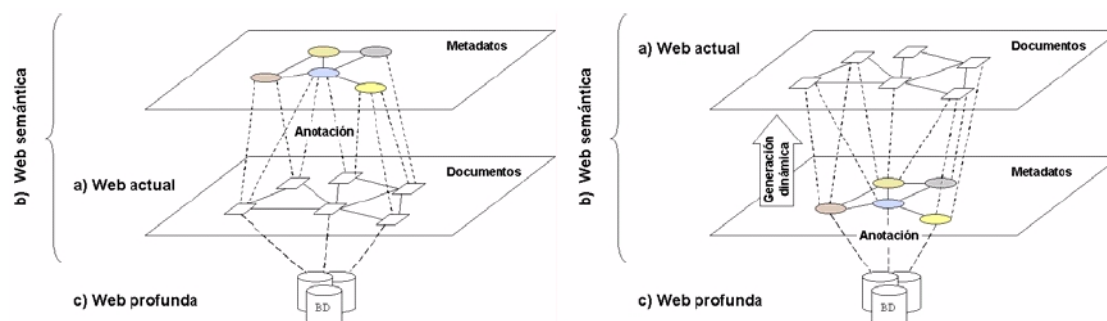


Fig. 18: De la Web actual a la Web Semántica

Esta última opción puede resultar factible cuando los documentos antiguos ya se estaban generando automáticamente a partir, por ejemplo, de una base de datos.

La transición de la Web actual a la Web Semántica puede implicar un coste altísimo si tenemos en cuenta el volumen de contenidos que ya forman parte de la Web. Crear y poblar ontologías supone un esfuerzo extra que puede resultar tedioso cuando se agregan nuevos contenidos, pero directamente prohibitivo por lo que respecta a integrar los miles de gigabytes de contenidos antiguos. Las estrategias más viables combinan una pequeña parte de trabajo manual con la automatización del resto del proceso. Las técnicas para la automatización incluyen, entre otras, el mapeo de la estructura de bases de datos a ontologías, el aprovechamiento, previa conversión, de los metadatos y estándares de clasificación presentes en la Web (y fuera de ella), y la extracción automática de metadatos a partir de texto y recursos multimedia.

Otra dificultad importante a la hora de realizar la Web Semántica en la práctica es la de consensuar ontologías en una comunidad por poco amplia que sea. Convergencia a una representación común es una labor más compleja de lo que puede parecer, ya que típicamente cada parte del sistema conlleva peculiaridades necesarias, y un punto de vista propio que a menudo necesitan incidir en la propia ontología. La representación del mundo no es neutra respecto al uso que se le va a dar: tanto un dietista como un biólogo tienen conocimiento sobre las plantas, pero su representación de esa materia es muy distinta, y probablemente no sería adecuado imponer la misma representación para ambas perspectivas. Las vías para salvar esta dificultad consisten en compartir ontologías para las áreas comunes en que puede tener lugar una interacción o intercambio de información entre las partes, y establecer formas de compatibilidad con las ontologías locales, mediante extensión y especialización de las ontologías genéricas, o por mapeo y exportación entre ontologías.



## 6.6 Conclusiones

Hoy en día todas las actividades y proyectos relacionados con la Web Semántica se encuentran en plena efervescencia. Se trata de un concepto nuevo, una nueva visión que, por otra parte, es completamente lógico dados los problemas de carencia de estructuración de la Web actual. Las diferencias respecto de los servicios actuales serían múltiples y, entre otras, se pueden citar las siguientes:

Característica	Web actual	Web Semántica
Lenguaje principal de uso	HTML	XML
Forma y estructura	Documentos no estructurados	Documentos estructurados siguiendo las pautas XML
Semántica usada	Implícita en el propio documento	Explícita mediante el uso profuso de metadatos
Relaciones entre contenido y forma	El código HTML fusiona la forma y el contenido	La forma y el contenido se mantienen separados
Público al que se dirige	Humanos	Humanos y computadores

**Tabla 5: Diferencias básicas entre la Web Actual y la Web Semántica**

Un aspecto que no se recoge en la tabla anterior y que también es fuerza importante es que la Web Semántica funciona por los principios de la comprensión parcial y la inferencia de la información, es decir, se pueden deducir nuevos conocimientos a partir de otros que se han entendido.

Aún queda mucho trabajo por hacer. Se necesita crear más y mejor tecnología e infraestructura, y más aún, desarrollar aplicaciones reales que pongan en práctica los principios de la Web semántica, que pueblen la Web con ontologías, y que hagan que la Web semántica adquiera la masa crítica imprescindible para hacerse realidad. En espera de que se alcance esta meta y al margen de ese debate, se han desarrollado ideas muy aprovechables a niveles específicos, y se han abierto nuevos campos para la innovación, suficientemente interesantes para motivar la investigación en esta área.

Algunas de estas aplicaciones de la Web Semántica son las que vamos a analizar más adelante en este TFC, intentando medir, si es posible, el alcance e impacto real dentro del uso de Internet.

## 7 Uso de la Web Semántica

---

Como ya se ha podido comprobar, la Web es una ubicación para el intercambio de información que ha sido diseñado por humanos y para su uso. Las páginas Web se crean por personas para ser entendidas por personas. No hay un estándar para mostrar la información, por lo que, los desarrolladores de páginas Web crean sus páginas dependiendo de los potenciales usuarios que van a visitarlas.

Los buscadores de Internet compilan la información demandada por los usuarios mediante palabras clave que aparecerán en el código HTML de las páginas Web dispersas en Internet o mediante la búsqueda en catálogos. Dichos resultados muchas veces no tienen nada que ver con lo que estamos buscando, aunque Internet sea la mayor recopilación de información elaborada en la historia del hombre.

¿Cómo se ha intentado solucionar el problema? Básicamente añadiendo meta-información a las páginas Web, primero usando el propio código HTML, incluyendo etiquetas <meta> que permiten definir atributos y asignarles un valor. No obstante, este sistema se reveló temprano como insuficiente, muy rudimentario, ya que no es estándar y no permite estructurar los atributos. Posteriormente se incorporó el uso del lenguaje XML, permitiendo a los creadores de sitios Web utilizar sus propias etiquetas y separar el contenido de su presentación. Sin embargo, se sigue manteniendo la imposibilidad de conocer automáticamente el significado de las etiquetas:

- cada grupo de trabajo modeliza su dominio de conocimiento a su manera pudiendo aparecer diferentes modelizaciones para un mismo dominio, y
- para intercambiar información las aplicaciones deben ponerse de acuerdo sobre un determinado vocabulario

Aunque Internet es indudable que ofrece enormes ventajas a la hora de buscar información, no proporciona una manera de encontrar dicha información de forma precisa ni tampoco permite realizar deducciones automáticamente con la información existente. Esto produce grandes dificultades para automatizar tareas, compartir información, organizarla, etc.

En esta parte del TFC se intentará mostrar el conjunto de soluciones prácticas elaboradas a partir de la arquitectura de la Web Semántica para solucionar estos problemas, e intentar visualizar las ventajas que puede aportar frente a la situación actual.

La Web, tal y cómo fue imaginada en sus inicios por su creador, Tim Berners-Lee, ya contemplaba la mayoría de ventajas que nos ofrecerá la Web Semántica. Solo las dificultades técnicas de la época para desarrollar tan revolucionarias ideas (1989) obligaron a concebir inicialmente la Web mediante una solución menos compleja como es la actual, más sencilla, usando documentos y enlaces de hipertexto.

La Web Semántica es una de las disciplinas actuales en la que mayor cantidad de profesionales realizan una tarea pujante en la confluencia de la Inteligencia Artificial y las tecnologías Web, proponiendo nuevas técnicas y paradigmas para la representación del conocimiento que faciliten la localización, compartición e integración de recursos a través de la Red. Estas nuevas técnicas se fundamentan en la introducción de conocimiento semántico explícito que describa y/o estructure la información y servicios disponibles. La Web Semántica mantiene los mismos principios de la Web actual:

- descentralización
- contenidos dinámicos
- compartición
- compatibilidad



- apertura al crecimiento y uso no previstos de antemano.

En este contexto un problema clave es alcanzar un entendimiento entre las partes: usuarios, desarrolladores y programas de muy diverso perfil.



## 7.1 Herramientas para la Web Semántica

---

En esta apartado se va a intentar realizar un seguimiento de diversas herramientas que se utilizan para el desarrollo de las aplicaciones propias de la Web Semántica, esto es, no se trata necesariamente de aplicaciones para el público general, sino más bien de las que permiten la construcción de las diversas partes de la Web Semántica.

### 7.1.1 Editores de Metadatos u Ontológicos

---

Son las herramientas que permiten la codificación de una determinada ontología en base a un determinado lenguaje. Permiten definir la estructura en base a la cual se clasificará la información realizada a través de anotaciones.

El uso de editores de ontologías facilita la tarea de definición de ontologías, la unión de diferentes ontologías y el desarrollo distribuido de ontologías.

**Apollo:** Aplicación amigable de modelado de conocimiento. El modelado está basado en torno a los principios básicos tales como clases, instancias, funciones, relaciones, etc. La interface de usuario tiene una arquitectura abierta y está escrito en lenguaje de programación JAVA.

**LinkFactory:** Se trata de una herramienta utilizada para construir completos sistemas de terminología corporativa capaz de extraer valor significativo de gran cantidad de datos no estructurados almacenados en bases de datos de contenido corporativo.

**OILED:** Es un editor de ontologías que permite al usuario construir ontologías utilizando DAML+OIL

**OntoEdit Free and Professional versions:** Permite crear y gestionar ontologías. Confía en los estándares del W3C y ofrece muchas interfaces exportables a la mayor parte de lenguajes de representación de ontologías. Esta herramienta permite crear, navegar y modificar ontologías.

**Ontolingua Server:** Provee de un entorno de colaboración distribuido para navegar, crear, editar, modificar y utilizar ontologías.

**OntoSaurus:** Es un navegador web para las bases de conocimiento de LOOM. Proporciona una interface gráfica hyperenlazada hacia varias de las bases de conocimiento.

**OpenKnoME:** Es la piedra angular de la aplicación utilizada por los motores de conocimiento topThing. Es un sistema de gestión del conocimiento y un motor de ontologías. Desde el 2001 el código fuente está abierto para la comunidad académica y la clínica sin ánimo de lucro.

**Protégé-2000:** Editor de ontologías y editor de bases de conocimiento. Es también de código abierto, herramienta JAVA que proporciona una arquitectura extensible para la creación de aplicaciones de bases de conocimiento customizadas.

**SymOntoX:** Es un software que almacena y gestiona un dominio de ontología.

**WebODE:** Herramienta para modelar el conocimiento utilizando ontologías. Facilita la máxima flexibilidad e interoperabilidad con otras aplicaciones de negocios necesarias para las empresas de hoy en día.



**WebOnto:** JAVA applet con un servidor web customizado que permite a los usuarios navegar y editar modelos de conocimiento sobre la web.

**Tabla 6: Herramientas de creación de ontologías**

### 7.1.2 Herramientas de anotación externas

Las herramientas de anotación externas son aquellas que permiten asociar meta información, del tipo asociado a la Web Semántica, a páginas Web que están diseñadas según el modelo convencional de la Web actual y que ya están presentes y operativas en la Web. Esta meta información adicional que se incluye no se almacena dentro de la misma página sino que se almacena de forma externa, utilizando un repositorio cuyo fin específico es mantener las anotaciones. Estos repositorios suelen ser bases de datos RDF, implementadas generalmente sobre sistemas gestores de base de datos relacionales. Esta meta información también puede estar basada, o no, en una ontología.

**COHSE (The Conceptual Open Hypermedia Project):** Proyecto de investigación sobre métodos que mejoren significativamente la calidad, consistencia y la amplitud de documentos Web enlazados mientras se recuperan (cuando los lectores navegan sobre los documentos) y al mismo tiempo que se crean (cuando los autores crean los documentos). COHSE utiliza tres tecnologías: un servicio de razonamiento de ontologías que se utiliza para representar un sofisticado modelo conceptual de términos documentales y sus relaciones, un servicio abierto de enlaces hipermedia basado en Web, que sea escalable, y la integración de estos dos para poder enlazar documentos vía meta datos describiendo sus contenidos. Para su empleo, el usuario final sólo necesita tener instalado el navegador Web Mozilla, junto con cierto software de COHSE para este navegador. Con el navegador, el usuario puede realizar anotaciones y ver las notaciones realizadas por el resto de usuarios.

**Annotea:** Proyecto que pretende mejorar el ambiente colaborativo de la W3C (el "Consortio de la World Wide Web") a través del uso de anotaciones compartidas; se entiende que una anotación puede ser un comentario, una nota, una explicación o cualquier texto que se pueda adjuntar a un documento Web externamente, es decir, sin necesidad de tocar el documento mismo. El usuario recibe las notas de un servidor o varios adjuntas a un documento junto con éste y puede ver qué piensan sus colegas del documento. En este proyecto se ha desarrollado un navegador Web propio llamado Amaya para poder ver y crear las anotaciones. Para su empleo, el usuario final sólo necesita tener instalado el navegador Web Amaya o el de Mozilla. En el caso de emplear Mozilla, necesita instalar el software de Annozilla para este navegador. Con estos navegadores, el usuario puede realizar anotaciones y ver las notaciones realizadas por el resto de usuarios.

**Annozilla:** Proyecto diseñado para ver y crear anotaciones asociadas a una página Web mediante el navegador Mozilla, como define el proyecto Annotea del W3C.

**Yawas (Yet Another Web Annotation System):** YAWAS es una herramienta que permite a los usuarios realizar anotaciones para expresar sus opiniones y personalizar los documentos a medida que analizan la información en páginas Web. Se trata de un editor de anotaciones creado por expertos que consideran que las anotaciones almacenadas en servidores, limitan la extensión de la tecnología de anotaciones en la Web. La propuesta es codificar las anotaciones en URLs extendidas, que pueden estar disponibles fácilmente y ser insertadas en documentos. Se trata de que el usuario disponga de una herramienta de trabajo similar a un libro de favoritos pero personalizado y adaptado a su necesidad. Para su empleo, el usuario final sólo necesita tener instalado el navegador Web Internet Explorer o el de Netscape, junto con cierto software de Yawas para estos navegadores. Con el navegador, el usuario puede realizar anotaciones y ver sólo las notaciones realizadas por él.

**Annotation System:** Sistema basado en tres arquitecturas vinculadas: un cliente que puede realizar / buscar anotaciones, un servidor de aplicaciones que gestiona todas las peticiones del cliente, un servidor de base de datos para almacenar las anotaciones realizadas por los usuarios. Para su empleo, el usuario final sólo necesita tener instalado el navegador Web Internet Explorer, junto con cierto software de Annotation System para este navegador. Con el navegador, el usuario puede realizar anotaciones y ver las notaciones realizadas por el resto de usuarios.

**Trellis Web:** Entorno interactivo que permite a los usuarios añadir sus observaciones, puntos de vista y conclusiones como la información que analizan realizando anotaciones semánticas a los documentos y a otros recursos on-line. En realidad es una forma de adquirir conocimiento donde el usuario puede añadir nuevo conocimiento al sistema basándose en su propia experiencia a medida que analiza la información. Para su empleo, el usuario final sólo necesita tener instalado un navegador Web capaz de soportar "frames". Con el navegador, el usuario puede realizar anotaciones y ver las notaciones realizadas por el resto de usuarios.

**Tabla 7: Herramientas de anotación externa**

### 7.1.3 Herramientas de autor

---

Las herramientas de autor necesitan para su utilización la existencia de un ontología definida previamente (mejor de forma específica). Bajo esta condición, estas herramientas permiten la incorporación de información estructurada, los meta datos, en la propia página web. Para ello se suelen utilizar los lenguajes de marcado promovidos por el W3C, a saber, XML o RDF. Para la definición de las ontologías se suele utilizar DAML, aunque actualmente el W3C está promoviendo la utilización de OWL.

Para poder realizar y ver anotaciones con las herramientas de anotación de autor, el usuario final (estas suelen ser herramientas para usuarios finales) necesita instalar en su ordenador una aplicación concreta para cada uno de los tipos de herramientas.

Las herramientas de anotación de autor están más orientadas a la Web semántica, y para que sean de utilidad para el usuario final requieren el desarrollo adicional de agentes inteligentes que entiendan las ontologías, sean capaces de realizar búsquedas inteligentes y de extraer información.

**MnM:** MnM es una herramienta de anotación basada en ontologías que permite anotar páginas Web con contenidos semánticos de forma automática y semi-automática. MnM integra un navegador Web con un editor de ontologías y proporciona unas APIs abiertas para enlazar MnM con servidores de ontologías y para integrar MnM con herramientas de extracción de información. MnM trabaja con diferentes lenguajes de ontologías como RDF, DAML+OIL y OCML.

**OntoMat-Annotizer:** Herramienta de anotación de páginas Web interactiva y amigable. Ayuda al usuario en la tarea de crear y mantener ontologías basadas en DAML+OIL, por ejemplo, para crear instancias, atributos y relaciones. Incluye un navegador de ontologías para explorar la ontología y las instancias y un navegador HTML que visualiza las partes anotadas del texto.

**SHOE Knowledge Annotator:** Herramienta que permite realizar anotaciones en documentos Web sin tener que preocuparse de los códigos HTML pues añade las etiquetas necesarias automáticamente. Las etiquetas están divididas en dos categorías: para la construcción de ontologías y para anotar documentos Web para suscribirlos a una o más ontologías, declarar entidades de datos y realizar afirmaciones sobre las entidades según las normas establecidas por las ontologías.





<b>SMORE:</b> Herramienta que permite a los usuarios etiquetar sus documentos en RDF utilizando ontologías existentes en Internet asociadas a los elementos y términos específicos.
<b>Melita:</b> Melita es una herramienta de anotación de texto semi-automática basada en ontologías que tiene integrada una máquina de extracción de información (Amilcare). Implementa una metodología con el propósito de gestionar para los usuarios todo el proceso de anotación. Algunos de los pasos del proceso, que hasta ahora se realizan manualmente, el sistema los puede automatizar y gestionar fácilmente. Las principales capacidades de Melita se pueden resumir en cuatro grupos, la tarea de gestión, la extracción, el aprendizaje, y el etiquetado de la Información de forma autónoma.
<b>OntoAnnotate:</b> Las ontologías funcionan como base conceptual para el intercambio de conocimientos con diferentes propósitos. Con OntoAnnotate se puede mediante la navegación habitual en la Web y en documentos internos ampliar su base de conocimientos, al instante. La herramienta permite aportar el conocimiento de cada uno. El usuario puede aprehender hechos individuales que corren directamente en la base de conocimientos, con ellos sus asociaciones, relaciones, sus contextos en las informaciones serán capaces de ser transmitidas, interpretadas y valoradas para sistemas capaces de procesar como la máquina de inferencias Ontobroker. OntoAnnotate puede ser utilizado junto a las anotaciones de los documentos para proveer de contenidos semántico, los cuales y de forma compatible con los estándares son directamente accesibles e interpretables como contenidos en la Web Semántica. Permite a cada usuario crear documentos con contenidos semánticos capaces de ser entendidos por las máquinas.
<b>GATE (General Architecture for Text Engineering):</b> es una infraestructura de componentes de software reutilizables para dar soporte al procesamiento del lenguaje humano.

**Tabla 8: Herramientas de autor**

## 7.1.4 Lenguajes de codificación de Ontologías

Se puede definir una ontología como un vocabulario para describir las cosas que existen. Son utilizadas para estudiar la existencia de todo tipo de entidades, abstractas y concretas, que componen nuestro universo de existencia.

De forma un tanto más técnica conviene definir una ontología como una especificación formal y explícita de una conceptualización compartida, la cual tiene por objeto principal ser comprensible por un ordenador.

Como tales, las ontologías son colecciones de entidades a través de las que se definen las relaciones entre conceptos y especifican reglas lógicas para realizar razonamientos sobre estas relaciones. Los computadores entenderán el significado de la semántica asociada a una página Web siguiendo los links que se especificarán en las ontologías. Las ontologías también contienen información muy importante sobre equivalencias semánticas.

Se llaman Lenguajes de codificación de ontologías a los lenguajes utilizados para realizar la codificación de dichas ontologías. Estos lenguajes permiten representar formalmente las ontologías y posibilitan su utilización por parte de los ordenadores.

Saber los diferentes lenguajes que existen, cual es su grado de implantación, cuales son los ámbitos de aplicación de cada uno de ellos, etc., es una tarea que en el actual grado de desarrollo expansivo de esta tecnología resulta casi inabordable. La información siguiente, simplificada, resulta útil para estudiar compatibilidades entre herramientas.

<p><b>DAML+OIL:</b> El DAML (DARPA Agent Markup Language) es un lenguaje para modelar ontologías creado como una extensión de RDF. El DAML provee un rico juego de construcciones con el que crear ontologías y aumentar la información para que sea legible y comprensible por las máquinas. En diciembre de 2000 el DAML pasó a decirse DAML+OIL, debido a la revisión de las especificaciones del lenguaje.</p>
<p><b>RDF (Resource Description Framework):</b> Se trata de un modelo para definir relaciones semánticas entre distintas URIs. RDF está basado en la sintaxis XML, y permite describir semánticamente una URI asociándole un conjunto de propiedades y valores. Los modelos RDF se construyen como grafos dirigidos especificando "triples" (URI, propiedad, valor). Los metadatos especificados con RDF son "comprensibles por las máquinas, y por tanto, procesables de forma automática.</p>
<p><b>OWL (Web Ontology Language):</b> Intenta proveer de un lenguaje que pueda ser utilizado para describir clases y relaciones entre ellas que sean inherentes a documentos web y aplicaciones.</p>
<p><b>FLogic (Frame Logic):</b> Lenguaje orientado a objetos y sistema de base de datos deductivo.</p>
<p><b>OCML (Operational Conceptual Modelling Language):</b> El lenguaje de modelado OCML sostiene la construcción de modelos de conocimiento por significados de varios tipos de construcciones. Permite la especificación y operación de funciones, relaciones, clases, instancias y reglas. También incluye mecanismos para definir ontologías y métodos para resolver problemas.</p>
<p><b>SHOE (Simple HTML Ontology Extensions):</b> Se trata de un lenguaje de representación del conocimiento para la Web compatible con XML. Permite a los autores de las páginas anotar sus documentos web.</p>
<p><b>CLIPS (C Language Integrated Production System):</b> Entre otras características, permite la representación del conocimiento. CLIPS provee de una herramienta para gestionar una extensa variedad de conocimiento con el soporte de tres paradigmas de programación diferentes: basado en reglas, orientado a objetos y procesal.</p>
<p><b>LOOM:</b> Es un lenguaje y un entorno para construir aplicaciones inteligentes. El corazón de LOOM es un sistema de representación del conocimiento que es utilizado para proveer un soporte deductivo para la porción declarativa del lenguaje LOOM. El conocimiento declarativo en LOOM consiste en definiciones, reglas, preguntas y herramientas por defecto.</p>

**Tabla 9: Lenguajes de codificación de ontologías**



## 8 Análisis de las herramientas para la construcción de la Web Semántica

### 8.1 Metodología para análisis de herramientas

#### 8.1.1 Esquema de tabla para el análisis las herramientas de anotación externas y herramientas de anotación de autor

<b>Nombre del editor de anotaciones</b>
<b>Descripción</b>
<b>Desarrolladores (organización a la que pertenece)</b>
<b>Información complementaria (dónde se puede encontrar)</b>
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
<b>Se pueden crear nuevas subclases fácilmente</b>
<b>Se puede crear más de una anotación en una misma página Web</b>
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>

<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
<b>Tecnologías de código abierto o de código propietario</b>
<b>Compatibilidad con otros sistemas</b>
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
<b>Proceso para realizar la anotación</b>
<b>Como se buscan las anotaciones</b>
<b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b>
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
<b>Si permite abrir debates (discusion threts)</b>
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
<b>Utilidad / facilidad de uso</b>
<b>Lenguajes y posibles formatos de importación y exportación</b>

Tabla 10: Esquema 1, anotaciones externas y de autor



### 8.1.2 Esquema de tabla para el análisis las herramientas de creación de ontologías

<b>Nombre de la herramienta de creación de ontologías</b>
<b>Versión y fecha de lanzamiento de la versión</b>
<b>Licencia</b>
<b>Desarrolladores (organización a la que pertenece)</b>
<b>Información complementaria (dónde se puede encontrar)</b>
<b>Descripción</b>
<b>Arquitectura software</b>
<b>Extensibilidad</b>
<b>Gestión de backups</b>
<b>Acceso a la base del conocimiento vía Web a través de un navegador</b>
<b>Almacenamiento de la ontología</b>
<b>Almacenamiento en base de datos</b>
<b>Tamaño de las ontologías o conocimiento base creado por la herramienta</b>

<b>Características del modelado</b>
<b>Lenguaje base usado para codificar la ontología</b>
<b>Plataformas donde puede ejecutarse</b>
<b>Formatos de importación/exportación de la ontología</b>
<b>Interfaz gráfica</b>
<b>Comprobación automática de consistencia (sintáctica, referencial y/o lógica) de la ontología</b>
<b>Soporte multi-usuario es decir, desarrollo concurrente de la ontología</b>
<b>Soporte para la comparación y fusión de ontologías creadas por aplicaciones independientes</b>
<b>Soporte léxico (capacidad de referenciación léxica de los elementos de la ontología (por ejemplo sinónimos) y el procesado léxico de contenido(búsqueda/filtrado de términos de la ontología)</b>
<b>Información adicional</b>
<b>Contacto</b>
<b>Compatibilidad con otros sistemas</b>
<b>Utilidad / facilidad de uso</b>

**Tabla 11: Esquema 2, creación de ontologías**



## 8.2 Análisis individualizado de las principales herramientas

### 8.2.1 Cohse

<b>Nombre del editor de anotaciones</b>
COHSE (Conceptual Open Hypermedia Services Environment)
<b>Descripción</b>
<p>COHSE es un proyecto de investigación sobre métodos que mejoren significativamente la calidad, consistencia y la amplitud de documentos Web enlazados mientras se recuperan (cuando los lectores navegan sobre los documentos) y al mismo tiempo que se crean (cuando los autores crean los documentos). COHSE utiliza tres tecnologías principales:</p> <ul style="list-style-type: none"><li>• Un servicio de razonamiento de ontologías que se utiliza para representar un sofisticado modelo conceptual de términos documentales y sus relaciones</li><li>• Un servicio abierto de enlaces hypermedia basado en Web, que sea escalable</li><li>• La integración de estos dos servicios para poder enlazar documentos vía metadata que describan sus contenidos</li></ul>
<b>Desarrolladores (organización a la que pertenece)</b>
<p>Las instituciones encargadas del proyecto COHSE son:</p> <ul style="list-style-type: none"><li>• IAM (Intelligence, Agents and Media), Universidad de Southampton del Reino Unido. Grupo dedicado al diseño y a la creación de aplicaciones utilizando información compleja y relacionada con tareas de procesamiento del conocimiento</li><li>• IMG (Information Management Group), Universidad de Manchester del Reino Unido. Realiza tareas de investigación en el diseño, desarrollo y uso de sistemas de gestión de datos y conocimiento</li></ul>
<b>Información complementaria (dónde se puede encontrar)</b>
En la dirección <a href="http://cohse.semanticweb.org/">http://cohse.semanticweb.org/</a> hay información disponible sobre esta herramienta
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
COHSE funciona con el navegador Mozilla/Firefox y con Internet Explorer superior o igual a 5.X.
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones se almacenan en una base de datos MySQL y el servicio de anotación añade y borra anotaciones de la misma
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Cada anotación realizada en una página Web está relacionada con un concepto de una ontología

<b>Se pueden crear nuevas subclases fácilmente</b>
COHSE no permite editar o crear nuevas ontologías. Existe un applet Java denominado Concept Browser que se utiliza para crear nuevas estructuras complejas dentro de una ontología ya creada previamente
<b>Se puede crear más de una anotación en una misma página Web</b>
Es posible crear más de una anotación en una misma página Web incluso se pueden hacer anotaciones con diferentes ontologías en una misma página Web
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
Si. El primer paso antes de realizar la anotación es elegir la ontología con la que se va a realizar dicha anotación y el concepto al que se refiere
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
COHSE representa un modelo conceptual y crea sus relaciones utilizando DAML+OIL como lenguaje de representación. COHSE utiliza el editor de ontologías OIEd aunque las ontologías se pueden construir con cualquier constructor de ontologías que soporte DAML.
<b>Tecnologías de código abierto o de código propietario</b>
El software se puede emplear libremente sólo para fines no comerciales. El software está desarrollado con Java, XML y Perl. Emplea la base de datos MySQL. Se puede instalar el COHSE Annotator tanto en el explorador de Internet Explorer como en Mozilla.  En las direcciones <a href="http://cohse.semanticweb.org/">http://cohse.semanticweb.org/</a> y <a href="http://www.ecs.soton.ac.uk/~tmb/cohse/annotator/">http://www.ecs.soton.ac.uk/~tmb/cohse/annotator/</a> hay información disponible sobre esta herramienta. Para instalar la herramienta se requiere Java 1.4.0 o superior
<b>Compatibilidad con otros sistemas</b>
No
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
Las anotaciones son compartidas por todos los usuarios ya que se almacenan en una base de datos y el servicio de anotaciones es el encargado de retirar o añadir las anotaciones de la misma
<b>Proceso para realizar la anotación</b>
Mozilla Annotator se utiliza para anotar documentos Web con conceptos de una ontología. Se puede anotar una sentencia de un documento o incluso se puede llegar a anotar toda la página entera. El proceso de anotación es el siguiente:  <ol style="list-style-type: none"><li>1. Se carga una página Web en el navegador Mozilla</li><li>2. Se selecciona el área a anotar en la página Web</li><li>3. Se elige la ontología y el concepto</li><li>4. Se pulsa sobre el botón "Create"</li><li>5. La anotación se envía al servicio de anotación donde se almacena</li></ol>





<b>Como se buscan las anotaciones</b>
El primer paso para realizar la búsqueda de las anotaciones es cargar la página Web en la que se desean ver dichas anotaciones. Una vez cargada la página mediante la barra de tareas se elige la ontología y en el navegador procede a recargar la página. Una vez realizado este último paso aparecen las anotaciones correspondientes de la ontología seleccionada
<b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b>
Al lado del texto o sentencia anotada, aparece un cuadrado pequeño que tiene en su interior la letra "A" de Annotation, y al pasar el ratón se resalta el texto anotado. Además de anotaciones, junto al texto del documento puede aparecer un cuadro pequeño que tiene en su interior la letra "L" de Link. Estos son enlaces creados automáticamente por COHSE cuando el texto coincide con instancias conceptos de la ontología en curso. COHSE proporciona opciones de configuración para poder filtrar los tipos de anotaciones o enlaces que el usuario desea que se muestren en la página Web
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
No
<b>Si permite abrir debates (discusion threts)</b>
No
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
Al lado de la anotación realizada aparece un cuadrado pequeño que funciona como enlace. Este enlace conduce a la descripción del concepto de la anotación. En ocasiones en esa misma descripción pueden aparecer direcciones de páginas Web relacionadas con el texto anotado. Se pueden ver todas las anotaciones realizadas por diferentes ontologías o sólo con la que en ese momento esta en curso
<b>Utilidad / facilidad de uso</b>
El proceso de anotación es muy intuitivo. Todo el funcionamiento se realiza a través de una barra de herramientas que se sitúa en la parte izquierda del navegador Mozilla y en la barra de herramientas habitual del navegador Internet Explorer
<b>Lenguajes y posibles formatos de importación y exportación</b>
Las ontologías están definidas con DAML+OIL. Las anotaciones se pueden guardar en formato RDF

**Tabla 12: Características de Cohse**

## 8.2.2 Amaya

---

<b>Nombre del editor de anotaciones</b>
Amaya
<b>Descripción</b>
Amaya es un editor de anotaciones basado en el proyecto Annotea, que pretende mejorar el ambiente colaborativo de la W3C (el "Consortio de la World Wide Web") a través del uso de anotaciones compartidas
<b>Desarrolladores (organización a la que pertenece)</b>
El desarrollo de la aplicación está siendo llevado a cabo por el W3C, el proyecto WAM (Web, Adaptation and Multimedia) y el proyecto INRIA. Entre los desarrolladores destacan: Irène Vatton (Jefe de proyecto, INRIA), Laurent Carcone (W3C), Paul Cheyrou-Lagreze (INRIA), Vincent Quint (INRIA), José Kahan (W3C)
<b>Información complementaria (dónde se puede encontrar)</b>
La dirección del proyecto amaya es: <a href="http://www.w3c.org/Amaya/">http://www.w3c.org/Amaya/</a> . Desde aquí podremos seguir el desarrollo de la aplicación y descargar la última versión
<b>Con que tipo de navegador se pueden visualizar las anotaciones)</b>
Las anotaciones, al estar basadas en el proyecto Annotea, pueden verse desde otros navegadores que también estén basados en el mismo, como por ejemplo Mozilla (a través del plug-in Annozilla) o Internet Explorer (a través del plug-in Snufkin)
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones se almacenan en bases de datos RDF genéricas, que son accesibles a través de un servidor HTTP. Cada anotación dispone de su propia URL
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Si, el programa nos permite definir que tipo de anotación deseamos realizar: comment, advice, example, seeAlso...
<b>Se pueden crear nuevas subclases fácilmente</b>
No, las subclases están predefinidas
<b>Se puede crear más de una anotación en una misma página Web</b>
Si, se pueden crear todas las anotaciones que se deseen
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
No, esta opción no está contemplada



<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
Se usa un esquema RDF para describir las anotaciones como metadatos y el protocolo Xpointer para indicar las posiciones de las anotaciones en el documento
<b>Tecnologías de código abierto o de código propietario</b>
En el diseño de Annotea era fundamental construir una infraestructura basada en RDF genérico, donde las anotaciones fueran una posible encarnación de esa infraestructura
<b>Compatibilidad con otros sistemas</b>
El uso de tecnología abierta permite la interoperabilidad con otros sistemas de anotación al tiempo que permiten la extensibilidad de los datos que el sistema puede llevar.
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
Las anotaciones son compartidas por todos los usuarios
<b>Proceso para realizar la anotación</b>
Para realizar una anotación a la página pulsamos en la opción “annotations/anótate document”. Se nos abrirá una página que podremos editar. Al final deberemos pulsar la opción “annotations/post to server”. Si lo que deseamos es realizar una anotación únicamente a una parte de la página, deberemos seleccionar la parte que deseamos anotar y pulsar en la opción “annotations/annotate selection”. A partir de aquí los pasos son idénticos al punto anterior
<b>Como se buscan las anotaciones</b>
Las anotaciones aparecen intercaladas en el cuerpo de la página
<b>Como se visualiza la anotación cuando el usuario navega por una página web que tiene anotaciones realizadas por otros usuarios</b>
Las anotaciones aparecen entremezcladas con el código y se señalan con la imagen de un lápiz
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
En principio no
<b>Si permite abrir debates (discusion threads)</b>
No
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
Se utiliza un icono que representa un lápiz
<b>Utilidad / facilidad de uso</b>
No es demasiado complicado aprender su funcionamiento si se tienen unos conocimientos básicos sobre anotaciones

**Lenguajes y posibles formatos de importación y exportación**

Las anotaciones, al estar basadas en el proyecto Annotea, pueden verse desde otros navegadores que también estén basados en el mismo, como por ejemplo Mozilla (a través del plug-in Annozilla) o Internet Explorer (a través del plug-in Snufkin) pero no contempla ningún tipo de importación o exportación hacia otros lenguajes

**Tabla 13: Características de Amaya**



### 8.2.3 Annozilla

<b>Nombre del editor de anotaciones</b>
Annozilla
<b>Descripción</b>
Anozilla es un editor de anotaciones basado en el proyecto Annotea, que pretende mejorar el ambiente colaborativo de la W3C (el "Consortio de la World Wide Web") a través del uso de anotaciones compartidas
<b>Desarrolladores (organización a la que pertenece)</b>
El desarrollo se está llevando a través de mozdev.org. El director del proyecto es Matthew Wilson y de entre sus colaboradores destacan Doug Daniels y Jeffrey Phillips.
<b>Información complementaria (dónde se puede encontrar)</b>
La dirección del proyecto Annozilla es: <a href="http://Annozilla.mozdev.org">http://Annozilla.mozdev.org</a> . Desde aquí podremos informarnos de las últimas novedades sobre el proyecto así como descargar el programa
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
Annozilla está implementado para funcionar únicamente en Mozilla, no obstante, al estar basado en Annotea, las anotaciones realizadas podremos verlas desde otros navegadores que también estén basados en Annotea, como por ejemplo el navegador Amaya o el plug-in Snufkin para Internet Explorer
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones se almacenan en bases de datos RDF genéricas, que son accesibles a través de un servidor HTTP. Cada anotación dispone de su propia URL.
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Si, el programa nos permite definir que tipo de anotación deseamos realizar: comment, advice, example, seeAlso...
<b>Se pueden crear nuevas subclases fácilmente</b>
No, las subclases están predefinidas
<b>Se puede crear más de una anotación en una misma página web</b>
Si, se pueden crear todas las anotaciones que se deseen
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
No, esta opción no está contemplada
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>

<p>Se usa un esquema RDF para describir las anotaciones como metadatos y el protocolo Xpointer para indicar las posiciones de las anotaciones en el documento</p>
<b>Tecnologías de código abierto o de código propietario</b>
<p>En el diseño de Annotea era fundamental construir una infraestructura basada en RDF genérico, donde las anotaciones fueran una posible encarnación de esa infraestructura</p>
<b>Compatibilidad con otros sistemas</b>
<p>El uso de tecnología abierta permite la interoperabilidad con otros sistemas de anotación al tiempo que permiten la extensibilidad de los datos que el sistema puede llevar. A parte de Annotilla, el proyecto annota se integra por defecto en el navegador amaya y en Internet explorer a través del plug-in snufkin</p>
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
<p>Las anotaciones son compartidas por todos los usuarios</p>
<b>Proceso para realizar la anotación</b>
<p>En la pestaña "Annotilla" de la barra lateral seleccionamos la opción "create" y rellenamos los campos requeridos</p>
<b>Como se buscan las anotaciones</b>
<p>En la misma pestaña seleccionamos la opción "view" y en esta misma pestaña nos aparecerá un listado de las anotaciones disponibles</p>
<b>Como se visualiza la anotación cuando el usuario navega por una página web que tiene anotaciones realizadas por otros usuarios</b>
<p>Todas las anotaciones se visualizan en la pestaña "Annotilla", por lo que no se entremezclan con el código</p>
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
<p>En principio no</p>
<b>Si permite abrir debates (discussion threads)</b>
<p>No</p>
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
<p>Se utiliza un icono que representa un lápiz</p>
<b>Utilidad / facilidad de uso</b>
<p>No es demasiado complicado aprender su funcionamiento si se tienen unos conocimientos básicos sobre anotaciones</p>
<b>Lenguajes y posibles formatos de importación y exportación</b>



Annotea trabaja con el formato RDF que, al ser standard, permite importar y exportar datos de forma sencilla

**Tabla 14: Características de Annotea**

## 8.2.4 Yawas

---

<b>Nombre del editor de anotaciones</b>
YAWAS (Yet Another Web Annotation System)
<b>Descripción</b>
YAWAS es una herramienta que permite a los usuarios realizar anotaciones para expresar sus opiniones y personalizar los documentos a medida que analizan la información en páginas Web (y pronto, también en sus correos de Outlook). Se trata de un editor de anotaciones creado por expertos que consideran que las anotaciones almacenadas en servidores, limitan la extensión de la tecnología de anotaciones en la Web. La propuesta es codificar las anotaciones en URLs extendidas, que pueden estar disponibles fácilmente y ser insertadas en documentos. Se trata de que el usuario disponga de una herramienta de trabajo similar a un libro de favoritos pero personalizado y adaptado a su necesidad. En realidad existen 3 versiones, y en la última sólo se permite resaltar textos y no anotar, pues el autor considera que esta acción es suficiente para organizar el conocimiento
<b>Desarrolladores (organización a la que pertenecen)</b>
Este sistema de anotación lo desarrolló Laurent Denoue en la Universidad de Saboya en Chambéry, Francia
<b>Información complementaria (dónde se puede encontrar)</b>
La dirección del proyecto YAWAS es: <a href="http://www.univ-savoie.fr/labos/syscom/Laurent.Denoue/">http://www.univ-savoie.fr/labos/syscom/Laurent.Denoue/</a> . Desde aquí podremos obtener más información sobre el proyecto
<b>Con qué tipo de navegador se pueden visualizar las anotaciones</b>
Las anotaciones se visualizan con Microsoft Internet Explorer, y Netscape Navigator en plataformas Windows
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones se almacenan en el PC local del autor
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Sí, el programa nos permite definir que tipo de anotación deseamos realizar, pero no son campos obligatorios a cumplimentar. Esta opción está sólo activa en versiones antiguas y su correcto funcionamiento no se garantiza
<b>Se pueden crear nuevas subclases fácilmente</b>
En principio no
<b>Se puede crear más de una anotación en una misma página Web</b>
Sí, se pueden crear todas las anotaciones que se deseen, basta seleccionar los textos a





anotar
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
No se enlaza con servicios de Ontología
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
Las fuentes específicas se identifican mediante URLs extendidas y para apuntar a la parte del texto a resaltar y/o anotar se usa el lenguaje XML Pointer (XPointer)
<b>Tecnologías de código abierto o de código propietario</b>
El sistema está continuamente actualizándose, y su código fuente está disponible libremente en: <a href="http://www.univsavoie.fr/labos/syscom/Laurent.Denoue">http://www.univsavoie.fr/labos/syscom/Laurent.Denoue</a> o en <a href="http://www.fxpal.com/people/denoue/yawas/index.htm">http://www.fxpal.com/people/denoue/yawas/index.htm</a> . Para instalarlo, se bajan los ficheros de la página y se descomprimen en c:\program files\yawas\. A continuación ir a ese directorio y hacer doble click en yawas.bat para registrarlo en Internet Explorer. Una vez reiniciado Internet Explorer, la herramienta estará disponible para su uso.
<b>Compatibilidad con otros sistemas</b>
En principio no se prevén problemas de incompatibilidad, al trabajar con URLs extendidas
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
El usuario es el que decide a quién va a hacer participe de sus documentos anotados, por lo que el nivel de privacidad está controlado desde el primer momento
<b>Proceso para realizar la anotación</b>
<p>Para crear una anotación en una página Web, el usuario resalta una parte del documento. Ésta será el ancla de la anotación.</p> <ul style="list-style-type: none"> <li>• Con Internet Explorer, con un clic derecho, emergerá un Menú de Contexto que permitirá realizar la anotación. Éste interconecta con Javascript que reemplaza la URL actual por: <a href="http://localhost/URL?annotation=selection&amp;title=title_current_frame">http://localhost/URL?annotation=selection&amp;title=title_current_frame</a></li> <li>• Con Netscape, el usuario debe copiar la selección (Ctrl+C) y después hacer clic en el botón "Annotate" de la interfaz de Yawas. Usando DDE Yawas consigue la URL de la actual ventana de Netscape y lee el contenido del clipboard.</li> </ul> <p>Cuando se realiza una nueva anotación, Yawas:</p> <ul style="list-style-type: none"> <li>• Añade la anotación en el Archivo de Anotaciones (similar a un sitio de favoritos en formato HTML)</li> <li>• Recoge el documento solicitado</li> <li>• Modifica la fuente HTML para que al aparecer las anotaciones conocidas por esa página, incluya la recién creada</li> <li>• Devuelve el archivo HTML modificado al navegador</li> </ul> <p>El historial de navegación del browser no se modifica hasta que la URL actual es reemplazada, por tanto añadir anotaciones no afecta el modelo del usuario de la actividad de su navegador.</p>

Se reitera el hecho que la función de anotaciones no se ha mantenido en las últimas versiones, y habrá que instalar una versión antigua para poder implementarla
<b>Cómo se buscan las anotaciones</b>
En caso de trabajar con anotaciones, es necesario abrir la interfaz de Yawas (YawasQuickSearch) donde se muestra la tabla donde se listan las anotaciones. Las columnas de la tabla son los distintos atributos de la anotación (URL del documento anotado, texto seleccionado en el documento, nombre del autor, fecha de creación, tipo de comentario, comentario...). Se pueden aplicar filtros para visualizar las anotaciones actuando en cualquiera de las columnas. Si sólo se trabaja con resaltes, en dicha interfaz aparecerá todo el listado de los links a las páginas donde se han efectuado resaltes
<b>Cómo se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b>
Antes de enviar el archivo HTML al browser, YAWAS modifica el archivo original. Los textos anclados aparecen resaltados en amarillo (se puede cambiar el color) insertando los siguientes HTML tags a cada palabra anclada: <font style="background-color:yellow">. Las anclas son fácilmente identificables y no modifican mucho el documento original. Se seleccionan las anotaciones que se desean visualizar, y se pulsa "View Annotations" en el navegador para remarcarlas en el documento original. Una vez seleccionadas las anotaciones, se pueden realizar otras acciones, además de visualizarlas, como exportarlas para compartirlas. Para ello, pulsando "Export" se copian las XURLs en el clipboard, pudiendo pegarse en cualquier sitio (página Web, email...). Para recibir las anotaciones, se deben copiar las XURLs y después pulsar en "Import"
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
En principio no
<b>Si permite abrir debates (discussion threads)</b>
La herramienta no contempla en la actualidad la posibilidad de abrir debates de discusión
<b>Cómo se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
El texto con anotación anclada se visualiza resaltado en amarillo por defecto (este color puede cambiarse a gusto del usuario). Cuando se buscan términos en una página se resaltan en naranja
<b>Utilidad / facilidad de uso</b>
No es complicado aprender su funcionamiento
<b>Lenguajes y posibles formatos de importación y exportación</b>
YAWAS es una aplicación Java y Javascript, y al estar codificadas las anotaciones en URLs extendidas, la importación y exportación de las mismas son un procedimiento muy simple

**Tabla 15: Características de Yawas**



## 8.2.5 Annotation System

<b>Nombre del editor de anotaciones</b>
Annotation System
<b>Descripción</b>
Annotation System es un editor de anotaciones basado en el proyecto Amaya, que pretende mejorar el ambiente colaborativo de la W3C (el "Consortio de la World Wide Web") a través del uso de anotaciones compartidas con el objetivo de añadir conocimiento al sistema a través de la experiencia a medida que se analiza la información. Se trata de una herramienta que permite a los usuarios realizar anotaciones para expresar sus opiniones a medida que analizan la información en páginas Web, y también en imágenes, documentos de Microsoft Office, como MS Word, MS Power Point, u otros documentos como Adobe PDF. En el caso de las páginas Web las anotaciones se podrán realizar tanto en secciones específicas de la misma como a la página completa, mientras que en el resto de los casos tan sólo se podrán asociar al documento completo
<b>Desarrolladores (organización a la que pertenecen)</b>
Este sistema de anotación se desarrolló como parte del proyecto Semantic Web de la división DAKE (Data and Knowledge Engineering division) de la NCST (The National Centre for Software Technology) en Bangalore, India. Los desarrolladores son Raman RKVS y Venkatasubramani
<b>Información complementaria (dónde se puede encontrar)</b>
La dirección del proyecto Annotation System es: <a href="http://www.ncb.ernet.in/groups/dake/annotate/details.shtml#system">http://www.ncb.ernet.in/groups/dake/annotate/details.shtml#system</a> . Desde aquí podremos informarnos de las últimas novedades sobre el proyecto
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
Las anotaciones se visualizan con Microsoft Internet Explorer. Se instala un plug-in desarrollado en Visual C++, y se genera una barra para la herramienta de anotación en el lado izquierdo de la de Internet Explorer
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones pueden almacenarse en el PC local del autor o en la base de datos del servidor central para que esté disponible para todos los usuarios. La decisión del lugar de almacenaje la toma el autor de la anotación tras crearla
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Sí, el programa nos permite definir que tipo de anotación deseamos realizar
<b>Se pueden crear nuevas subclases fácilmente</b>
Los tipos de anotaciones están predefinidas (Advice, Change, Comment, Correction, Example, Explanation, Question, See Also) y no pueden crearse nuevos tipos. Sin embargo, si el usuario no encuentra una categoría que se adecue a su anotación, puede añadir una

nueva categoría a las preexistentes
<b>Se puede crear más de una anotación en una misma página Web</b>
Sí, se pueden crear todas las anotaciones que se deseen, basta seleccionar los textos a anotar
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
No se enlaza con servicios de Ontología
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
El estándar usado es RDF sobre XML para describir las anotaciones como metadatos
<b>Tecnologías de código abierto o de código propietario</b>
Indefinido en la actualidad, pues están en pleno proceso de fusión con una Institución de Investigación hindú, y van a replantearse temas de licencia
<b>Compatibilidad con otros sistemas</b>
Al estar la herramienta desarrollada con RDF y al trabajar con conceptos estándares son pocos los problemas de incompatibilidad que pueden surgir
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
Las anotaciones pueden hacerse compartidas por todos los usuarios o solo para un grupo. Una característica adicional es la provisión de categorización de temas de acuerdo a los usuarios aparte de la categoría general. Esto resulta especialmente útil en intranets
<b>Proceso para realizar la anotación</b>
La anotación la puede realizar el usuario: <ul style="list-style-type: none"><li>• Sobre un texto de una página Web, seleccionándolo</li><li>• Sobre el documento completo</li></ul> En el proceso de anotación se le pide al usuario que de ciertos detalles que constituirán los atributos de la anotación y son los siguientes: <ol style="list-style-type: none"><li>1) Annotation title: Título de la anotación que se está llevando a cabo</li><li>2) Source Document: Página Web en la que se está haciendo la anotación</li><li>3) Annotation author: Usuario que está haciendo la anotación</li><li>4) Subject Category: El usuario puede seleccionar la categoría de un pop-up.</li><li>5) Annotation type: Tipo de anotación: Advice, Change, Comment, Correction, Example, Explanation, Question, See Also</li><li>6) Annotation Rating: El usuario puede calificar la sección de la Web que está anotando, como: Excellent, Very Good, Good, Average, Poor</li><li>7) Created Date: Fecha y hora</li><li>8) Annotation text: Los comentarios del autor. * - Campos que se rellenan automáticamente</li></ol>
<b>Cómo se buscan las anotaciones</b>



<p>A la hora de visualizar las anotaciones, el usuario puede buscar sólo las pertenecientes a un determinado tema, las realizadas por un determinado autor, a través de un keyword, o realizar una búsqueda avanzada. Tras cargar las anotaciones, se pueden aplicar filtros para visualizar sólo las que al usuario le interesen. Los criterios de filtrado incluyen atributos como: anotaciones concretas, categorías de temas, calificación, tipo de anotación, autor o servidor donde se aloja</p>
<p><b>Cómo se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b></p>
<p>Las anotaciones almacenadas tanto en el PC propio como en el servidor central pueden ser cargadas en la página. Tras cargarlas, el botón 'Show/Hide Annotations' permite para ver/ocultar las anotaciones. Las anotaciones se indican por el icono:</p> <ul style="list-style-type: none"> <li>• Con un simple click en el icono, se remarca el texto seleccionado por el autor de la anotación</li> <li>• Con un doble click, se mostrará un cuadro de diálogo con todos los detalles de la anotación</li> <li>• Con un click derecho, se mostrará un menú de contexto a través del cual se tiene acceso a comandos como: remarcar texto, ver detalles de anotación, etc.</li> </ul>
<p><b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b></p>
<p>En principio no</p>
<p><b>Si permite abrir debates (discusion tretas)</b></p>
<p>Cualquier usuario puede responder a las anotaciones para expresar su opinión al respecto. La respuesta la puede dirigir al autor de la anotación vía e-mail, o almacenarla en la misma anotación creando un foro de discusión</p>
<p><b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b></p>
<p>Se utiliza un icono que representa un lápiz rojo</p>
<p><b>Utilidad / facilidad de uso</b></p>
<p>No es demasiado complicado aprender su funcionamiento si se tienen unos conocimientos básicos sobre anotaciones.</p>
<p><b>Lenguajes y posibles formatos de importación y exportación</b></p>
<p>Annotation System trabaja con el formato RDF que, al ser estándar, permite importar y exportar datos de forma sencilla</p>

**Tabla 16: Características de Annotation System**

## 8.2.6 Trellis Web

---

<b>Nombre del editor de anotaciones</b>
Trellis
<b>Descripción</b>
<p>Trellis es un entorno interactivo que permite añadir a los usuarios sus observaciones, opiniones y conclusiones a medida que analizan la información realizando anotaciones semánticas en documentos y en otros recursos on-line como páginas Web. En realidad es una forma de adquirir conocimiento donde el usuario puede añadir nuevo conocimiento al sistema basándose en su propia experiencia a medida que analiza la información. Trellis dispone de un vocabulario para realizar las anotaciones semánticas. La versión inicial de este vocabulario está disponible como un esquema/ontología en XMLS, RDFS y DAML+OIL. Además, Trellis permite a los usuarios extender este vocabulario y sus correspondientes esquemas. Trellis se utiliza para anotar decisiones (por ej.: un viaje), organizar materiales (por ej.: resultados de una búsqueda), analizar desacuerdos y controversias sobre un tema (por ej.: un debate político), para manipular información incompleta (por ej.: investigación genealógica), etc. Trellis genera las anotaciones en XML, RDF y en DAML+OIL teniendo en cuenta los esquemas y ontologías del vocabulario de anotación</p>
<b>Desarrolladores (organización a la que pertenece)</b>
Trellis ha sido desarrollado por el grupo de investigación "Interactive Knowledge Capture" del ISI (Information Sciences Institute) de la universidad del sur de California
<b>Información complementaria (dónde se puede encontrar)</b>
<p>La herramienta Trellis se puede encontrar on-line en <a href="http://trellis.semanticweb.org">http://trellis.semanticweb.org</a>. Se puede obtener información adicional en las siguientes direcciones:</p> <ul style="list-style-type: none"><li>• <a href="http://excalibur.isi.edu:8888/newTrellis/">http://excalibur.isi.edu:8888/newTrellis/</a></li><li>• <a href="http://excalibur.isi.edu:8888/newTrellis/toolHelp.htm">http://excalibur.isi.edu:8888/newTrellis/toolHelp.htm</a></li><li>• <a href="http://trellis.semanticweb.org/">http://trellis.semanticweb.org/</a></li></ul>
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
La aplicación Web funciona con Internet Explorer 5.5+ y también soporta parcialmente navegadores como Netscape 6.x y Mozilla 1.x.
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Todas las anotaciones se almacenan en el servidor en un directorio no compartido. Si se necesitan los ficheros pueden ser almacenados localmente. Se generan las anotaciones extraídas del análisis de un determinado usuario siguiendo varios tipos de lenguaje de marcas como pueden ser: XML, RDF y DAML+OIL
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Cada anotación se relaciona con una sentencia y esta a su vez esta relacionada con una



determinada fuente (URL)
<b>Se pueden crear nuevas subclases fácilmente</b>
Si. Cada anotación se relaciona con una sentencia y esta a su vez esta relacionada con una determinada fuente (URL), y el usuario es el encargado de crear las sentencias y fuentes
<b>Se puede crear más de una anotación en una misma página Web</b>
Esta aplicación permite realizar más de una anotación en una misma página Web basta con seleccionar el texto deseado
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
Las anotaciones se pueden guardar en formato RDF, DAML+OIL y OWL, pero cada anotación no está enlazada con un concepto de una ontología. Cada anotación se relaciona con una sentencia y esta a su vez esta relacionada con una determinada fuente (URL)
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
Los estándares que soporta son los siguientes: XML, RDF, DAML+OIL y OWL. Las ontologías se crean y se visualizan en forma de página HTML siguiente el formato XML
<b>Tecnologías de código abierto o de código propietario</b>
Para poder emplear el software de Trellis fuera de EEUU es necesario tener una licencia de control de exportación del gobierno de EEUU. Trellis básicamente es un CGI en Perl y una aplicación DHTML. Se proporciona su código fuente. Para el correcto funcionamiento de Trellis se requiere de un servidor Apache junto con soporte Perl superior a 5.6. Se puede obtener el software en la siguiente dirección: <a href="http://www.isi.edu/licensed-sw/trellis/">http://www.isi.edu/licensed-sw/trellis/</a> y en esa misma página se puede encontrar un documento que sirve de ayuda a la hora de comenzar la instalación
<b>Compatibilidad con otros sistemas</b>
Todo aquel que disponga de conexión a Internet y algún navegador como: Internet Explorer 5.5, Netscape 6.x y Mozilla 1.x será capaz de manejar esta aplicación debido a que funciona a través de Web
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
El primer paso para utilizar la aplicación es introducir la dirección de correo electrónico y la contraseña. De este modo cada usuario puede ver las anotaciones que ha realizado, pero también existe la posibilidad de ver las anotaciones sobre un determinado tema realizadas por otros usuarios siempre y cuando se introduzca el término de búsqueda deseado mediante un buscador
<b>Proceso para realizar la anotación</b>
Una vez que el usuario ha introducido su dirección de correo como login, aparece un listado de propósitos o motivos (purpose) que ha realizado y da la posibilidad de crear otros nuevos. Cada propósito puede tener asociados varias fuentes (source); se entiende como fuente la dirección de una página Web y a su vez cada fuente tiene asociadas varias sentencias (statement) que en realidad son una lista de anotaciones realizadas sobre una determinada fuente. Estas anotaciones se realizan simplemente seleccionando en la página

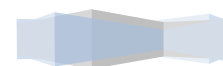
<p>Web del recurso elegido la sentencia a anotar. Para realizar una anotación, lo primero que se debe tener creado es un propósito. En el caso que no exista ninguno, se pulsa el botón “New Purpose”. Un propósito puede ser:</p> <ul style="list-style-type: none"> <li>• Una decisión a tener en cuenta</li> <li>• Algo que se quiere buscar</li> <li>• Una simple anotación de un conocimiento presente en un texto</li> </ul> <p>Se introduce el texto correspondiente a dicho propósito, tras lo cual aparece una nueva ventana dividida en cinco partes:</p> <ol style="list-style-type: none"> <li>1) Se describe el propósito junto con la conclusión que puede o no tener asociada.</li> <li>2) Se lista el conjunto de fuentes asociadas a un determinado propósito. Para crear una nueva basta con pulsar el botón “Add”. Se introduce el título de la fuente y la dirección Web asociada</li> <li>3) Lista de sentencias asociadas una determinada fuente. Para crear una nueva sentencia sólo se tiene que pulsar sobre el botón “Add”. Lo primero que hay que hacer es seleccionar la fuente sobre la cual se va a crear la sentencia. Se selecciona el texto que va a ser anotado en la página Web y se da un nombre significativo a la sentencia. El último paso es pulsar sobre el botón “Save” para guardar todo el proceso</li> <li>4) Ventana de Análisis donde se visualizan las unidades de análisis que el usuario ha introducido</li> <li>5) Ventana de edición de análisis en la el usuario puede editar, borrar, renombrar unidades de análisis. Además utilizando los botones “Import” e “Import Src” se puede obtener sentencias y recursos añadidos por otros usuarios</li> </ol>
<p><b>Como se buscan las anotaciones</b></p>
<p>Cada usuario tiene sus anotaciones siempre visibles, si desea ver anotaciones de otro usuario en la parte inferior de la ventana aparecen una serie de botones. Mediante las opciones “Import” e “Import Src” se puede realizar búsquedas de anotaciones introduciendo palabras clave o Keywords</p>
<p><b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b></p>
<p>La anotación aparece seleccionada en color amarillo dentro de la página Web</p>
<p><b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b></p>
<p>No</p>
<p><b>Si permite abrir debates (discusion threts)</b></p>
<p>No</p>
<p><b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b></p>
<p>La anotación aparece seleccionada en color amarillo dentro de la página Web</p>
<p><b>Utilidad / facilidad de uso</b></p>
<p>La aplicación es fácil de utilizar por medio de ventanas y botones se puede trabajar de</p>





manera muy cómoda, solamente hay que conocer la relación existente entre los propósitos (purpose), fuentes (source) y sentencias (statement)
<b>Lenguajes y posibles formatos de importación y exportación</b>
Trellis trabaja con ontologías de tipo RDF, DAML+OIL y OWL que se crean automáticamente por la aplicación y que el usuario no necesita conocer sus sintaxis

**Tabla 17: Características de Trellis Web**



## 8.2.7 MnM

---

<b>Nombre del editor de anotaciones</b>
MnM
<b>Descripción</b>
<p>MnM es una herramienta de anotación basada en ontologías que permite anotar páginas Web con contenidos semánticos de forma automática y semi-automática. MnM integra un navegador Web con un editor de ontologías y proporciona unas APIs abiertas para enlazar MnM con servidores de ontologías y para integrar MnM con herramientas de extracción de información. MnM trabaja con diferentes lenguajes de ontologías como RDF, DAML+OIL y OCML. El objetivo de la herramienta es permitir anotar documentos empleando anotaciones derivadas de ontologías preexistentes. La herramienta funciona</p> <ul style="list-style-type: none"><li>• anotando un conjunto de documentos de texto y/o HTML de entrenamiento y</li><li>• empleando esto para generar reglas léxicas que se pueden emplear para extraer información de forma automática de otro conjunto de documentos, a través de una máquina de extracción de información</li></ul> <p>Las instancias derivadas en este proceso se pueden emplear para propagar la ontología empleada en la anotación. Permite realizar anotaciones de forma automática y semiautomática, gracias a la integración con tecnologías de extracción de información. La herramienta está desarrollada en Java y la versión proporcionada en la Web está preparada para trabajar con el módulo de extracción de información Amilcare, desarrollado en la universidad de Sheffield. De todas formas, el API de la máquina de extracción de información es genérico, por lo tanto, se puede conectar a MnM otras herramientas de extracción de información</p>
<b>Desarrolladores (organización a la que pertenece)</b>
MnM ha sido desarrollado por el Knowledge Media Institute (KMI) del Open University del Reino Unido
<b>Información complementaria (dónde se puede encontrar)</b>
En la dirección <a href="http://kmi.open.ac.uk/spotlight/spotlight.cfm">http://kmi.open.ac.uk/spotlight/spotlight.cfm</a> hay información disponible sobre la herramienta y desde esta Web se puede bajar el software de la misma
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
Una vez realizadas las anotaciones de un documento de texto o HTML, estas se guardan junto al contenido del documento en un archivo nuevo local en formato XML y por tanto al abrirlo sólo se deduce la anotación por la etiqueta XML, tanto en el navegador Web de MnM como en cualquier otro navegador. Sólo se pueden visualizar gráficamente con la herramienta MnM en el momento inicial de crear la anotación en el documento de texto o HTML, o cuando las anotaciones son generadas de forma automática por la máquina de extracción de conocimiento. En ambos casos, las anotaciones se visualizan con un color de resaltado (diferente para cada caso) y al pasar el ratón por encima, se indica en una caja de texto el concepto y la ontología a los que corresponde la anotación



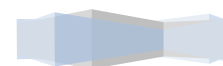
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones se guardan junto al documento anotado en formato XML en un archivo nuevo de tipo texto, HTML o XML en el ordenador local. Las ontologías a emplear para realizar las anotaciones se pueden acceder desde un archivo local (tipo RDF o DAML+OIL) o desde un servidor de ontologías remoto tipo WebOnto. Las instancias generadas con las anotaciones se pueden guardar en un archivo de base de conocimiento local (tipo RDF o DAML+OIL) o en un servidor tipo WebOnto
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Cada anotación de un documento es asignada a una relación de un concepto de la ontología activa
<b>Se pueden crear nuevas subclases fácilmente</b>
MnM no permite editar las ontologías. Estas deben de estar ya creadas con otra herramienta
<b>Se puede crear más de una anotación en una misma página Web</b>
Si, se pueden crear todas las anotaciones que se deseen
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
Las anotaciones siempre se realizan en base a una ontología. Las ontologías soportadas son las de OCML de WebOnto, RDF y DAML+OIL
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
Las ontologías con las que trabaja son OCML de WebOnto, RDF y DAML+OIL, y los documentos con las anotaciones se pueden almacenar en formato texto, HTML o XML donde las anotaciones se indican con etiquetas XML. Para instalar la herramienta se requiere Java 1.4.1 (o superior) y Ant 1.5.1 (o superior)
<b>Tecnologías de código abierto o de código propietario</b>
Se proporciona el código fuente de la herramienta. El software se puede emplear y modificar libremente sólo para fines no comerciales. La herramienta está desarrollada en Java pero sólo se ha testeado sobre Windows. Además, proporciona unas APIs abiertas para enlazar MnM con servidores de ontologías y para integrar MnM con herramientas de extracción de información
<b>Compatibilidad con otros sistemas</b>
MnM puede acceder a ontologías de servidores de WebOnto y almacenar en estos servidores las instancias creadas a partir de las anotaciones. También puede emplear herramientas de extracción de conocimiento desarrolladas por terceras personas, para lo cual previamente hay que desarrollar un plugin empleando las APIs proporcionadas por MnM. Actualmente MnM funciona con la herramienta de extracción de conocimiento de Amilcare desarrollado por Fabio Ciravegna ( <a href="mailto:f.ciravegna@dcs.shef.ac.uk">f.ciravegna@dcs.shef.ac.uk</a> ) de la universidad de Sheffield
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>

Las anotaciones son compartidas por todos los usuarios
<b>Proceso para realizar la anotación</b>
Para realizar una anotación, primeramente hay que seleccionar una ontología. Una vez que la ontología y el documento a anotar están abiertos, se selecciona el texto del documento a anotar y se asigna este texto a una relación de un concepto de la ontología. Se sigue este proceso hasta realizar todas las anotaciones necesarias en el documento, y se guarda el documento con las anotaciones en formato XML. Una vez finalizado el proceso de anotación de una serie de documentos que servirán para realizar el aprendizaje y posterior anotación de nuevos documentos con la herramienta de extracción de conocimiento, se guardan las instancias creadas con las anotaciones en una base de conocimientos. Con los documentos anotados, se pueden anotar nuevos documentos de forma automática empleando una herramienta de extracción de conocimiento como Amilcare, creando nuevas instancias de la ontología y actualizando así la base de conocimientos
<b>Como se buscan las anotaciones</b>
Las anotaciones están guardadas en formato XML en archivos de texto, HTML o XML. Estas anotaciones son empleadas por la herramienta de extracción de conocimiento seleccionada para anotar de forma automática nuevos documentos y a su vez generar nuevas instancias de la ontología y así actualizar la base de conocimientos
<b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b>
Las anotaciones se guardan en formato XML en un archivo de tipo texto, HTML o XML en modo local, por tanto no son accesibles por usuarios que navegan por la Web. La herramienta está orientada a crear bases de conocimientos con instancias de ontologías a partir de anotaciones de documentos
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
No
<b>Si permite abrir debates (discussion threts)</b>
No
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
Una vez realizadas las anotaciones de un documento de texto o HTML, éstas se guardan junto al contenido del documento en un archivo nuevo local en formato XML y por tanto al abrirlo sólo se deduce la anotación por la etiqueta XML, tanto en el navegador Web de MnM como en cualquier otro navegador. Sólo se pueden visualizar gráficamente con la herramienta MnM en el momento inicial de crear la anotación en el documento de texto o HTML, o cuando las anotaciones son generadas de forma automática por la máquina de extracción de conocimiento. En ambos casos, las anotaciones se visualizan con un color de resaltado (diferente para cada caso) y al pasar el ratón por encima, se indica en una caja de texto el concepto y la ontología a los que corresponde la anotación
<b>Utilidad / facilidad de uso</b>



Es fácil de aprender y de emplear
<b>Lenguajes y posibles formatos de importación y exportación</b>
MnM trabaja con ontologías de tipo RDF, DAML+OIL y OCML de WebOnto, creadas empleando herramientas de terceras personas, y las instancias creadas a partir de las anotaciones se pueden almacenar en bases de conocimientos de tipo RDF o DAML+OIL, o en un servidor WebOnto

**Tabla 18: Características de MnM**



## 8.2.8 OntoMat-Annotizer

---

<b>Nombre del editor de anotaciones</b>
OntoMat-Annotizer
<b>Descripción</b>
<p>OntoMat-Annotizer es una herramienta de anotación de páginas WEB interactiva y de fácil manejo. Su principal objetivo es asistir a los usuarios en las tareas de creación y mantenimiento de ontologías basadas en DAML+OIL como por ejemplo la creación de instancias DAML, atributos y relaciones. Incluye un browser de ontologías, que provee los servicios de navegación en la ontología y en sus instancias, y un browser HTML que muestra las partes de texto que han sido anotadas. Está basada en Java y lleva incorporada una interfaz para la instalación de extensiones a la aplicación en forma de plug-ins. El usuario objetivo de la herramienta es el anotador individual, por ejemplo, usuarios que desean enriquecer sus páginas WEB mediante la incorporación de metadatos DAML+OIL. En lugar de anotar las páginas de forma manual, con un editor de textos, OntoMat facilita la tarea mediante el uso de las facilidades de las APIs de interfaz gráfica de usuario como son la selección de texto copiar y pegar, arrastrar, etc... Soporta la fase de creación de metadatos y está previsto, que en un futuro próximo, se desarrolle un plug-in de extracción automática de información basado en un asistente que sugiera a los usuarios las partes del texto que son relevantes de cara a ser anotadas. Esta facilidad va a reducir considerablemente el tiempo necesario para realizar las anotaciones</p>
<b>Desarrolladores (organización a la que pertenece)</b>
Herramienta desarrollada dentro del proyecto OntoAgent en el cual participan investigadores de las universidades de Stanford y Karlsruhe
<b>Información complementaria (dónde se puede encontrar)</b>
En la dirección <a href="http://www.wdb.stanford.edu/OntoAgents/">http://www.wdb.stanford.edu/OntoAgents/</a> se puede encontrar información adicional del proyecto. El enlace para la descarga del software es <a href="http://annotation.semanticweb.org/ontomat/index.html">http://annotation.semanticweb.org/ontomat/index.html</a>
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
Las anotaciones sólo se pueden visualizar como marcas XML con cualquier editor de texto
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones se guardan como marcas XML en el documento anotado en un archivo local. Las ontologías a emplear para realizar las anotaciones se acceden desde un archivo local de extensión ".daml", donde la ontología estará escrita en RDF
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Cada anotación de un documento es asignada a un concepto de la ontología activa
<b>Se pueden crear nuevas subclases fácilmente</b>



Ontomat lleva incorporado un navegador para las ontologías con el que se pueden crear o modificar todos los elementos de la ontología
<b>Se puede crear más de una anotación en una misma página Web</b>
Si, se pueden crear todas las anotaciones que se deseen
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
Las anotaciones siempre se realizan en base a una ontología. Las ontologías soportadas son las de tipo DAML+OIL, lenguaje basado en RDF
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
Las ontologías con las que trabaja deben estar escritas con el lenguaje DAML+OIL, un lenguaje de ontologías basado en RDF. Las anotaciones se almacenan en los documentos como marcas XML en formato RDF
<b>Tecnologías de código abierto o de código propietario</b>
La herramienta está desarrollada en Java y utiliza las API de Jena RDF/DAML
<b>Compatibilidad con otros sistemas</b>
En principio OntoMat Annotizer es compatible con todas las herramientas desarrolladas en los proyectos de la Universidad de Kalsruhe de la familia Onto (On-To-Knowledge, OntoBroker, OntoServer, OntoWeb, OntoWise). Además también es compatible con las herramientas que utilizan el lenguaje DAML+OIL
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
Las anotaciones son compartidas por todos los usuarios
<b>Proceso para realizar la anotación</b>
Para empezar a realizar anotaciones en documentos, primeramente el usuario debe realizar las siguientes acciones de configuración: <ul style="list-style-type: none"><li>• Seleccionar la ontología en la que se va a basar (archivo local o URI remota)</li><li>• Seleccionar el conjunto de documentos locales (archivos locales) a emplear en la anotación</li><li>• Seleccionar el texto que se desea anotar sobre el HTML-Browser y pegarlo sobre la instancia correspondiente en el Ontology Browser</li><li>• Una vez finalizada la anotación se puede publicar sobre el documento HTML y guardarlo en local o en remoto</li></ul>
<b>Como se buscan las anotaciones</b>
No permite buscar anotaciones
<b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b>

No se visualizan
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
No
<b>Si permite abrir debates (discussion threts)</b>
No
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
No se visualizan
<b>Utilidad / facilidad de uso</b>
Es fácil de aprender y de emplear
<b>Lenguajes y posibles formatos de importación y exportación</b>
OntoMat trabaja con ontologías de tipo DAML+OIL, lenguaje basado en RDF. No tiene opciones de importación/exportación

**Tabla 19: Características de OntoMat-Annotizer**





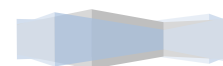
## 8.2.9 SHOE Knowledge Annotator

<b>Nombre del editor de anotaciones</b>
SHOE (Simple HTML Ontology Extensions) KNOWLEDGE ANNOTATOR
<b>Descripción</b>
<p>Se trata de un programa en Java (Java 1.0) que permite realizar anotaciones en documentos Web sin tener que preocuparse de los códigos HTML pues añade los tags necesarios automáticamente. El sistema está disponible como un applet de Java. Existen dos tipos de tags en SHOE:</p> <ul style="list-style-type: none"> <li>• Tags para construir ontologías.</li> <li>• Tags para anotar los documentos Web según una o varias ontologías.</li> </ul> <p>SHOE está todavía en desarrollo por lo que puede cambiar en algún momento. Incluso los desarrolladores piden a los usuarios que envíen sugerencias para su mejora. SHOE trabaja con categorías, relaciones, atributos, deducciones, etc. definidas por alguna ontología</p>
<b>Desarrolladores (organización a la que pertenece)</b>
Esta herramienta fue desarrollada por el Grupo de Investigación e Inteligencia Artificial PLUS (Parallel Understanding Systems) del departamento Computer Science de la Universidad de Maryland, en College Park, Maryland, USA
<b>Información complementaria (dónde se puede encontrar)</b>
La dirección del proyecto SHOE KNOWLEDGE ANNOTATOR es: <a href="http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html">http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html</a> . Desde aquí podremos obtener más información sobre el proyecto e incluso bajarnos la herramienta en su versión 0.93 de noviembre de 2000
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
No se han detectado problemas para Netscape 3.0+ o Internet Explorer 4.0 en Windows 95, NT. Tampoco para Netscape 4.0+ en SunOS
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones se insertan en el propio código HTML
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Sí, el programa nos permite definir que tipo de anotación deseamos realizar
<b>Se pueden crear nuevas subclases fácilmente</b>
Sí, pues se permite la creación de ontologías propias
<b>Se puede crear más de una anotación en una misma página Web</b>
Se pueden crear todas las anotaciones que se deseen, basta seleccionar los textos a anotar. n

Permite enlazar la anotación con alguna ontología para su mejor clasificación: Sí, de hecho, antes de realizar la anotación es necesario tener una ontología aplicable, o bien, crearnos una a la medida de nuestras necesidades
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
Sí, de hecho, antes de realizar la anotación es necesario tener una ontología aplicable, o bien, crearnos una a la medida de nuestras necesidades
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
Se necesita Java Development Kit (version 1 o superior) o Java Runtime Environment para bajarse y trabajar con la herramienta. Al tratarse de un applet de Java tiene las restricciones usuales de seguridad de los applets, como que sólo puede abrir páginas Web que estén en el servidor donde se encuentra el applet y no puede leer ni escribir en el PC del usuario. La versión aplicación no tiene estas restricciones pero para usarla primero debemos bajarnos el código. El código Java no está compilado por lo que es necesario un interpretador Java como java o jre para ejecutarlo. Es necesario tener la librería de SHOE (en Java 1.0) para poder analizar y manejar los documentos SHOE. La librería básica consta de los paquetes SHOE, HTML y Tree. La licencia de la librería es GNU Lesser General Public License , licencia para garantizar la libertad del software.
<b>Tecnologías de código abierto o de código propietario</b>
El sistema está continuamente actualizándose, y su código fuente está disponible libremente en: <a href="http://www.cs.umd.edu/projects/plus/SHOE/downloads/#ka">http://www.cs.umd.edu/projects/plus/SHOE/downloads/#ka</a>
<b>Compatibilidad con otros sistemas</b>
Existe una variante de la sintaxis SHOE, para propiciar la compatibilidad con XML. La naturaleza del lenguaje SHOE hace posible desarrollar numerosas herramientas y arquitecturas para su procesado. Para evaluar las anotaciones existen varias herramientas, como un buscador de páginas con anotaciones SHOE, que las almacena en una base de datos: Exposé, SHOE Search, PIQ (PARKA Interface for Queries) y Semantic Search
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
Se puede crear una base de datos con los registros de las páginas con anotaciones SHOE. Por tanto el tema de privacidad se limita a la gestión de permisos de acceso a dichos registros
<b>Proceso para realizar la anotación</b>
En primer lugar debemos tener una ontología aplicable ( <a href="http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html">http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html</a> ) o bien, crearnos una a nuestra medida ( <a href="http://www.cs.umd.edu/projects/plus/SHOE/ontologies.html">http://www.cs.umd.edu/projects/plus/SHOE/ontologies.html</a> ). Después podremos realizar las anotaciones en nuestra página Web con SHOE KNOWLEDGE ANNOTATOR siguiendo los siguientes pasos: <ol style="list-style-type: none"><li>1. Abrir el applet</li><li>2. En el menú, picar: View   Source HTML.</li><li>3. Abrir el código HTML de la página Web en un editor de texto local</li><li>4. Copiar todo el contenido que muestra el editor de texto</li><li>5. Pegarlo en la ventana del Knowledge Annotator</li><li>6. Picar una de las otras vistas del Knowledge Annotator</li><li>7. Usar las funciones que desea del Annotator</li></ol>



8. Picar View   Source HTML del menú 9. Copiar todo el código HTML de la ventana del Annotator 10. Pegarlo en el editor de texto como una nueva fila o sobrescribiendo el antiguo 11. Validar la anotación
<b>Como se buscan las anotaciones</b>
Es necesario abrir la interfaz de SHOE Knowledge Annotator. Seleccionando File>Open URL de la barra de menú, se abre la página cuyas anotaciones en SHOE se desean visualizar. Para visualizar la información anotada picar en View> y seleccionar qué se desea visualizar, se abrirá un cuadro y picando sobre la dirección, se mostrará en los cuadros de la interfaz la ontología usada y las anotaciones
<b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b>
No se visualizan
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
En principio no
<b>Si permite abrir debates (discusion threts)</b>
La herramienta no contempla en la actualidad la posibilidad de abrir debates de discusión
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
Para encontrar páginas con anotaciones SHOE, tendremos que usar un motor de búsqueda. Exposé ( <a href="http://www.cs.umd.edu/projects/plus/SHOE/downloads/">http://www.cs.umd.edu/projects/plus/SHOE/downloads/</a> ) es un robot web escrito en Java que reúne y asocia el conocimiento. Para extraer el conocimiento recopilado por Exposé existen otras herramientas Java: PIQ (Parka Interface for Queries) para usuarios expertos ( <a href="http://www.cs.umd.edu/projects/plus/SHOE/piq.html">http://www.cs.umd.edu/projects/plus/SHOE/piq.html</a> ) y SHOE Search para usuarios más inexpertos ( <a href="http://www.cs.umd.edu/projects/plus/SHOE/search.html">http://www.cs.umd.edu/projects/plus/SHOE/search.html</a> ). Tras la búsqueda, se visualiza la relación de anotaciones en una columna
<b>Utilidad / facilidad de uso</b>
No es demasiado complicado aprender su funcionamiento
<b>Lenguajes y posibles formatos de importación y exportación</b>



- SHOE&SGML: SHOE ha sido diseñada como una aplicación de SGML (Standard Generalized Markup Language). SGML es un estándar ISO para anotar documentos, y HTML está ahora especificada como una aplicación SGML. Todas las aplicaciones SGML requieren una DTD (Document Type Definition) que especifique qué tags pueden usarse en un documento y la estructura de relaciones de esos tags con otros. La DTD de SHOE se puede encontrar en:  
<http://www.cs.umd.edu/projects/plus/SHOE/SHOE.dtd>.
- SHOE&XML: El lenguaje XML fue diseñado para ser una versión simplificada de SGML para Internet. SHOE fue originalmente diseñada para trabajar con páginas HTML, pero al usar sintaxis SGML puede usarse con documentos XML realizando unas mínimas modificaciones. La versión XML del SHOE DTD puede encontrarse en:  
[http://www.cs.umd.edu/projects/plus/SHOE/SHOE\\_xml.dtd](http://www.cs.umd.edu/projects/plus/SHOE/SHOE_xml.dtd).
- SHOE&DAML: DAML (DARPA Agent Markup Language) es un programa de DARPA para desarrollar lenguaje y herramientas para la Web Semántica. Aquellos que estén interesados en relacionar DAML con SHOE, o deseen usar las ontologías de SHOE con sus proyectos DAML pueden obtener información al respecto en:  
<http://www.cs.umd.edu/projects/plus/DAML/>

**Tabla 20: Características de SHOE Knowledge Annotator**



## 8.2.10 SMORE

<b>Nombre del editor de anotaciones</b>
SMORE
<b>Descripción</b>
<p>SMORE es una herramienta que permite a los usuarios marcar sus documentos en RDF (W3C's Resource Description Framework) usando ontologías Web asociadas con términos y elementos específicos del usuario. Además de páginas Web, se pueden marcar E-mails y también fotos o parte de ellas. El objetivo del software es:</p> <ul style="list-style-type: none"> <li>• Proporcionar al usuario un entorno flexible con el que pueda crear sus páginas Web sin demasiados obstáculos a la hora de incluir marcaciones</li> <li>• Realizar las marcaciones con el mínimo conocimiento de RDF</li> <li>• Facilitar una referencia a ontologías existentes en Internet para que se use la más adecuada a las necesidades del usuario. Incluso se puede crear su propia ontología a partir de las ontologías existentes</li> <li>• Asegurar una marcación RDF exacta y completa que permita modificaciones sencillas.</li> </ul> <p>La herramienta usa Ekit API, un API de Java de código abierto para la edición de HTML. Se puede bajar libremente de la página de Hexidec en <a href="http://www.hexidec.com/">http://www.hexidec.com/</a>. SMORE dispone de una Ayuda que detalla cómo usar cada uno de sus componentes</p>
<b>Desarrolladores (organización a la que pertenece)</b>
Aditya Kalyanpur como miembro del Grupo de Investigación Mindswap de la Universidad de Maryland, en College Park, Maryland, USA
<b>Información complementaria (dónde se puede encontrar)</b>
La dirección del proyecto SMORE es: <a href="http://www.mindswap.org/~aditkal/editor.shtml">http://www.mindswap.org/~aditkal/editor.shtml</a> . En esa dirección se detallan las utilidades de la herramienta. También se puede bajar libremente la versión SMORE v3.0 de <a href="http://www.mindswap.org/~aditkal/editor2.shtml">http://www.mindswap.org/~aditkal/editor2.shtml</a>
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
Las anotaciones RDF, al estar basadas en XML, se pueden ver en la mayoría de los navegadores, como Netscape, Internet Explorer, etc.
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Existe la opción de almacenarlas en un archivo separado, o de insertarlas en el propio código HTML
<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Sí, pues cada anotación se asigna a un concepto de la ontología activa
<b>Se pueden crear nuevas subclases fácilmente</b>

Sí, pues se permite la creación de ontologías propias
<b>Se puede crear más de una anotación en una misma página Web</b>
Sí, todas las que el usuario estime necesarias
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
Sí, de hecho además de proporcionar una referencia a ontologías existentes en Internet para que se use la más adecuada a las necesidades del usuario, se puede crear su propia ontología a partir de las ontologías existentes
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
Es necesario tener JDK 1.4 o superior instalado en el PC. La última versión se puede encontrar en <a href="http://java.sun.com/j2se/1.4/download.html">http://java.sun.com/j2se/1.4/download.html</a> . Los APIs usados en el programa son: <ol style="list-style-type: none"><li>1. EKit version 0.9d (disponible en <a href="http://www.hexidec.com/ekit.php">http://www.hexidec.com/ekit.php</a>)</li><li>2. Michael Grove's RDF Parser API. (<a href="http://www.mindswap.org/~mhgrove">www.mindswap.org/~mhgrove</a>)</li><li>3. PhotoSMORE plug-in (<a href="mailto:glapizco@hotmail.com">glapizco@hotmail.com</a>)</li><li>4. ImageJ API (<a href="mailto:Wayne@codon.nih.gov">Wayne@codon.nih.gov</a>)</li></ol>
<b>Tecnologías de código abierto o de código propietario</b>
El código fuente está disponible libremente en: <a href="http://www.mindswap.org/~aditkal/editor2.shtml">http://www.mindswap.org/~aditkal/editor2.shtml</a> . Se puede distribuir y/o modificar libremente bajo los términos de la GNU <a href="http://www.gnu.org/copyleft/gpl.html#SEC1">http://www.gnu.org/copyleft/gpl.html#SEC1</a>
<b>Compatibilidad con otros sistemas</b>
La herramienta trabaja con Ekit y ImageJ API PhotoSMORE, y el API de Michael Grove's RDF Parser
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
Las anotaciones son compartidas por todos los usuarios que tienen acceso al documento con las anotaciones
<b>Proceso para realizar la anotación</b>
Con la página Web abierta en el editor y la ventana "Triples window" activa, basta seleccionar la palabra o texto al que se desea adjuntar una anotación y hacer un click izquierdo. El texto seleccionado aparecerá automáticamente en el campo Selected Text, pudiendo nosotros completar el resto de los campos: ontología escogida y clase. También podremos introducir los datos del triplete: un sujeto, objeto y predicado asociados. Es importante mencionar que mientras el sujeto y el objeto del triplete a introducir no tienen referencias ontológicas, el predicado sí que necesita tener establecidos elementos ontológicos. De ese modo, SMORE ayuda al usuario a distinguir entre predicados definidos por él mismo y los de una ontología, marcando los propios del usuario con un asterisco (*). Este asterisco permite continuar trabajando y poder retrasar el momento de la asignación de la ontología para cuando se tenga toda la información necesaria
<b>Como se buscan las anotaciones</b>



Es necesario disponer de un buscador que tenga opciones avanzadas que permitan encontrar archivos .rdf
<b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b>
No son visualizables
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
No
<b>Si permite abrir debates (discusion threts)</b>
No
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
No se visualizan
<b>Utilidad / facilidad de uso</b>
No parece difícil de comprender ni de usar
<b>Lenguajes y posibles formatos de importación y exportación</b>
Trabaja con ontologías RDF, y no tiene opciones de importación/exportación

**Tabla 21: Características de SMORE**

### 8.2.11 Melita

---

<b>Nombre del editor de anotaciones</b>
Melita
<b>Descripción</b>
<p>Melita es una herramienta de anotación de texto semiautomática basada en ontologías que tiene integrada una máquina de extracción de información (Amilcare). Implementa una metodología con el propósito de gestionar para los usuarios todo el proceso de anotación. Algunos de los pasos del proceso, que hasta ahora se realizan manualmente, el sistema los puede automatizar y gestionar fácilmente.</p> <p>Las principales capacidades de Melita se pueden resumir en cuatro grupos, la tarea de gestión, la extracción, el aprendizaje, y el etiquetado de la Información de forma autónoma. Esto se lleva a cabo gracias al empleo de un buen interfaz junto con un potente algoritmo de extracción de información.</p> <p>Su principal objetivo es ayudar al usuario en el proceso de anotación. Melita va cambiando gradualmente el role del usuario del de anotador al de supervisor. El sistema es proactivo en el sentido de que toma la iniciativa para llevar a acabo cualquier preprocesamiento que se empleará en el futuro.</p> <p>La novedad que aporta Melita es que se puede ajustar el sistema de extracción de información para que proporcione el nivel que se desee de proactividad e intrusión. Melita también proporciona un mecanismo de almacenamiento de documentos que ordena los documentos de forma dinámica después de cada anotación para encontrar el documento que mejor cubre las áreas sin explorar del dominio.</p> <p>Los documentos con el menor número de etiquetas se toman para cubrir las áreas sin explorar del dominio donde se pueden aprender nuevas reglas si es que se anotan. Este enfoque ha llevado a una convergencia más rápida del algoritmo mientras que se ha superado el problema de la dispersión de los datos</p>
<b>Desarrolladores (organización a la que pertenece)</b>
Melita ha sido desarrollado por el grupo de Procesamiento de Lenguaje Natural en el departamento de informática de la universidad de Sheffield del Reino Unido, como demostrador de cómo una máquina de extracción de información, tras una fase de aprendizaje, permite anotar documentos automáticamente en base a una ontología
<b>Información complementaria (dónde se puede encontrar)</b>
En la dirección <a href="http://www.dcs.shef.ac.uk/~alexiei/WebSite/University/Melita/index.html">http://www.dcs.shef.ac.uk/~alexiei/WebSite/University/Melita/index.html</a> hay información disponible sobre la herramienta. Para obtener el software es necesario ponerse en contacto con los desarrolladores, Alexiei Dingli ( <a href="mailto:A.Dingli@dc.shef.ac.uk">A.Dingli@dc.shef.ac.uk</a> ) y Fabio Ciravegna ( <a href="mailto:F.Ciravegna@dc.shef.ac.uk">F.Ciravegna@dc.shef.ac.uk</a> )
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
Las anotaciones sólo se pueden visualizar gráficamente con Melita. También se pueden ver como marcas XML con cualquier editor de texto





<p><b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b></p>
<p>Melita es un sistema cliente-servidor, donde:</p> <ul style="list-style-type: none"> <li>• El cliente contiene todas las funciones relacionadas con el interfaz gráfico de usuario. Se encarga de todas las interacciones entre el usuario y la aplicación. Además, también proporciona un interfaz con el servidor para permitirle al usuario gestionar el servidor.</li> <li>• El servidor es la parte de la aplicación que gestiona el algoritmo de la máquina de extracción de información. Esta separación se debe a que este tipo de algoritmos necesita gran cantidad de recursos de memoria y procesamiento. No sería factible tener al interfaz de usuario y a la máquina de extracción de información ejecutándose en el mismo proceso. Las dos principales razones son que primero Melita puede funcionar sin la máquina de extracción de información (aunque esta máquina es importante para darle valor añadido a Melita). Segundo, si los dos se ejecutaran en el mismo proceso, la máquina de extracción de información cogería la mayor parte de los recursos del sistema al interfaz de usuario haciendo que el sistema fuera poco servible.</li> </ul> <p>Las anotaciones se guardan como marcas XML en el documento anotado en un archivo local. Las ontologías a emplear para realizar las anotaciones se acceden desde un archivo local de extensión “.ont”, donde la ontología estará escrita en XI, lenguaje basado en PROLOG</p>
<p><b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b></p>
<p>Cada anotación de un documento es asignada a un concepto de la ontología activa</p>
<p><b>Se pueden crear nuevas subclases fácilmente</b></p>
<p>Melita no permite editar las ontologías. Estas deben de estar ya creadas con otra herramienta</p>
<p><b>Se puede crear más de una anotación en una misma página Web</b></p>
<p>Si, se pueden crear todas las anotaciones que se deseen</p>
<p><b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b></p>
<p>Las anotaciones siempre se realizan en base a una ontología. Las ontologías soportadas son las de tipo XI, lenguaje basado en PROLOG</p>
<p><b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b></p>
<p>Las ontologías con las que trabaja deben estar escritas con el lenguaje XI, un lenguaje de ontologías basado en PROLOG. Las anotaciones se almacenan en los documentos como marcas XML. Melita es un sistema cliente-servidor:</p> <ul style="list-style-type: none"> <li>• El cliente requiere Windows 2000 o Linux, 128 Mbs de RAM, 800 Mhz de CPU y Java 1.4 (o superior)</li> <li>• El servidor necesita Windows 2000 o Linux, 512 Mbs de RAM, 800 Mhz de CPU y Java 1.4 (o superior)</li> </ul>
<p><b>Tecnologías de código abierto o de código propietario</b></p>
<p>Para obtener la herramienta es necesario disponer de nombre de usuario y contraseña en la</p>

Web de Melita. La herramienta está desarrollada en Java y emplea una arquitectura cliente-servidor basada en RMI. En el servidor está integrada la máquina de extracción de información de Amilcare desarrollada por la universidad de Sheffield
<b>Compatibilidad con otros sistemas</b>
Melita tiene integrada la máquina de extracción de información de Amilcare desarrollada por la universidad de Sheffield y emplea ontologías escritas en XI, lenguaje basado en PROLOG
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
Las anotaciones son compartidas por todos los usuarios
<b>Proceso para realizar la anotación</b>
<p>Para empezar a realizar anotaciones en documentos, primeramente el usuario debe realizar las siguientes acciones de configuración:</p> <ul style="list-style-type: none"><li>• Seleccionar la ontología en la que se va a basar (archivo local)</li><li>• Seleccionar el conjunto de documentos locales (archivos locales) a emplear en la anotación</li><li>• Configurar la barra de intervención de la máquina de extracción de información indicando los niveles de certeza y de sugerencia. Estos niveles son empleados por el sistema para saber como mostrarle al usuario aquellas anotaciones que genera automáticamente la máquina de extracción de información. Se mostrarán como anotaciones si están por encima del nivel de certeza, y como sugerencias si están entre los niveles de sugerencia y de certeza. El resto no se mostrarán.</li></ul> <p>Una vez realizada la configuración, el usuario podrá realizar anotaciones sobre el documento abierto seleccionando un concepto de la ontología y a continuación un determinado fragmento de texto a anotar del documento. Las anotaciones se mostrarán con un rectángulo relleno del mismo color que el color asignado al concepto de la ontología. Al mismo tiempo la máquina de extracción de información aprenderá de las anotaciones realizadas por el usuario para sugerirle nuevas anotaciones generadas automáticamente. El usuario tendrá la opción de aceptarlas o no.</p>
<b>Como se buscan las anotaciones</b>
Quando se abre un documento en Melita, se muestran las anotaciones que contenga y que correspondan a los conceptos de la ontología en curso. Estas anotaciones se mostrarán con rectángulos del mismo color que el concepto correspondiente. El usuario tiene la opción de seleccionar aquellos conceptos de la ontología sobre los que desea que se muestren las anotaciones del documento, para poder así filtrar y ver mejor los conceptos que le interesan
<b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b>
Las anotaciones se guardan como marcas de XML en el documento anotado en modo local, por tanto no son accesibles por usuarios que navegan por la Web
<b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b>
No



<b>Si permite abrir debates (discusion threts)</b>
No
<b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b>
<p>Existen tres tipos diferentes de anotaciones las cuales se visualizan de diferente manera desde Melita:</p> <ul style="list-style-type: none"><li>• Anotaciones insertadas por el usuario. Estas tienen el mismo color que el respectivo concepto de la ontología (cada concepto de la ontología tiene un color asignado) y se muestran con un rectángulo relleno coloreado</li><li>• Sugerencias proporcionadas por el algoritmo de aprendizaje. Estas anotaciones se muestran como un rectángulo blanco con un bordo coloreado (teniendo el mismo color que el correspondiente concepto de la ontología)</li><li>• Certezas proporcionadas por el algoritmo de aprendizaje. Estas anotaciones son similares a las anotaciones del usuario excepto por el hecho de que tiene un borde negro alrededor</li></ul>
<b>Utilidad / facilidad de uso</b>
Es fácil de aprender y de emplear. Tras una fase inicial de anotación manual por parte del usuario, el sistema va aprendiendo y sugiriendo automáticamente nuevas anotaciones, haciendo que el proceso de anotación de nuevos documentos sea más rápido
<b>Lenguajes y posibles formatos de importación y exportación</b>
Melita trabaja con ontologías de tipo XI, lenguaje basado en PROLOG. No tiene opciones de importación/exportación

**Tabla 22: Características de Melita**



## 8.2.12 GATE

<b>Nombre del editor de anotaciones</b>
GATE (General Architecture for Text Engineering)
<b>Descripción</b>
<p>GATE es una infraestructura de componentes de software reutilizables para dar soporte al procesamiento del lenguaje humano, entendiendo el procesamiento de lenguaje natural como la principal forma de emplear y hacer frente a la Web, lo cual supone acercar el conocimiento y las máquinas inteligentes a la gente a través de la tecnología del lenguaje natural.</p> <p>GATE tiene integrado el sistema de extracción de información ANNIE, desarrollado por el mismo departamento de la universidad de Sheffield que GATE, y permite anotar documentos de forma automática. También tiene integrado el editor de ontologías Protégé. Soporta los siguientes tipos de documentos: XML, RTF, email, HTML, SGML y texto plano, y permite abrir documentos locales y documentos de la Web. Para todos estos tipos, se analiza el formato y se convierte en un único modelo de anotación unificado. El formato de anotación es una modificación del formato TIPSTER (<a href="http://www.itl.nist.gov/div894/894.02/related_projects/tipster/">http://www.itl.nist.gov/div894/894.02/related_projects/tipster/</a>). Los documentos GATE y las anotaciones se almacenan en una base de datos (Oracle o PostgreSQL) o en archivo local empleando la serialización de Java, y se visualizan a través del entorno GATE.</p> <p>GATE está desarrollado en JAVA y está disponible como software de libre distribución bajo la licencia de librería de GNU. Permite integrar nuevos beans de JAVA para darle nuevas funcionalidades al sistema. Se pueden guardar las anotaciones como instancias de una ontología en formato DAML+OIL. Por defecto, GATE emplea una ontología que se puede extender fácilmente. Si se desea cargar en la herramienta otra ontología ya creada, es necesario cargar el módulo "Ontogazetteer" que no es de libre distribución y que para obtenerlo es necesario pedir una licencia separada</p>
<b>Desarrolladores (organización a la que pertenece)</b>
GATE ha sido desarrollado por el grupo NLP (Natural Language Processing) del departamento de informática de la universidad de Sheffield del Reino Unido y por el Institute for Language, Speech and Hearing (ILASH)
<b>Información complementaria (dónde se puede encontrar)</b>
En la dirección <a href="http://gate.ac.uk/">http://gate.ac.uk/</a> hay información disponible sobre la herramienta y desde esta Web se puede bajar el software de la misma
<b>Con que tipo de navegador se pueden visualizar las anotaciones</b>
Las anotaciones sólo se pueden visualizar gráficamente con GATE
<b>Dónde se almacenan las anotaciones (distribuido o centralizado)</b>
Las anotaciones se pueden almacenar en una base de datos (Oracle o PostgreSQL), o en un archivo local empleando la serialización de Java. El formato de anotación es una modificación del formato TIPSTER ( <a href="http://www.itl.nist.gov/div894/894.02/related_projects/tipster/">http://www.itl.nist.gov/div894/894.02/related_projects/tipster/</a> )



<b>Se puede clasificar por tipo de anotaciones o subclases (ejemplo, referencia, comentario,...)</b>
Cada anotación de un documento es asignada a un concepto de la ontología activa
<b>Se pueden crear nuevas subclases fácilmente</b>
GATE permite crear fácilmente nuevos tipos de anotaciones aparte de los tipos que tiene definidos por defecto
<b>Se puede crear más de una anotación en una misma página Web</b>
Si, se pueden crear todas las anotaciones que se deseen
<b>Permite enlazar la anotación con alguna ontología para su mejor clasificación</b>
Las anotaciones siempre se realizan en base a una ontología. Si se instala el módulo Ontogazetteer, GATE permite cargar otras ontologías tipo RDF, aparte de la que está instalada por defecto. Este módulo no es de libre distribución y requiere una licencia separada
<b>Estándares en los que está basado (RDF,...) y requerimientos de infraestructuras</b>
GATE está desarrollado en Java, con Java beans, y requiere Java 1.4 (o superior). Se ha testeado que funciona correctamente sobre LINUX, Windows y Solaris. Permite anotar los siguientes tipos de documentos: XML, RTF, email, HTML, SGML y texto plano, y permite abrir documentos locales y documentos de la Web. Para todos estos tipos, se analiza el formato y se convierte en un único modelo de anotación unificado. El formato de anotación es una modificación del formato TIPSTER ( <a href="http://www.itl.nist.gov/div894/894.02/related_projects/tipster/">http://www.itl.nist.gov/div894/894.02/related_projects/tipster/</a> ). Las ontologías con las que trabaja son de tipo RDF y tiene integrada la herramienta de edición de ontologías Protégé. Las anotaciones se pueden almacenar en una base de datos (Oracle o PostgreSQL), o en un archivo local empleando la serialización de Java. GATE permite guardar las anotaciones como instancias de una ontología en formato DAML+OIL
<b>Tecnologías de código abierto o de código propietario</b>
La herramienta está desarrollada en Java con Java beans y se proporciona el código fuente de la misma bajo la licencia GNU. Se puede integrar en la herramienta nuevos componentes desarrollados como Java beans para darle nuevas funcionalidades. En el manual de usuario se explica como se pueden cargar estos componentes. GATE tiene integrada la máquina de extracción de información ANNIE desarrollada por el mismo departamento de la universidad de Sheffield que GATE, y también tiene integrado el editor de ontologías Protégé
<b>Compatibilidad con otros sistemas</b>
GATE tiene integrada la máquina de extracción de información ANNIE desarrollada por el mismo departamento que GATE, y también tiene integrado el editor de ontologías Protégé. También permite cargar ontologías tipo RDF si se instala el módulo Ontogazetteer, el cual no es de libre distribución
<b>Se comparten todas las anotaciones o pueden ser privadas con un grupo</b>
Las anotaciones son compartidas por todos los usuarios que tienen acceso al documento con las anotaciones. En el caso de que el documento anotado se encuentre almacenado como

<p>archivo local, los derechos de acceso al mismo estarán controlados por el sistema operativo, pero en el caso de que se almacene en una base de datos ORACLE, habrá definidos diferentes tipos de usuario y grupos de acceso</p>
<p><b>Proceso para realizar la anotación</b></p>
<p>Para empezar a realizar anotaciones en los documentos, primero hay que cargar la máquina de extracción de información ANNIE, abrir el documento o conjunto de documentos que se desean anotar e indicarle a GATE que aplique ANNIE sobre dichos documentos. A continuación, se mostrarán las anotaciones generadas automáticamente por la herramienta con rectángulos del mismo color que los conceptos de la ontología a los que se refieren. El usuario podrá modificar dichas anotaciones o eliminarlas, y crear nuevos conceptos y anotaciones</p>
<p><b>Como se buscan las anotaciones</b></p>
<p>Cuando se abre un documento en GATE y se ordena que lo procese la máquina de extracción de información ANNIE, se muestran automáticamente las anotaciones que contenga y que correspondan a los conceptos de la ontología en curso. Estas anotaciones se mostrarán con rectángulos del mismo color que el concepto correspondiente. El usuario tiene la opción de seleccionar aquellos conceptos de la ontología sobre los que desea que se muestren las anotaciones del documento, para poder así filtrar y ver mejor los conceptos que le interesan</p>
<p><b>Como se visualiza la anotación cuando el usuario navega por una página Web que tiene anotaciones realizadas por otros usuarios</b></p>
<p>Las anotaciones se guardan en el documento anotado en modo local empleando la serialización de Java, o en una base de datos (Oracle o PostgreSQL) por tanto no son accesibles por usuarios que navegan por la Web</p>
<p><b>Si permite o permitirá imágenes, por ejemplo, una foto del autor</b></p>
<p>No</p>
<p><b>Si permite abrir debates (discusion threts)</b></p>
<p>No</p>
<p><b>Como se visualizan las anotaciones (subrayado, círculo, lápices,...)</b></p>
<p>Las anotaciones se muestran con un rectángulo relleno del mismo color que el respectivo concepto de la ontología (cada concepto de la ontología tiene un color asignado)</p>
<p><b>Utilidad / facilidad de uso</b></p>
<p>GATE está orientado a emplear sus componentes para crear aplicaciones de procesamiento de lenguaje. Es fácil de aprender y de emplear</p>
<p><b>Lenguajes y posibles formatos de importación y exportación</b></p>
<p>GATE puede trabajar con ontologías de tipo RDF si se carga el módulo Ontogazetteer y permite guardar las anotaciones como instancias de una ontología en formato DAML+OIL</p>

**Tabla 23: Características de GATE**



### 8.2.13 LinkFactory

<b>Nombre de la herramienta de creación de ontologías</b>
LinkFactory
<b>Versión y fecha de lanzamiento de la versión</b>
Release. 7/01/02
<b>Licencia</b>
No es gratuita. Se necesita comprar la licencia
<b>Desarrolladores (organización a la que pertenece)</b>
Compañía L&C, Language & Computing
<b>Información complementaria (dónde se puede encontrar)</b>
<a href="http://www.landcglobal.com/">http://www.landcglobal.com/</a>
<b>Descripción</b>
<p>LinkFactory es un sistema de gestión de ontologías que se utiliza para construir y gestionar el conocimiento lingüístico médico LinKBase (representación formal del mundo de la medicina). LinkFactory es una herramienta que puede ser utilizada para construir grandes y complejas ontologías independientes del lenguaje formal. No se limita a pequeñas ontologías como la mayoría de editores de ontologías. Entre los objetivos principales que llevaron al desarrollo de LinkFactory destacan:</p> <ul style="list-style-type: none"> <li>• La habilidad para conectar lenguajes con este modelo conceptual y utilizarlo para el procesamiento del lenguaje natural</li> <li>• Construcción de un modelo completo de clasificación de conceptos junto con sus relaciones y conceptos</li> <li>• La capacidad para conectar la asociación de términos y conceptos con otros sistemas como SNOMED o ICD-9</li> </ul>
<b>Arquitectura software</b>
LinkFactory está diseñada como un plataforma independiente, es una aplicación Java 100%. El sistema contiene una parte servidora, donde se almacenan los datos en una base de datos relacional y una cliente. Permite que varios clientes trabajen a la vez sobre una misma ontología
<b>Extensibilidad</b>
LinkFactory es una solución flexible y escalable para la gestión de una ontología. Se ha utilizado para modelar términos médicos y también se ha testeado sobre otros dominios
<b>Gestión de backups</b>
No

<b>Acceso a la base del conocimiento vía Web a través de un navegador</b>
El servidor de LinkFactory soporta clientes de Internet
<b>Almacenamiento de la ontología</b>
LinkFactory almacena la ontología en una base de datos compatible con Java que ha sido probada en Oracle, Sybase y SqlServer
<b>Almacenamiento en base de datos</b>
Los datos se almacenan en una base de datos relacional. El acceso a dicha base de datos se realiza por medio de unas funciones API. Las bases de datos relacionales con las que se puede utilizar son: Oracle, Sysbase y SQLServer
<b>Tamaño de las ontologías o conocimiento base creado por la herramienta</b>
LinkFactory permite disponer de cerca de 1,5 millones de conceptos, 3 millones de términos y 3 millones de enlaces, obteniendo así cerca de 7.000.000 elementos de conocimiento. Diariamente se pueden realizar alrededor de 2.000 modificaciones
<b>Características del modelado</b>
Modelos de lógica de descripción T-Box (terminológica) y ABox (asercional). Soporta herencia múltiple de conceptos y relaciones.  Dispone de la identificación de los criterios necesarios y suficientes para definir un concepto. Gestiona múltiples ontologías en un mismo T-Box
<b>Lenguaje base usado para codificar la ontología</b>
El lenguaje primario utilizado para codificar la ontología es una lógica de descripción extendida
<b>Plataformas donde puede ejecutarse</b>
LinkFactory ha sido probado con éxito en plataformas como Windows, Linux y Solaris
<b>Formatos de importación/exportación de la ontología</b>
Los primeros desarrollos de la herramienta son compatibles con el sistema DAML+OIL. En estos momentos se puede asegurar que LinkFactory es compatible con DAML+OIL en un 90%. Otros formatos de exportación son: XML y RDF(S)
<b>Interfaz gráfica</b>
Posibilidad de visualizar las clases y relaciones en forma de árbol. El interfaz funciona con Java beans y cada bean tiene una funcionalidad y vista específicas para la ontología en cuestión. El usuario puede combinar los beans como desee, proporcionando así al usuario una herramienta potente para visualizar y manejar los datos
<b>Comprobación automática de consistencia (sintáctica, referencial y/o lógica) de la ontología</b>
Realiza chequeos utilizando: reglas de restricción, disyunciones formales, etc.





<b>Soporte multi-usuario es decir, desarrollo concurrente de la ontología</b>
Posibilita un entorno multiusuario siempre y cuando se tengan privilegios de autor y se realicen auditorías específicas a los conceptos de la jerarquía
<b>Soporte para la comparación y fusión de ontologías creadas por aplicaciones independientes</b>
Compara y enlaza ontologías a través de una ontología principal, los conceptos se relacionan en base a relaciones formales e información léxica
<b>Soporte léxico (capacidad de referenciación léxica de los elementos de la ontología (por ejemplo sinónimos) y el procesado léxico de contenido(búsqueda/filtrado de términos de la ontología)</b>
Existe una distinción estricta entre concepto-término, descripción del lexema. La búsqueda se puede realizar mediante comodines
<b>Información adicional</b>
LinkFactory incluye una base de datos servidor, aplicación cliente y servidor. Fue diseñado para realizar el diseño de complejas ontologías médicas. Dispone de un API de Java Beans y opcionalmente dispone de una aplicación que genera índices semánticos, código automático y extracción de información
<b>Contacto</b>
Información general: Oficinas principales: <a href="mailto:info@landc.be">info@landc.be</a> Inversiones: <a href="mailto:maarten@landc.be">maarten@landc.be</a> Ventas y Marketing : <a href="mailto:dan@landc.be">dan@landc.be</a> , <a href="mailto:gordon@landc.be">gordon@landc.be</a> (Norte América)
<b>Compatibilidad con otros sistemas</b>
Permite abrir ontologías de tipo DAML+OIL y RDFS
<b>Utilidad / facilidad de uso</b>
La herramienta es muy intuitiva y fácil de utilizar. Las relaciones entre conceptos se representan mediante un árbol. Las relaciones se pueden modificar, añadir o eliminar simplemente pinchando y arrastrando con el ratón. La herramienta permite autorizar o denegar ciertos permisos a un usuario o grupo de usuarios

**Tabla 24: Características de LinkFactory**

## 8.2.14 OIEd

---

<b>Nombre de la herramienta de creación de ontologías</b>
OilEd
<b>Versión y fecha de lanzamiento de la versión</b>
3.5.4. Septiembre de 2003
<b>Licencia</b>
Open source bajo licencia GPL
<b>Desarrolladores (organización a la que pertenece)</b>
Universidad de Manchester del Reino Unido (Information Management Group)
<b>Información complementaria (dónde se puede encontrar)</b>
<a href="http://oiled.man.ac.uk/">http://oiled.man.ac.uk/</a>
<b>Descripción</b>
<p>OilEd es un editor gráfico de ontologías que permite al usuario construir ontologías usando DAML+OIL. Está basado en Protégé solo que no proporciona un entorno de desarrollo de ontologías completo. Es compatible con RDF y RDFs.</p> <p>El modelo del conocimiento es el mismo que el usado por DAML+OIL, aunque mejorado por el uso de un marco para representar el modelado. Las clases están definidas en función de sus super clases y las restricciones de sus propiedades, con axiomas adicionales captando más relaciones tales como las desuniones.</p> <p>Una de las tareas principales de OilEd está centrada en editar ontologías o schemas, en oposición a la adquisición de conocimiento o la construcción de grandes bases de conocimientos formados por instancias.</p> <p>Un aspecto clave del comportamiento de OilEd es el uso del “razonador” llamado FaCT para clasificar ontologías y chequear la consistencia a través de una traducción desde DAML+OIL hasta la descripción lógica de SHIQ. Esto permite al usuario describir las clases de su ontología y que su razonador determine el sitio apropiado en la jerarquía</p>
<b>Arquitectura software</b>
Standalone
<b>Extensibilidad</b>
No
<b>Gestión de backups</b>
No



<b>Acceso a la base del conocimiento vía Web a través de un navegador</b>
<ul style="list-style-type: none"> <li>• RDF URIs</li> <li>• nombres de dominio limitados</li> <li>• XML Schema muy limitado</li> </ul>
<b>Almacenamiento de la ontología</b>
En ficheros de extensión “.daml”
<b>Almacenamiento en base de datos</b>
No
<b>Tamaño de las ontologías o conocimiento base creado por la herramienta</b>
No muy grandes. No soporta el desarrollo de ontologías a gran escala
<b>Características del modelado</b>
Axiomas restrictivos de DAML; same-class-as; tipos de datos XML Schema limitados; creación de metadatos; permite el uso de expresiones arbitrarias como rellenos y en los axiomas restrictivos; uso explícito de los cuantificadores, carece de vista jerárquica de las propiedades
<b>Lenguaje base usado para codificar la ontología</b>
DAML+OIL
<b>Plataformas donde puede ejecutarse</b>
Multiplataforma, al estar desarrollado en Java, con lo cual en cualquier plataforma con JAVA 2, es decir JDK 1.4.0 o superior
<b>Formatos de importación/exportación de la ontología</b>
RDFS; SHIQ
<b>Interfaz gráfica</b>
Visualización de clases y propiedades globales desde ficheros de tipo Graphviz
<b>Comprobación automática de consistencia (sintáctica, referencial y/o lógica) de la ontología</b>
La consistencia de la ontología es comprobada a través del razonador FaCT
<b>Soporte multi-usuario es decir, desarrollo concurrente de la ontología</b>
No
<b>Soporte para la comparación y fusión de ontologías creadas por aplicaciones independientes</b>
No

<b>Soporte léxico (capacidad de referenciación léxica de los elementos de la ontología (por ejemplo sinónimos) y el procesado léxico de contenido(búsqueda/filtrado de términos de la ontología)</b>
Uso limitado de sinónimos
<b>Información adicional</b>
--
<b>Contacto</b>
<a href="mailto:seanb@cs.man.ac.uk">seanb@cs.man.ac.uk</a>
<b>Compatibilidad con otros sistemas</b>
Abre ontologías de los siguientes tipos: <ul style="list-style-type: none"><li>• RDF, RDFs</li><li>• GO</li><li>• OIL Text</li><li>• SHIQ Knowledge Bases</li></ul>
<b>Utilidad / facilidad de uso</b>
Muy fácil e intuitivo de usar

**Tabla 25: Características de OLEd**



## 8.2.15 OntoEdit

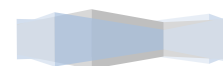
<b>Nombre de la herramienta de creación de ontologías</b>
OntoEdit Professional (OntoStudio)
<b>Versión y fecha de lanzamiento de la versión</b>
2.6.4. La fecha de lanzamiento de la versión 2.6 es 30/11/2002 aunque existe una primera versión de OntoEdit desde el 1/1/2002. Ahora está en su forma OntoStudio 2.0
<b>Licencia</b>
Se vende bajo licencia aunque existe una versión limitada por 3 meses de uso
<b>Desarrolladores (organización a la que pertenece)</b>
La herramienta ha sido desarrollada por la empresa Ontoprise GmbH. Ontoprise es una empresa fundada en 1999 como una spin off de la Universidad de Karlsruhe (Alemania). El AIFB (Institute of Applied Informatics and Formal Description Methods) de la Universidad de Karlsruhe, uno de los centros pioneros en el mundo en la investigación sobre temas de Web Semántica, forma parte de su parte de su partenariado de investigación. La mayoría de los productos comercializados por Ontoprise han sido creados en modo experimental por el AIFB. Estos productos son transferidos a la empresa para que desarrolle una versión comercial de los mismos
<b>Información complementaria (dónde se puede encontrar)</b>
<a href="http://www.ontoprise.com/">http://www.ontoprise.com/</a>
<b>Descripción</b>
OntoEdit Professional es un Ontology Engineering Environment que soporta el desarrollo y mantenimiento de ontologías de forma gráfica. OntoEdit es la parte visible de un potente modelo ontológico interno. El paradigma soporta un lenguaje de representación que permite el modelado de conceptos, relaciones y axiomas. Además, dispone de varias vistas gráficas que dan soporte a las diferentes fases del ciclo de vida de la ingeniería ontológica. La herramienta es muy abierta ya que está basada en un mecanismo muy flexible de plug-in s que permiten extender fácilmente su funcionalidad. La interfaz de plug-in es abierta por lo que terceros fabricantes pueden extender la funcionalidad de la herramienta mediante la implementación de sus propios plug-ins. Además, este mecanismo de plug-ins, permite la extensión de la herramienta mediante nuevos módulos desarrollados por la misma empresa Ontoprise. En este sentido, tenemos OntoEdit Professional que es una extensión de OntoEdit que a su vez, es una extensión de OntoEditFree
<b>Arquitectura software</b>
Standalone
<b>Extensibilidad</b>
A través del uso de plug-ins

<b>Gestión de backups</b>
No
<b>Acceso a la base del conocimiento vía Web a través de un navegador</b>
OntoEdit no proporciona soporte directo para acceder a la base de conocimiento a través de Internet, aunque, al ser esta exportable a una BBDD se podría consultar a partir de esta mediante una aplicación que accediera directamente a la BBDD
<b>Almacenamiento de la ontología</b>
OXML
<b>Almacenamiento en base de datos</b>
OntoEdit soporta cualquier base de datos con una interfaz JDBC, en la práctica esto implica la mayor parte de las bases de datos relacionales como son Oracle, MySQL, Microsoft SQL Server, DB2, PostgreSQL, etc.
<b>Tamaño de las ontologías o conocimiento base creado por la herramienta</b>
No se han encontrado datos
<b>Características del modelado</b>
Permite herencia múltiple y jerarquías de relaciones. Como aspecto a destacar, a través del plug-in OntoMap, soporta mappings entre estructuras de datos de ontologías diferentes. Los mappings se pueden establecer a varios niveles: <ul style="list-style-type: none"><li>• conceptos sobre conceptos</li><li>• atributos sobre conceptos</li><li>• atributos sobre atributos</li><li>• relaciones sobre relaciones</li></ul>
<b>Lenguaje base usado para codificar la ontología</b>
F-Logic
<b>Plataformas donde puede ejecutarse</b>
OntoEdit funciona sobre la máquina virtual Java 1.4 por lo que es un software portable que puede ejecutarse sobre la mayoría de sistemas (Windows, Linux, etc...)
<b>Formatos de importación/exportación de la ontología</b>
<ul style="list-style-type: none"><li>• DAML+OIL</li><li>• RDF Schema</li><li>• OXML</li><li>• Frame-Logic</li><li>• Base de datos JDBC: Un proyecto de OntoEdit se puede almacenar en cualquier base de datos JDBC compatible.</li></ul>



<b>Interfaz gráfica</b>
OntoEdit permite navegar y editar de forma gráfica las ontologías a través de un plug-in visualizador. Este plug-in representa los conceptos y las instancias como nodos y las relaciones como arcos entre los nodos
<b>Comprobación automática de consistencia (sintáctica, referencial y/o lógica) de la ontología</b>
A través del plug-in OntoBroker
<b>Soporte multi-usuario es decir, desarrollo concurrente de la ontología</b>
Permite el bloqueo de objetos
<b>Soporte para la comparación y fusión de ontologías creadas por aplicaciones independientes</b>
Si, de forma manual mediante mappings
<b>Soporte léxico (capacidad de referenciación léxica de los elementos de la ontología (por ejemplo sinónimos) y el procesado léxico de contenido(búsqueda/filtrado de términos de la ontología)</b>
Si a través de plug-ins
<b>Información adicional</b>
--
<b>Contacto</b>
<a href="mailto:Ontoedit@ontoprise.de">Ontoedit@ontoprise.de</a>
<b>Compatibilidad con otros sistemas</b>
Sí
<b>Utilidad / facilidad de uso</b>
OntoEdit es una herramienta muy intuitiva y fácil de usar. Sus características gráficas facilitan enormemente las diferentes acciones a realizar en el proceso de ingeniería para el desarrollo de una ontología, desde la definición de la base de conocimiento hasta la introducción de los datos. Cabe destacar sus facilidades de cara a la visualización de la ontología, esta característica está muy lograda y, de un vistazo, proporciona al usuario una idea del alcance de la ontología en la que está trabajando

**Tabla 26: Características de OntoEdit**



## 8.2.16 Ontolingua Server

---

<b>Nombre de la herramienta de creación de ontologías</b>
Ontolingua
<b>Versión y fecha de lanzamiento de la versión</b>
Versión 1.0.650. 14 de octubre de 2002
<b>Licencia</b>
La herramienta es de uso libre, basta registrarse
<b>Desarrolladores (organización a la que pertenece)</b>
Universidad de Stanford
<b>Información complementaria (dónde se puede encontrar)</b>
Información complementaria (dónde se puede encontrar): Se puede encontrar más información acerca de la herramienta en: <a href="http://www.ksl.stanford.edu/software/ontolingua/">http://www.ksl.stanford.edu/software/ontolingua/</a> . La herramienta está disponible en: <a href="http://www-ksl-svc.stanford.edu:5915">http://www-ksl-svc.stanford.edu:5915</a> o <a href="http://www-ksl-svc.stanford.edu/">http://www-ksl-svc.stanford.edu/</a> , donde también se ofrece información sobre el uso del servidor. Para usarla es necesario registrarse y es accesible desde cualquier browser estándar. El uso de la herramienta es libre siempre y cuando no se haga uso indebido de ella
<b>Descripción</b>
Ontolingua es un software que ofrece un entorno de colaboración distribuido para crear, editar, modificar y usar ontologías, basado en el principio de la jerarquía múltiple. El sistema consiste en un servidor y en un lenguaje de representación. El servidor está localizado en la Universidad de Stanford y soporta 150 usuarios activos. Ofrece distintas réplicas (mirrors) como el servidor web de webODE. La creación de la ontología se facilita mediante la posibilidad de incluir partes de otras ya existentes en un registro de ontologías y la posibilidad de incluir términos básicos incluidos en una frameontology
<b>Arquitectura software</b>
Ontolingua esta diseñada como una plataforma independiente, es una aplicación Java 100%. El sistema contiene en una parte servidora, donde se almacenan los datos en una base de datos relacional, y una cliente. Permite que varios clientes trabajen a la vez sobre una misma ontología
<b>Extensibilidad</b>
Ontolingua es una solución flexible y escalable para la gestión de ontologías. Se está utilizando para modelar distintos tipos de términos como la gestión de documentos, vocabulario físico, químico, matemático, etc.





<b>Gestión de backups</b>
No
<b>Acceso a la base del conocimiento vía Web a través de un navegador</b>
El servidor de Ontolingua soporta sus clientes vía web a través de los navegadores: NETSCAPE >=0.93, MSIE >=3.0, AOL >4.0, MOSAIC 2.5B2, MACWEB >=1.0ALPHA3, y lynux >=2.3.7)
<b>Almacenamiento de la ontología</b>
Las ontologías se almacenan en el servidor. Además éstas pueden ser convertidas a distintos formatos para su uso por otras aplicaciones
<b>Almacenamiento en base de datos</b>
Los datos se almacenan en una base de datos relacional
<b>Tamaño de las ontologías o conocimiento base creado por la herramienta</b>
El tamaño medio de las ontologías es de unas 1000 definiciones
<b>Características del modelado</b>
El modelo permite incluir relaciones de ontologías, dependencias circulares (usando nombres cualificados para referirse a símbolos desde la perspectiva de otra ontología) y el perfeccionamiento polimórfico (cuando alguna propiedad sutil de inclusión de ontologías implícitas resulta clara)
<b>Lenguaje base usado para codificar la ontología</b>
El lenguaje utilizado para codificar la ontología es el lenguaje tradicional Ontolingua/KIF, que puede ser traducido a otros sistemas de representación implementados en la herramienta
<b>Plataformas donde puede ejecutarse</b>
Se puede acceder a Ontolingua mediante un navegador web convencional. Al tratarse de una aplicación Java, puede ejecutarse desde la mayoría de las plataformas. Está basado en la plataforma J2EE -Java 2 Enterprise Edition- ( <a href="http://java.sun.com/j2ee">http://java.sun.com/j2ee</a> )
<b>Formatos de importación/exportación de la ontología</b>
Los usuarios de Ontolingua pueden exportar e importar la ontología. Ontolingua Server es compatible con Kif, Ontolingua, OKBC, LOOM, EPIKIT y GENERIC-FRAME
<b>Interfaz gráfica</b>
La interfaz de Ontolingua consta de dos marcos: <ul style="list-style-type: none"><li>• El marco superior, posee un número de iconos y menús.</li><li>• El marco inferior se usa para mostrar la información y para introducir los nuevos datos. Este marco cambia cada vez que lo hace la información, lo que asegura que la información presentada al usuario siempre está actualizada. Si el servidor de</li></ul>

<p>Ontolingua está siendo usado por varios usuarios trabajando en la misma ontología, los cambios sugeridos por uno de ellos se refrescan cada vez que se pique en cualquier sitio.</p> <p>Ontolingua utiliza un navegador gráfico. Este navegador puede usarse para obtener una visión de las jerarquías de los conceptos junto con los ejemplos</p>
<p><b>Comprobación automática de consistencia (sintáctica, referencial y/o lógica) de la ontología</b></p>
<p>Realiza comprobaciones de consistencia. Las restricciones de sentencias se usan en deducciones no-terminológicas como la herencia de sentencias usadas para asegurar la consistencia. Posee un motor de validaciones diseñado para resolver cualquier inconsistencia que pudiera haber sido introducida durante el desarrollo y proceso de mantenimiento de la ontología. Se realizan validaciones en tiempo real durante el proceso de edición / creación (comprueba tipos de valores, violaciones cardinales...), pero para una completa comprobación de la consistencia, el usuario debe recurrir al motor de validaciones</p>
<p><b>Soporte multi-usuario es decir, desarrollo concurrente de la ontología</b></p>
<p>El servidor está diseñado para permitir la cooperación entre distintos usuarios para el desarrollo de una ontología. Esto permite el trabajo cooperativo entre distintos usuarios de cualquier parte del mundo. Para ello, el servidor usa la idea de usuarios y grupos. El propietario de la ontología puede darles a ciertos grupos de usuarios el acceso de lectura y escritura a la ontología. Además se puede trabajar simultáneamente en una ontología durante una sesión de grupo. Cuando alguien realice un cambio, el servidor lo notificará al resto de los usuarios. Otro tipo de acceso definible es el “no permitir editar una ontología” a ciertos usuarios. Existe una lista de deshacer compartida, que permite a cualquier usuario deshacer los cambios realizados por otros. Existe también una lista de rehacer</p>
<p><b>Soporte para la comparación y fusión de ontologías creadas por aplicaciones independientes</b></p>
<p>Usando el comando “Compare” del menú, se pueden comparar los contenidos de una ontología con otra. El servidor de Ontolingua proporciona un registro de ontologías, permitiendo modificar las existentes. Además las ontologías del registro pueden ser fundidas o incluidas en una nueva ontología</p>
<p><b>Soporte léxico (capacidad de referenciación léxica de los elementos de la ontología (por ejemplo sinónimos) y el procesado léxico de contenido(búsqueda/filtrado de términos de la ontología)</b></p>
<p>Existe un browser de ontologías para búsqueda de términos. Es posible definir composiciones and/or</p>
<p><b>Información adicional</b></p>
<p>También se puede acceder al servidor de Ontolingua con otros programas que conozcan cómo usar ontologías almacenadas en lenguaje Ontolingua. Esto permite a agentes inteligentes la búsqueda de información en las ontologías</p>
<p><b>Contacto</b></p>
<p><a href="mailto:ontology-librarian@ksl.stanford.edu">ontology-librarian@ksl.stanford.edu</a></p>
<p><b>Compatibilidad con otros sistemas</b></p>
<p>Sí</p>



<b>Utilidad / facilidad de uso</b>
La interfaz es sencilla de entender y usar. El sentido de los comentarios es claro y en caso de necesitar más información acerca de un cierto comando existe una función de ayuda disponible. Además existe un tour guiado para familiarizarse con los comandos básicos

**Tabla 27: Características de OntolinguaServer**



### 8.2.17 Protégé

---

<b>Nombre de la herramienta de creación de ontologías</b>
Protégé-2000
<b>Versión y fecha de lanzamiento de la versión</b>
Release 1.9. 8 de agosto de 2003
<b>Licencia</b>
El código fuente es gratis bajo la licencia de código abierto Mozilla Public License (MPL)
<b>Desarrolladores (organización a la que pertenece)</b>
Stanford Medical Informatics- The Stanford University School of Medicine-California
<b>Información complementaria (dónde se puede encontrar)</b>
<a href="http://protégé.stanford.edu/index.html">http://protégé.stanford.edu/index.html</a>
<b>Descripción</b>
<p>Protégé-2000 es una herramienta de software integrado que permite a los desarrollares crear sistemas basados en el conocimiento. Las aplicaciones desarrolladas con Protégé-2000 se usan para la resolución de problemas y la toma de decisiones en un dominio determinado.</p> <p>El usuario puede realizar las siguientes operaciones:</p> <ul style="list-style-type: none"><li>• Creación de una ontología</li><li>• Optimización de datos de entrada de formularios</li><li>• Inserción de datos</li></ul> <p>Es también una plataforma que puede ser extendida con elementos gráficos como tablas, diagramas, componentes de animación para acceder a otros sistemas basados en el conocimiento embebidos en otras aplicaciones.</p> <p>También puede ser usado como librería por otras aplicaciones para acceder y mostrar bases del conocimiento</p>
<b>Arquitectura software</b>
Standalone
<b>Extensibilidad</b>
A través del uso de plug-in's
<b>Gestión de backups</b>
No



<b>Acceso a la base del conocimiento vía Web a través de un navegador</b>
Protégé-2000 no proporciona soporte directo para esto pero se podría hacer a través de servlets. Protégé-2000 puede ser ejecutado a través de un applet
<b>Almacenamiento de la ontología</b>
<ul style="list-style-type: none"><li>• Ficheros de texto standard</li><li>• RDF Schema</li><li>• Base de datos JDBC: Un proyecto de Protégé 2000 se puede almacenar en cualquier base de datos JDBC compatible.</li></ul>
<b>Almacenamiento en base de datos</b>
Protégé-2000 soporta cualquier base de datos con un driver JDBC 1.0, en la práctica esto implica la mayor parte de las bases de datos relacionales entre las que se han probado: Oracle, MySQL, Microsoft SQL Server y Microsoft Access
<b>Tamaño de las ontologías o conocimiento base creado por la herramienta</b>
<p>El sistema real más grande creado contiene más de 150.000 frames (clases e instancias). Los sistemas de este tamaño requieren el uso de una base de datos.</p> <p>Los proyectos basados en ficheros (CLIPS, XML, RDF) más grandes que 50.000 frames llegan a ser impracticables debido a los elevados requerimientos de memoria (&gt;1 GB) así como a los significativos tiempos de carga (~1 minuto). Usando una base de datos se elimina estas limitaciones debido al uso del caching.</p> <p>Los frames se almacenan en memoria al ser necesarios y los que no son necesarios son eliminados. Se han hecho experimentos con sistemas más grandes.</p> <p>Existen problemas de funcionamiento con la versión 1.8 de Protégé para conocimientos base superiores a un millón de frames. Con cinco millones de frames el sistema deja de ser utilizable</p>
<b>Características del modelado</b>
Concepto de herencia múltiple y jerarquías de relación (pero con una única clase por instancia); metaclasses; soporte de especificación de instancias; axiomas con restricciones Prolog, FLogic, OIL y el lenguaje de axioma de Protégé (PAL) mediante plug-ins
<b>Lenguaje base usado para codificar la ontología</b>
OKBC Open Knowledge Base Connectivity (OKBC) es un API para acceder a la base del conocimiento almacenados en sistemas de representación del conocimiento (KRSS)
<b>Plataformas donde puede ejecutarse</b>
Protégé-2000 funciona en cualquier plataforma que soporte la versión 1.3 del JDK. Esto incluye todas las versiones de 32 bits de MS Windows (95/98/NT/2000), las versiones más usuales de Unix incluyendo Linux, Solaris, AIX, y HP/UX, y sobre el Mac OS X v 10.1. Protégé-2000 no funciona sobre Windows 3.1 ni sobre el sistema operativo Mac "Classic" ya que dichas plataformas no soportan la versión 1.3 del JDK

<b>Formatos de importación/exportación de la ontología</b>
RDF(S); XML Schema; RDB schema a través del plug-in Data Genie; (DAML+OIL backend debido a 4Q'02 desde SRI)
<b>Interfaz gráfica</b>
Visualización de clases y propiedades globales mediante el plug-in GraphViz y vistas de graficos anidados con el plug-in de edición Jambalaya
<b>Comprobación automática de consistencia (sintáctica, referencial y/o lógica) de la ontología</b>
Existen plug-ins para añadir y chequear los axiomas que imponen restricciones: PAL, FaCT
<b>Soporte multi-usuario es decir, desarrollo concurrente de la ontología</b>
Varios usuarios pueden leer de la misma base de datos y hacer cambios que no afecten a los demás. Sin embargo, no hay soporte para varios usuarios intentando modificar el mismo elemento del conocimiento base. El soporte multi-usuario está en fase de desarrollo
<b>Soporte para la comparación y fusión de ontologías creadas por aplicaciones independientes</b>
Semi-automatizado mediante Anchor-PROMPT
<b>Soporte léxico (capacidad de referenciación léxica de los elementos de la ontología (por ejemplo sinónimos) y el procesado léxico de contenido(búsqueda/filtrado de términos de la ontología)</b>
WordNet plug-in ; emparejamiento de caracteres especiales (sólo el API)
<b>Información adicional</b>
Soporta la metodología CommonKADS
<b>Contacto</b>
<a href="mailto:protégé-help@smi.stanford.edu">protégé-help@smi.stanford.edu</a>
<b>Compatibilidad con otros sistemas</b>
Sí
<b>Utilidad / facilidad de uso</b>



Protégé2000 es muy intuitivo y fácil de usar. La creación de ontologías se hace a través de distintos menús y los datos se introducen mediante formularios

#### Proceso para crear una ontología

El proceso de creación de una ontología se realiza llevando a cabo los siguientes pasos:

- Creación de un proyecto: se inicializa Protégé asegurándose que se escoge la opción "New Project" y está seleccionada "Standard Text Files", se guarda seleccionando nombres para el proyecto (archivo con extensión .pproj), archivo que va a contener las clases (extensión .pont) así como las instancias.(extensión .pins).
- Creación de clases y subclases: Por defecto aparece seleccionada la pestaña de "Classes". Todas las clases que se vayan a crear van a estar subordinadas de la clase :THING. Seleccionando la letra C, se accede al menú de creación de una clase, el nombre por defecto lo proporciona el sistema pero se cambia inmediatamente. Para crear una subclase se sigue el mismo proceso pero seleccionando siempre la clase de la que cuelga.
- Creación de slots (atributos de las clases), se selecciona la clase primero y se despliega el formulario de "Template Slots" tras pulsar sobre la C, aparece el menú para introducir los datos del slot.
- Creación de instancias: Las instancias son los datos reales del conocimiento base. Es conveniente no introducir instancias hasta no tener completamente creada la estructura de clases y atributos. Se selecciona la pestaña de "Instancias" y tras elegir la clase se escoge la opción C del panel "Direct instances".

Una vez definida la ontología se almacena en la estructura de directorios previamente indicada los ficheros extensión rdf y rdfs. El primero contiene los datos es decir, las instancias mientras que el segundo contiene toda la estructura de la jerarquía de clases de la ontología.

**Tabla 28: Características de Protégé**

## 8.2.18 WebODE

---

<b>Nombre de la herramienta de creación de ontologías</b>
WebODE (Web Ontology Design Environment)
<b>Versión y fecha de lanzamiento de la versión</b>
WebODE 2.0. 23 de diciembre de 2002
<b>Licencia</b>
El editor de ontologías de la plataforma disponible como una aplicación Web es de libre acceso, basta registrarse. Si posteriormente se desea integrar la plataforma en algún otro sistema, se realizaría un estudio de los módulos y servicios que serían necesarios y las licencias se harían en función a las necesidades
<b>Desarrolladores (organización a la que pertenece)</b>
WebODE ha sido desarrollada por Ontology & Knowledge Reuse Group, del Departamento de Inteligencia Artificial de la Facultad de Ciencias Informáticas de la Universidad Técnica de Madrid (UPM)
<b>Información complementaria (dónde se puede encontrar)</b>
Se puede encontrar más información a cerca de la herramienta en: <a href="http://delicias.dia.fi.upm.es/webODE/">http://delicias.dia.fi.upm.es/webODE/</a> . Para instalar la herramienta habrá que solicitar un login y password de acceso en: <a href="http://delicias.dia.fi.upm.es/webODE/Access.html">http://delicias.dia.fi.upm.es/webODE/Access.html</a>
<b>Descripción</b>
<p>WebODE es una herramienta para modelar el conocimiento usando ontologías. Está basado en el ampliamente probado y testado Methontology de la Escuela Técnica de Ciencias Informáticas de Madrid. Es la parte Web de ODE (Ontology Design Environment) e incluye muchas características nuevas e interesantes que la convierten en una herramienta simple y a la vez potente para la creación de ontologías. Las ontologías disponen de los siguientes componentes principales: conceptos, grupos de conceptos, relaciones predefinidas, taxonomías (herencias simples y múltiples), relaciones ad-hoc, constantes, fórmulas, ejemplos (de conceptos y relaciones), propiedades, términos importados y referencias. Otras características de interés son:</p> <ul style="list-style-type: none"><li>• Colección de ejemplos, que permite vislumbrar el mismo modelo conceptual en distintos escenarios.</li><li>• Distintas vistas del mismo modelo conceptual, que permiten crear y almacenar distintas partes de la ontología, personalizando la visualización de la misma según los requerimientos de uso.</li><li>• Posibilidad de añadir a las relaciones ad-hoc, propiedades matemáticas como: reflexión, simetría, etc. y otras propiedades definidas por el usuario.</li></ul>
<b>Arquitectura software</b>
La implementación de WebODE está realizada con: Java, RMI, CORBA, XML, y tecnología





propia como Minerva Application Server. Se proporciona así la máxima flexibilidad e interoperatividad con otras aplicaciones empresariales. WebODE se basa en un modelo de tres niveles: nivel del cliente, intermedio, y de las bases de datos.

- En el nivel del cliente, se encuentra la interfaz de usuario, para cuya presentación se usa HTML, CSS (Cascading Style Sheets) y XML, para permitir la interoperatividad con otras aplicaciones. Para que el cliente trabaje más rápidamente y así se aligere la carga del servidor, se usa JavaScript y Java.
- Se recomienda Microsoft Internet Explorer 5.0 o superior para obtener las máximas ventajas que presenta la interfaz de usuario. Por otro lado el Java plug-in versión 1.2.x debe estar instalado en el browser (este plug-in puede bajarse de <http://java.sun.com>).
- En el nivel intermedio, existen dos subniveles:
  - Presentación: Responsable de generar el contenido que estará presente en el browser del usuario, y del manejo de las solicitudes/preguntas de los clientes. En este subnivel se usa tecnología de servlets o JSPs (Java Server Pages).
  - Lógica: Permite el acceso directo a las ontologías por medio de una API proporcionada por el servidor de aplicaciones Minerva Application Server, que ofrece acceso a servicios a través de RMI-IIOP (Remote Method Invocation-Internet Inter ORB Protocol).
- El tercer nivel comprende la información, los datos. Para las ontologías se usan bases de datos relacionales (Oracle) accesibles por medio de JDBC (Java Database Connectivity).

#### **Extensibilidad**

WebODE es una solución flexible y escalable para la gestión de ontologías. Se ha utilizado para desarrollar distintas ontologías, como vocabulario físico, químico, matemático, físicos, informáticos, de negocios, etc.

#### **Gestión de backups**

Sí, se trata de un servicio de WebODE cuando se realiza la administración a través de la consola de administración remota Minerva Management Console

#### **Acceso a la base del conocimiento vía Web a través de un navegador**

WebODE es una herramienta 100% vía web

#### **Almacenamiento de la ontología**

WebODE almacena la ontología en una base de datos compatible con Java

#### **Almacenamiento en base de datos**

WebODE usa para el almacenamiento de las ontologías bases de datos relacionales accesibles por medio de JDBC (Java Database Connectivity ). Ha sido probado tanto con Oracle como con MySQL. El acceso a la base de datos es un servicio independiente de la plataforma. De hecho está diseñado como un componente plug en la arquitectura, lo cual permite una fácil sustitución por otros módulos encargados de la gestión de datos, como las bases de datos basadas en XML o los repositorios basados en RDF

<b>Tamaño de las ontologías o conocimiento base creado por la herramienta</b>
La ontología de mayor tamaño en la actualidad posee 23755 términos
<b>Características del modelado</b>
WebODE es una herramienta 100% vía web. Se trata de un modelo basado en un motor de deducciones, y un constructor de axiomas
<b>Lenguaje base usado para codificar la ontología</b>
El servidor de ontología genera automáticamente código Ontoligua, que proporciona un conjunto de traductores que transforman dicho código en lenguajes objetivos como LOOM, pure-KIF, Prolog, etc.
<b>Plataformas donde puede ejecutarse</b>
WebODE está desarrollado en Java. Por tanto, puede ser ejecutado en cualquier plataforma. Los desarrolladores lo utilizan actualmente en Windows 2000, pero también ha sido probado en otros entornos
<b>Formatos de importación/exportación de la ontología</b>
Las ontologías de WebODE pueden ser integradas con otros sistemas usando los servicios automáticos de importación (WebODE, XML, X-CARIN) y exportación (WebODE, XML, FLOGIC, RDFS, OIL, X-CARIN, DAML+OIL)
<b>Interfaz gráfica</b>
Las interfaces de usuario son gráficas y basadas en formas
<b>Comprobación automática de consistencia (sintáctica, referencial y/o lógica) de la ontología</b>
Realiza chequeos utilizando: restricciones de clases, de valores numéricos, verificación de consistencia taxonómica (ejemplos comunes de clases desenlazadas, bucles, etc.)
<b>Soporte multi-usuario es decir, desarrollo concurrente de la ontología</b>
Posibilita un entorno multiusuario y de uso sincronizado de los mecanismos, que permite que varios usuarios editen la misma ontología sin errores
<b>Soporte para la comparación y fusión de ontologías creadas por aplicaciones independientes</b>
Para realizar fusión de ontologías existen un servicio, ODEMerge, que permite mezclar dos ontologías en el mismo dominio, utilizando tablas de sinónimos e hiperónimos
<b>Soporte léxico (capacidad de referenciación léxica de los elementos de la ontología (por ejemplo sinónimos) y el procesado léxico de contenido(búsqueda/filtrado de términos de la ontología)</b>
El modelo de conocimientos de WebODE permite definir sinónimos y acrónimos para conceptos, atributos, relaciones, etc. Respecto a las capacidades de búsqueda, la interfaz de usuario del editor de ontologías no tiene actualmente ninguna función de búsqueda/filtrado. Sin embargo, estas funciones están implementadas actualmente en la plataforma, y se utilizan por ejemplo en otras de las aplicaciones de la misma: ODESeW. Un ejemplo de estas



búsquedas se puede ver en <a href="http://www.esperonto.net">http://www.esperonto.net</a> . Se permite búsquedas por palabra clave o por valores de atributos
<b>Información adicional</b>
Este servidor de web, ofrece también una réplica (mirror) del servidor web del Knowledge System Laboratory de la Universidad de Stanford
<b>Contacto</b>
<a href="mailto:webode@delicias.dia.fi.upm.es">webode@delicias.dia.fi.upm.es</a>
<b>Compatibilidad con otros sistemas</b>
<p>WebODE proporciona una API orientada al servicio que facilita la integración con otros sistemas. Las ontologías pueden ser fácilmente integradas en otros servidores de ontologías. Las distintas posibilidades de interoperatividad incluyen:</p> <ul style="list-style-type: none"><li>• API de acceso a ontologías WebODE accesible desde otras aplicaciones usando RMI.</li><li>• Las ontologías WebODE pueden ser exportadas e importadas en XML usando un DTD (<a href="http://webode.dia.fi.upm.es/webode/dtd/webode_2_0.dtd">http://webode.dia.fi.upm.es/webode/dtd/webode_2_0.dtd</a>)</li><li>• WebODE genera todos los conceptos como beans de Java, con información incluso de sus atributos y relaciones ad-hoc.</li></ul>
<b>Utilidad / facilidad de uso</b>
La herramienta resulta intuitiva de utilizar, y además posee gran cantidad de documentación relacionada

**Tabla 29: Características de WebODE**

### 8.3 Comparación de herramientas

#### 8.3.1 Comparación de herramientas de anotación externas

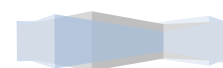
NOMBRE	NAVEGADOR	ALMACENAMIENTO	PERMITEN CLASIF.	CREAR NUEVAS SUBCLASES	MÁS DE UNA ANOTACIÓN POR WEB	ENLAZAR ANOT.	ESTÁNDARES
COHSE	Mozilla 1.0.2 y con Internet Explorer superior o igual a 5.X.	mysql	Sí, en base a conceptos de ontología.	No	Sí	Sí	DAML+OIL,OILEd, cualquier constructor de ontologías que soporte DAML
AMAYA	Mozilla o Internet Explorer	RDF Genéricas	Sí	No	Sí	No	RDF, y protocolo Xpointer
ANNOZILLA	Mozilla y también los basado sen Annotea (Amaya o <i>plugin</i> snufkin)	RDF Genéricas	Sí	No		No	RDF, y protocolo Xpointer
YAWAS	Microsoft Internet Explorer y Netscape Navigator	PC local del autor	Sí	No	Sí	No	URLs extendidas, para anotar lenguaje XML Pointer.
ANNOTATION SYSTEM	Microsoft Internet Explorer	PC local del autor o BBDD servidor central	Sí	No. Sin embargo, si el usuario no encuentra una categoría que se adecue a su anotación, puede añadir una nueva categoría a las preexistentes	Sí	No	RDF sobre XML
TRELLIS	Internet Explorer 5.5+ soporta. Parcialmente navegadores como Netscape 6.x y Mozilla 1.x	Servidor en un directorio no compartido	Sí, en base a sentencias creadas por el usuario	Sí	Sí	No	XML, RDF, DAML+OIL y OWL

Tabla 30: Comparativa de herramientas de anotación externas (I)



NOMBRE	CÓDIGO	COMPATIB.	COMPARTICIÓN	FACILIDAD DE USO	PERMITE IMAGEN	PERMITE ABRIR DEBATES	LENGUAJES Y POSIBLES FORMATOS DE IMP. Y EXP.
COHSE	abierto para fines no comerciales	No	Todos los usuarios	amigable	No	No	DAML+OIL - RDF
AMAYA	abierto?	Sí	Todos los usuarios	?	No	No	Navegadores basados en el mismo (no permite imp. y exp.)
ANNOZILLA	?	Sí. Navegador Amaya y en Internet Explorer a través del <i>plug-in</i> snufkin.	Todos los usuarios	?	No	No	RDF (fácil imp. y exp.)
YAWAS	abierto	Sí	Nivel de privacidad a elección del usuario	amigable	No	No	Aplicación Java y Javascript (fácil imp. y exp.)
ANNOTATION SYSTEM	Indefinido	Sí	Todos los usuarios o sólo para un grupo característica adicional, provisión de categorización	No es demasiado complicada	No	Sí	RDF (fácil imp. y exp.)
TRELLIS	Para poder emplear el software de Trellis fuera de EEUU es necesario tener una licencia de control de exportación del gobierno de EEUU.	Sí (funciona a través de Web)	Posibilidad de ver cada usuario sus anotaciones o buscar por temas las de otros usuarios	No muy complicada una vez entendidas las funcionalidades de la herramienta	No	No	RDF, DAML+OIL y OWL

Tabla 31: Comparativa de herramientas de anotación externas (II)



### 8.3.2 Comparación de herramientas de anotación de autor

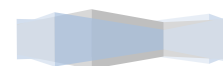
NOMBRE	NAVEGADOR	ALMACENAMIENTO	PERMITEN CLASIF.	CREAR NUEVAS SUBCLASES	MÁS DE UNA ANOTACIÓN POR WEB	ENLAZAR ANOT.	ESTÁNDARES
MnM	Gráficamente sólo con MnM. Como marcas XML en cualquier editor de texto	junto al documento anotado en formato XML en un archivo nuevo de tipo texto, HTML o XML en el ordenador local	Sí	No	Sí	Sí	ontologías: OCML de WebOnto, RDF y DAML+OIL, documentos con anotación en formato texto, HTML o XML con etiquetas XML. se requiere Java 1.4.1 y Ant 1.5.1 (o superior).
ONTOMAT-ANNOTIZER	Cualquier editor de texto (como marcas XLM)	en el documento anotado en formato XML en un archivo local	Sí	Sí	Sí	Sí	lenguaje DAML+OIL (basado en RDF)
SHOE KNOWLWDGE ANNOTATOR	Netscape 3,0+ o 4,0+, Internet Explorer 4,0, WINDOWS 95 y NT, SunOS	en el propio código HTML	Sí	Sí	Sí	Sí	Java Development Kit, Java Runtime Environment
SMORE	Netscape, Internet Explorer	archivo separado o en propio código HTML	Sí	Sí	Sí	Sí	JDK 1,4
MELITA	Gráficamente sólo con Melita. Como marcas XML en cualquier editor de texto	en el documento anotado en formato XML en un archivo local	Sí	No	Sí	Sí	lenguaje XI (basado en PROLOG)
GATE	gráficamente sólo con GATE	BBDD (Oracle o PostgreSQL) o archivo local empleando serialización de Java)	Sí	Sí	Sí	Sí	Java con Java beans y requiere Java 1,4

Tabla 32: Comparativa de herramientas de anotación autor (I)



NOMBRE	CÓDIGO	COMPATIBILIDAD	COMPARTICION	FACILIDAD DE USO	PERMITE IMAGEN	PERMITE ABRIR DEBATES	LENGUAJES Y POSIBLES FORMATOS DE IMP. Y EXP.
MnM	Abierto para fines no comerciales	Proporciona APIs para desarrollar otras htas de extracción de conocimiento.	Todos los usuarios	Amigable	No	No	RDF, DAML+OIL y OCML de WebOnto
ONTOMAT-ANNOTIZER	?	Sí	Todos los usuarios	Amigable	No	No	DAML+OIL - RDF (no permite imp. Y exp.)
SHOE KNOWLWDGE ANNOTATOR	Abierto	Sí, XML	Nivel de privacidad según gestión de permisos	No demasiado complicado	No	No	SHOE&SGML, SHOE&XML, SHOE&DAML
SMORE	Abierto	Sí, trabaja con Ekit y ImageJ API PhotoSMORE, y el API de Michael Grove's RDF Parser.	Todos los usuarios	No demasiado complicado	No	No	RDF (no tiene opciones imp. y exp.)
MELITA	No, disponible libremente	No	Todos los usuarios	Amigable	No	No	Lenguaje XI (basado en PROLOG) (No tiene opciones de imp. y exp.)
GATE	Abierto	Sí, DAML+OIL	Se puede elegir	Amigable	No	No	DAML+OIL. Además, se puede trabajar con RDF empleando el módulo Ontogazetteer, que es de pago.

Tabla 33: Comparativa de herramientas de anotación autor (II)



### 8.3.3 Comparación de herramientas de creación de ontologías

NOMBRE	LICENCIA	ARQUITECTURA SOFTWARE	EXTENSIBILIDAD	GESTIÓN DE BACKUPS	ACCESO	ALMAC. DE LA ONTOLOGÍA
<b>WebODE</b>	Uso libre tras registrarse	Tecnologías usadas: Java, RMI, CORBA, XML y Minerva Application Server	Escalable. Capaz de editar ontologías de diversos dominios	SI	Vía web a través de navegador	En BBDD compatible con Java
<b>ONTOLINGUA</b>	Uso libre tras registrarse	Aplicación cliente/servidor. 100% Java	Flexible y escalable	NO	Vía web a través de navegador	En el servidor
<b>ONTOEDIT PROFESSIONAL</b>	Bajo licencia. Versión gratis limitada a 50 conceptos, relaciones e instancias	Standalone	A través del uso de <i>plug-in's</i>	NO	No proporciona acceso via web. Para acceder al conocimiento habría que diseñar aplicación que accediera a BBDD.	OXML
<b>LINK FACTORY</b>	No gratuita	Aplicación cliente/servidor. 100% Java	Flexible y escalable. Probado en dominios médicos y otros.	NO	Vía web a través de navegador	En BBDD compatible con Java
<b>OiEd</b>	Open Source bajo GPL	Standalone	NO	NO	RDF URIs, nombres de dominio limitados, XML Schema muy limitado	En ficheros de extensión .daml
<b>PROTÉGÉ</b>	Gratis bajo licencia MPL	Standalone	A través del uso de <i>plug-in's</i>	NO	A través de servlets o applets	En ficheros de texto, RDF Schema o en cualquier BBDD compatible con JDBC.

Tabla 34: Comparativa de herramientas de creación de ontologías (I)





NOMBRE	ALMAC. EN LA BBDD	TAMAÑO	CARACTERÍSTICAS DEL MODELO	LENGUAJE BASE	PLATAFORMA EJECUTARSE	FORMATOS DE IMP/EXP
WebODE	BBDD relacionales accesibles por JDBC	<=23755 términos	Basado en un motor de deducciones y un constructor de axiomas	Ontolingua que luego es traducido a LOOM, pure- KIF, Prolog,...	Basado en Java, con lo cual es multi-plataforma	Importación: WebODE, XML, X-CARIN Exportación: WebODE, XML, FLOGIC, RDFS, OIL, X-CARIN, DAM+OIL
ONTOLINGUA	BBDD relacionales	1000 términos	Incluye relaciones de ontologías, dependencias circulares y el perfeccionamiento polimórfico	Ontolingua/KIF	Basado en Java, con lo cual es multi-plataforma	Compatible con: KI, Ontolingua, OKBC, LOOM, EPIKIT y GENERIC-FRAME
ONTOEDIT PROFESSIONAL	BBDD relacionales accesibles por JDBC	Sin datos	Permite herencia múltiple y jerarquías de relaciones. A través de <i>plug-in</i> soporta mappings.	F-Logic	Basado en Java, con lo cual es multi-plataforma (Java 1.4)	DAML+OIL, RDF Schema, OXML, Frame-Logic, base de datos JDBC
LINK FACTORY	BBDD relacionales accesibles por JDBC	1,5 millones de conceptos	Modelos de lógica de descripción T-Box y A-Box. Soporta herencia múltiple de conceptos y relaciones.	Lógica de descripción extendida	Multi-plataforma. Probado con éxito en Windows, Solaris y Linux.	DAML+OIL, RDF Schema, XML
OilEd	NO	No soporta el desarrollo de ontologías a gran escala	Axiomas restrictivos de DAML, same-classes, tipos de datos XML Schema limitados, creación de metadatos.	DAML+OIL	Basado en Java, con lo cual es multi-plataforma (Java 1.4)	RDFS y SHIQ
PROTÉGÉ	BBDD relacionales accesibles por JDBC	150.000 clases e instancias	Concepto de herencia múltiple y jerarquías de relación, metaclasses, axiomas con restricciones Prolog, Flogig, OILy PAL mediante <i>plug-in</i> 's.	Open Knowledge Base Connectivity(OKBC)	Basado en Java, con lo cual es multi-plataforma (Java 1.3)	RDFS, XML Schema, RDB Schema a través de <i>plug-in</i>

Tabla 35: Comparativa de herramientas de creación de ontologías (II)

NOMBRE	INTERFAZ GRÁFICA	CONSISTENCIA	SOPORTE MULTI-USUARIO	SOPORTE COMPARACIÓN	SOPORTE LÉXICO	COMPATIBILIDAD OTROS SISTEMAS	UTILIDAD
WebODE	Basada en formas	Realiza chequeos usando: restricciones de clases, de valores numéricos, verificación de consistencia taxonómica, etc.	Entorno multiusuario que permite un uso sincronizado de los mecanismos.	El servicio ODEMerge permite mezclar dos ontologías en el mismo dominio	Permite definir sinónimos y acrónimos para conceptos, atributos, relaciones, etc.	Posibilidades de interoperatividad: API de acceso por RMI, exportación/importación en XML, uso de tecnología beans de Java	Muy intuitiva y con mucha documentación
ONTOLINGUA	La interfaz consta de dos marcos. Usa un navegador gráfico.	Validaciones de consistencia en tiempo real durante el proceso de edición/creación.	Entorno multiusuario con posibilidad de gestión de usuarios y grupos	Incorporación opción para comparar ontologías	Browser ontologías para búsqueda de términos	SI	Interfaz sencilla e intuitiva. Ayuda y tour guiado
ONTOEDIT PROFESSIONAL	Representación gráfica de los conceptos e instancias como nodos y las relaciones como arcos entre nodos.	A través del <i>plug-in</i> OntoBroker	Permite el bloqueo de objetos.	De forma manual mediante mappings.	Si. A través de <i>plug-in's</i>	SI	Muy intuitiva. Muy destacable la visualización gráfica de la ontología.
LINK FACTORY	Interfaz gráfica basada en beans. Visualización de ontologías en árbol	Usa chequeos mediante: reglas de restricción, disyunciones formales, etc.	Entorno multiusuario con posibilidad de gestión de usuarios.	Compara y enlaza ontologías a través de una ontología principal.	Distinción estricta entre concepto-término, descripción del lexema.	Compatible con DAM+OIL y RDFS.	Muy intuitiva. Visualización en forma de árbol.
OilEd	Visualización de clases y propiedades globales a través de ficheros tipo Graphviz	A través del razonador FaCT	NO	NO	Uso limitado de sinónimos	Compatible con RDF, RDFS, GO, OIL Text, SHIQ Knowledge Bases	Muy fácil e intuitivo de usar.
PROTÉGÉ	Visualización de clases y propiedades globales a través de ficheros tipo Graphviz	Mediante <i>plug-in's</i> .	En fase de desarrollo. Soporte multiusuario para lectura solo	Semi-automatizado mediante Anchor-PROMPT.	WorNet <i>plug-in</i> , emparejamiento de caracteres especiales	SI	Muy fácil e intuitivo de usar. Los datos se introducen mediante formularios

Tabla 36: Comparativa de herramientas de creación de ontologías (III)



## 9 Estudio de las Aplicaciones en uso de la Web Semántica

### 9.1 FOAF (Friend-of-a-Friend)

#### 9.1.1 ¿Qué es FOAF?

FOAF (Friend Of A Friend) es un proyecto dentro del entorno de la Web Semántica para describir personas, vínculos entre ellas, y cosas que hacen o crean. Se trata de un vocabulario que permite disponer de información personal de forma sencilla y simplificada para ser procesada, compartida y reutilizada. Es un lenguaje de definición basado en RDF, utilizado para la definición de datos sobre redes sociales. De esta forma se pueden definir los datos de una persona: nombre, apellidos, dirección, correo, intereses, personas que conoce, páginas que le gustan, etc.

Se muestra a continuación un ejemplo de fichero FOAF:

```
<?xml version="1.0" standalone="yes"?>
<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/">
  <foaf:Person>
    <foaf:name>Ricardo Garcia</foaf:name>
  </foaf:Person>
</rdf:RDF>
```

Fig. 19: Ejemplo de codificación en FOAF

FOAF permite la compartición de información sobre personas, así como facilita el acceso y recuperación de información sobre las mismas de forma automatizada mediante búsquedas. Gracias al uso de metadatos y XML se pueden crear de forma automatizada directorios de personas, listas de correo, etc. y realizar búsquedas sobre esta información.

## 9.1.2 Identificación de personas

---

Existen en el mundo miles de personas que comparten los mismos atributos, nombre, color de ojos, pelo, altura, intereses, etc. Sabiendo esto, ¿se puede llegar a distinguir de quien se habla? ¿Cómo identificar de un modo unívoco a alguien? Pues bien, como FOAF es un proyecto basado en Internet, los desarrolladores buscaron algún atributo propio de los usuarios de Internet, y lo encontraron: el email. Citando el documento que describe FOAF (en inglés):

*FOAF is expressed as an RDF Schema, annotated with DAML to express the fact that a foaf:mbox uniquely picks out an individual. FOAF:mbox is an UnambiguousProperty, in the DAML+OIL sense. Furthermore it is a 'static unambiguous property', in that it picks out at most a single individual, across time and change.*

De manera más informal: el email es considerado una propiedad no ambigua acerca de una persona, en el sentido que toma a lo sumo una sola persona como destinatario, a lo largo del tiempo. Esto podría parecer poco cierto, si se sabe que existen emails corporativos o que se distribuyen a varias personas a pesar de tener una sola dirección, pero también es cierto que cualquier puede obtener un email propio en los cientos de servidores que brindan cuentas de email gratuitas. Así que, en definitiva, el email funciona (y de hecho lo hace muy bien) como un identificador único de alguien.

Ahora ya se sabe cómo identificar a una persona, pero ¿cómo indicarlo en un archivo FOAF? He aquí un ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<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/">
  <foaf:Person>
    <foaf:name>Leandro Mariano Lopez</foaf:name>
    <foaf:mbox rdf:resource="mailto:yo@inkel.com.ar"/>
  </foaf:Person>
</rdf:RDF>
```

Fig. 20: Ejemplo de identificación de personas en FOAF

Se hace notar como se que indica el email como un recurso de RDF. Esto es así debido a que en el mundo de RDF (y por extensión de FOAF) hablamos de recursos (RDF, Resource Description Framework), y estos se indican por medio de URIs.

Bien, ya tenemos una forma de identificar unívocamente a una persona, a través de su dirección de correo electrónico. Y no solo eso, sino que podemos indicar todos los emails correspondientes a esa persona, dado que todos son identificadores únicos y exclusivos de ella. Bienvenido al interesante y prometedor proyecto FOAF.

Las ventajas que a futuro presenta FOAF es la rápida y efectiva constitución de comunidades virtuales, búsqueda de personas con los mismos intereses que nosotros, documentos escritos por otros, entre otros, y teniendo la tranquilidad de saber que lo que aún no esta hecho, se puede hacer: tan solo es cuestión de definir un esquema, utilizarlo y motivar a otros a que lo hagan.



### 9.1.3 Extendiendo FOAF

---

#### **FOAF People Map**

Presenta un mapa completo de la Tierra ubicando a todos aquellos usuarios que completaron sus datos de nearestAirport en la latitud y longitud correspondiente. De aquí se linkea a foafnaut! para poder ver más información del usuario. Necesita tener un navegador con soporte de SVG (ver foafnaut para mayor información).

#### **foafnaut!**

Una de las mejores herramientas de la actualidad, permite ver un usuario y todos aquellos a quien conoce o quienes lo conocen, e ir armando una "red" entre usuarios. Permite también ver cuál es el "camino más corto" entre dos usuarios distintos. Actualmente se encuentra funcionando normalmente, pero no se pueden agregar nuevos usuarios ni modificar los existentes. Vale la pena probarlo.

Para poder visualizar este sitio se debe tener instalado soporte de SVG en el navegador.

#### **Foaf Explorer**

Foaf Explorer permite ver la información contenida en un archivo FOAF sin necesidad de instalar ningún plugin ni enviar el archivo para ser procesado. Todo lo hace en tiempo real, basándose en PHP y XSLT.

#### **FOAF Web View**

FOAF Web View muestra los datos de una persona de modo similar a Foaf Explorer, solo que en este servicio se debe enviar primero la URL del archivo FOAF que se desea visualizar, ya que los datos son almacenados y procesados en el servidor.

#### **DOAC**

DOAC (Description Of A Career) es una extensión del vocabulario FOAF que permite compartir un curriculum de forma que pueda ser procesado por cualquier aplicación. Al estilo de un Curriculum Vitae clásico, incluye información sobre educación, experiencia laboral, publicaciones, idiomas y otras habilidades.

#### **FOAF-a-Matic**

Una consecuencia de FOAF es FOAF-a-Matic, una aplicación Javascript que permite crear una descripción FOAF de una persona. Con esta descripción, las señas personales pasan a integrar un motor de búsqueda dónde será posible obtener información de una persona en concreto y de las comunidades a las que pertenezca, de una forma rápida y sencilla.



## 9.2 *Sindicación de los contenidos Web*

---


Se puede definir la sindicación Web como el proceso para hacer que ciertos datos de un sitio Web estén disponibles para su acceso, transmisión, agregación o publicación a la red. Esto puede significar simplemente dar licencia del contenido de estos datos para que se pueda utilizar, o hacer disponible el contenido de un sitio Web (*WebFeed*) para que se pueda visualizar una lista de contenidos que se encuentre actualizada.

Es en este último caso, el uso de los *Web Feed*, donde se centra este apartado. Así se explica qué es una *Web Feed*, también conocidos como *Syndications Feeds*, qué herramientas para sindicación existen, y qué son exactamente, y cómo podemos utilizar, unos programas denominados *agregadors* o *feed readers*.

### 9.2.1 ¿Qué son los Syndications Feeds?

---

Los *Syndication feeds* se han convertido en una herramienta estándar de la Web. Es fácil encontrarse con un pequeño botón naranja marcado con unas letras blancas que dicen XML, o quizás Atom, RSS 2.0, RSS 1.0 o, incluso, Syndication o Feed.

También podríamos reconocer este botón: 

Todos estos son ejemplos de *syndication feeds*: son archivos con contenido en formato XML, y que muestran un conjunto de los ítems más recientemente publicados ordenados cronológicamente, empezando por el más reciente y acabando por el más antiguo.

Uno de los formatos de feed más común es el RSS 2.0. El RSS 2.0 es un formato XML, considerado altamente estable, que permite extensiones exclusivamente a través de los namespaces.

La especificación RSS 2.0 es mantenida actualmente por la Universidad de Harvard, y la podemos encontrar en la Web <http://blogs.law.harvard.edu/tech/rss>. Aunque la especificación está actualmente parada, hay una licencia *Creative Common* que permite a cualquier desarrollador escribir código con este formato. Un consejo consultor de tres personas toman todas las decisiones referentes al status de la especificación.

Una versión anterior al RSS 2.0 es el formato RSS 0.91 que se utiliza ampliamente en varios lugares Web.

El otro de los formatos RSS más importante es el RSS 1.0, el cual es mantenido por una organización autónoma de desarrolladores que mantienen su trabajo en una lista de mail de Yahoo Groups localizada en <http://groups.yahoo.com/group/rssdev/>. La especificación se ha mantenido estable por varios años, y se puede localizar en <http://Web.resource.org/rss/1.0/>.

El último formato principal se denomina Atom (<http://atom-enabled.org/>). Fue creado a partir del uso de una wiki con toda la gente interesada invitada a participar y contribuir. Atom se encuentra bajo el auspicio del Internet *Engineering Task Force (IETF)*, una organización internacional de estándares.

Hay otras variaciones y versiones de feeds, pero estas cuatro (RSS 2.0, RSS 0.91, RSS 1.0 y Atom) son las más comúnmente usadas.



## 9.2.2 El apoyo de la industria

---

Aunque los feeds se han extendido ampliamente por diferentes clases de lugares Web, han llegado a su nivel de popularidad actual debido a su uso dentro del fenómeno del *Weblogging*. Los *Weblogs*, o simplemente blogs, son lugares Web periódicamente actualizados que recopilan cronológicamente textos o artículos de uno o varios autores dónde el más reciente aparece primero, con un uso o temática particular, y permiten siempre al autor publicar el que quiera con total libertad. Habitualmente están escritos con un estilo personal e informal.

Todas las herramientas de creación y mantenimiento de *Weblogging* proporcionan funcionalidades para generar feeds en uno o más formatos. Algunas de estas herramientas son:

### **Blogger:**

El más anciano de todos los entornos de hospedaje de *Weblogs*, Blogger todavía responde manteniendo un porcentaje significativo de *Weblogs*. Originariamente era una entidad separada, pero ahora es propiedad de Google. Blogger proporciona sólo Atom como tipo de feed predeterminado aunque si se tiene una cuenta, se puede también seleccionar RSS 2.0. Actualmente RSS 1.0 no está soportado.

<http://www.blogger.com/start>

### **Six Apart:**

Ofrece tres herramientas *Weblogging*, la más popular conocida como *LiveJournal*, que sólo soportan Atom y RSS 2.0.

<http://www.sixapart.com/>

<http://www.livejournal.com/>

### **WordPress:**

WordPress proporciona apoyo para los tres formatos de feeds más importantes (RSS 1.0, RSS 2.0 Y Atom), así como algunos formatos heredados tales como RSS 0.9x. <http://wordpress.org/>

### **ExpressionEngine:**

Ofrece soporte para los tres *syndication feeds* más importantes.

<http://www.expressionengine.com/>

### **Blosxom:**

Blosxom tiene varios plugins que soportan RSS 1.x, RSS 2.0, Atom, y el formato RSS 3.0, que es un feed no XML puramente textual que fue creado más con finalidades didácticas que para un uso real.

<http://www.blosxom.com/>

### **Drupal:**

Se puede considerar considerarlo más un completo Content Management System (CMS), sistema de gestión de contenidos, que una herramienta de Weblogs. Drupal proporciona apoyo por Atom, RSS 1.0, y RSS 2.0, así como para RSS 0.91 (en su implementación principal).

<http://drupal.org/>

### 9.2.3 Descubriendo los feeds

---

Como ya se mencionó, aquellos enlaces, botones de color naranja, etiquetados como Atom, RSS o XML, todos conducen a un *syndication feed*. Pinchando sobre estos botones se abrirá este feed en el navegador (o la aplicación que se haya definido para gestionar este feed). La herramienta de publicación añade ésta a su sitio Web, o se puede añadir fácilmente, creando un enlace de hipertexto con la etiqueta apropiada.

Para facilitar la suscripción al sitio Web se pueden también incluir botones y enlaces hacia los agregadores (o lectores de feeds).

Los agregadores, *feed readers* o *aggregators*, son un tipo de software que reducen el esfuerzo y el tiempo utilizado para verificar regularmente los lugares webs preferidos (suscritos) buscando las actualizaciones de los mismos, creando un solo espacio de información, a la manera de un “tablero de noticias personal”. Los agregadores facilitan la suscripción a los feeds.

La mejor manera por aproximarse a la publicación de un feed es utilizar el método *autodiscovery*. Usar el *autodiscovery* implica poner una simple línea dentro la línea HEAD de cada documento web, y apuntar la referencia a la localización del feed que lo contiene. También se ofrece información respecto al tipo de feed, de forma que los agregadores sepan como encontrarlo. Cuando los lectores acceden a la página con esta línea con un navegador sensible a la sindicación feed se ve un indicador que señala que el sitio Web tiene uno, o más de uno, feed asociado y que pueden suscribirse. Si los lectores ponen la URL del sitio Web dentro del agregador que usen, este encontrará de nuevo el enlace al feed, sin ningún esfuerzo.





## 9.2.4 Feeds: cómo suscribirse y leerlos

---

Igual que las herramientas para publicar, y los formatos para syndicar, se tienen muchas elecciones para escoger qué clase de agregador feed usar. Se puede utilizar una herramienta basada en navegador -como FireFox, Safari u Opera-, entonces el feed aparece de una manera similar a un marcador. O bien podemos usar agregadores basados en web -integrados dentro de portales, como My Yahoo! o Google -, o de escritorio -integrados al correo como Mozilla Thunderbird-, o por aparatos de telefonía móvil, etc.

Se pueden encontrar una extensa lista de agregadores en el sitio Web <http://www.newsonfeeds.com/faq/aggregators>. Algunas herramientas sólo se encuentran disponibles para algunos sistemas operativos y otras requieren que software adicional se encuentre instalado.

Un agregador de escritorio permite configurar el tiempo de comprobación de las verificaciones de las actualizaciones de los feed. Una opción común es hacerlo cada hora: hacerlo más frecuentemente es considerado como "*malos modales*", puesto que añade tráfico innecesario al servidor del feed.

Una vez instalada la herramienta y configurada, ya sólo hace falta suscribirse a los sitios Web que se encuentren más interesantes: desde weblogs personales hasta las publicaciones más importantes, como el *New York Times*.

Si se escoge un agregador basado en Web se tendrá la ventaja de que no se debe instalar ningún software. La desventaja es que no se tiene acceso a los feeds cuando se está desconectado.



### 9.2.5 Nuevos retos para la sindicación Web

---

No siendo más que un simple concepto, los *syndication feed* han generado un enorme e intenso interés a su alrededor en los últimos años. Una de las razones es la introducción de los *podcasting*.

El *podcasting* consiste en hacer una grabación de una retransmisión, y entonces colgar esta grabación a la red. Los sitios Web de *podcasting* pueden también ofrecer una descarga del archivo directamente, pero el feed a que se puede suscribir que automáticamente genera nuevo contenido, y es lo que distingue un *podcast* de un simple lugar de descargas o de *streaming* en tiempo real.

La esencia del *podcasting* es crear contenido (audio o vídeo) para una audiencia que quieren ver y sentir en tiempo real lo que desean.

Se encuentra el fenómeno feed en otros ámbitos, como por ejemplo:

- Muchas organizaciones de noticias, incluyendo Reuters, la CNN y la BBC. Estas organizaciones permiten a otros websites incorporar su "titular sindicado" o línea de cabecera que hace de corto resumen, bajo algunas restricciones de uso.
- RSS es actualmente usado para varios propósitos, incluyendo Marketing, informes o cualquier otra actividad relacionada con publicaciones o actualizaciones periódicas.
- Muchas corporaciones están escogiendo RSS o Atom para publicar sus noticias y sustituir email.

El hecho es que muchos consumidores y periodistas pueden actualmente disponer al instante de las noticias más recientes en vez de tener que buscarlas: los agregadores verifican las listas en formato RSS o Atom y muestran cualquier actualización de los artículos que encuentran.



## 9.3 Análisis de RSS y Atom

---

### 9.3.1 RSS

---

La abreviatura RSS se refiere a uno de los siguientes estándares:

- Rich Site Summary (RSS 0.91)
- RDF Site Summary (RSS 0.9 y 1.0)
- Really Simple Syndication (RSS 2.0)

Ciertas deficiencias en el formato RSS 2.0 motivaron el desarrollo del formato *Atom*, el cual está también basado en XML.

El formato y el uso de los *syndication feed* es bastante sencillo. Cada feed contiene un cierto número de items individuales, normalmente con un *title* (título), *last update* (última actualización), *update frequency* (frecuencia de actualización), *site owner* (propietario del lugar Web), y un largo etcétera. Dentro de estos campos simples, hay una gran variedad de opciones que conducen a una enorme diferenciación entre feeds. Por hacerlo más comprensible, se analizará sucintamente cada uno de los formatos de los feeds para ver qué campos son compartidos y cuales son específicos para cada formato.



### **Elementos Container**

El feed “container” es información de todo el sitio Web, consistiendo en varios campos que se encuentran listados dentro del feed. Incluidos dentro de este container se encuentran los siguientes campos que se repiten a través de la mayoría de *syndication feeds*:

<b>Campo</b>	<b>Descripción</b>
<i>link</i>	Se trata de un enlace hacia la URL del lugar web
<i>title</i>	Es el título del sitio Web
<i>description</i>	Es una descripción del sitio Web, y habitualmente contiene el subtítulo del lugar. En Atom es el campo subtitle
<i>autor</i>	En Atom, author es una estructura que contiene el nombre del autor y su email, o sólo el mail del autor. En RSS 2.0 este campo es cambiado por webMaster y managingEditor, las cuales son direcciones mail. En RSS1.x, normalmente se trata por dc:creator, el cual puede tener o bien una estructura o un valor simple
<i>date</i>	En RSS 1.x, este campo habitualmente dc:dato es la fecha de cuando el feed fue actualizado. En Atom es el campo update; en RSS 2.0, este es lastBuildDate
<i>generator</i>	En RSS 2.0, esta es la herramienta utilizada para generar el syndication feed. El feed Atom también utiliza generator, mientras que RSS 1.x usa generatorAgent
<i>copyright</i>	Información del copyright
<i>language</i>	En qué lengua se encuentra el documento
<i>id</i>	Camp específic per Atom, que proporciona un identificador únic pel lloc web
<i>image</i>	Un icono o imagen representando el feed o el lugar web tanto por RSS 1.x como por RSS 2.x. En Atom esto puede ser un icono aunque logo puede ser usado por una representación del logo. En RSS 1.x y RSS 2.x image es una estructura apuntando a una URL con la imagen, el título y un enlace al lugar web.

**Tabla 37: Características de container**



### Elementos Entry

Cada *entry* entrada individual de un sitio Web tiene su propia *entry* dentro del feed hasta el número designado por cada feed (habitualmente se encuentra entre 5 y 15). En Atom las *entry* se encuentran designadas por elementos entry, mientras que en RSS se listan como item.

Campo	Descripción
<i>link</i>	Se trata de un enlace al item individual
<i>title</i>	Es el título del item, si tiene
<i>description</i>	En RSS, <i>description</i> es un resumen o descripción del item. En Atom es el campo <i>summary</i>
<i>autor</i>	En Atom, <i>author</i> es una estructura que contiene el nombre del autor y su email, o sólo el mail del autor. En RSS se trata por <i>dc:creator</i>
<i>guid</i>	Proporciona un identificador único por el item. En Atom, es <i>id</i>
<i>content</i>	Es el completo contenido codificado del item incluyendo cualquier etiqueta de marcaje HTML o XHTML. También puede ser denominado <i>content:encoded</i> en RSS 1.x.
<i>pubDate</i>	Fecha de publicación del item. Esto es <i>dc:creator</i> en RSS
<i>category</i>	La categoría del <i>item</i>

Tabla 38: Características de Entry

### Resumen de la especificación Rich Site Summary (RSS 0.91)

Etiqueta	Descripción	Subelementos obligatorios
<code>&lt;channel&gt;</code>	Información respecto a un canal particular. Todo lo que pertenezca a un canal se encuentra dentro de esta etiqueta	<i>description</i> <i>language</i> <i>link</i> <i>title</i>
<code>&lt;description&gt;</code>	Texto con la descripción de un ítem, canal, imagen o entrada de texto	
<code>&lt;language&gt;</code>	Especifica el lenguaje de un canal	
<code>&lt;link&gt;</code>	URL donde un usuario puede navegar	
<code>&lt;pubDate&gt;</code>	Fecha de cuando el canal fue publicado	
<code>&lt;rss&gt;</code>	Identifica el inicio y el final del contenido RSS. Es obligatorio el uso del atributo <code>version="0.91"</code>	<i>channel</i>
<code>&lt;title&gt;</code>	Una cadena de texto identificadora del recurso. Cuando se utiliza con un canal es el propio título	

Tabla 39: Características Rich Site Summary (RSS 0.91)

**Resumen de la especificación RDF Site Summary (RSS 1.0)**

Etiqueta con la sintaxis siguiente	Descripción	Subelementos obligatorios
<code>&lt;channel rdf:about="{recurs}"&gt;</code>	Información de un canal particular, incluyendo un título, una breve descripción y un link URL al canal descrito	<i>description items link title</i>
<code>&lt;description&gt;</code>	Texto con la descripción del contenido, función, origen etc de un canal	
<code>&lt;items&gt; &lt;rdf:Seq&gt; &lt;rdf:li resource={item_uri}"/&gt; .... &lt;/rdf:Seq&gt; &lt;/items&gt;</code>	Una tabla RDF de contenidos, asociando los ítems del documento con este canal RSS en particular	
<code>&lt;link&gt;</code>	URL que enlaza con el sitio Web HTML relacionado con el canal	
<code>&lt;rdf:RDF xmlns:rdf="{recurs}" xmlns="{recurs}"&gt;</code>	Identifica el inicio y el final del contenido RSS	<i>channel</i>
<code>&lt;title&gt;</code>	Un título descriptivo de un canal	

**Tabla 40: Características RDF Site Summary (RSS 1.0)**

**Resumen de la especificación Really Simple Syndication (RSS 2.0)**

Etiqueta	Descripción	Subelementos obligatorios
<code>&lt;channel&gt;</code>	Información respecto a un canal particular. Todo lo que pertenezca a un canal se encuentra dentro de esta etiqueta	<i>description link title</i>
<code>&lt;description&gt;</code>	Texto con las descripción de un canal	
<code>&lt;link&gt;</code>	URL que enlaza con el sitio Web HTML relacionado con el canal	
<code>&lt;pubDate&gt;</code>	Fecha de cuando el canal fue publicado	
<code>&lt;rss&gt;</code>	Identifica el inicio y el final del contenido RSS. Es obligatorio el uso del atributo <code>version="2.00"</code>	<i>channel</i>
<code>&lt;title&gt;</code>	Una cadena de texto identificadora del recurso. Cuando se utiliza con un canal es el propio título	

**Tabla 41: Características Really Simple Syndication (RSS 2.0)**



### 9.3.2 Atom

Atom 1.0, presentado en agosto del 2005, es un formato de documentos basado en XML que describe listas de información relacionada -conocido como feeds-. Los Feeds se encuentran compuestos de items conocidos como *entries* (entradas) cada una con un grupo extensible de Metadatos adjuntos. Por ejemplo, cada entrada tiene un *title* (título).

Existen dos clases de documentos Atom: los *Atom Feed Documents* y los *Atom Entry Documents*. Un *Atom Feed Document* es una representación de un *feed Atom*, incluyendo Metadatos del feed y algunas o todas las entradas asociadas con él. Su raíz es el elemento *atom:feed*.

Un *Atom Entry Document* representa exactamente una entry Atom fuera del contexto de un *feed Atom*. Su raíz es el elemento *atom:entry*.

Los documentos Atom deben ser XML bien formados y tienen la extensión atom.

Elements	Descripción	Subelements obligatorios
<atom:feed>	Es el elemento raíz de los documentos Atom Feed Document	<i>atom:author</i> (1 o más)* <i>atom:id</i> (exactamente 1) <i>atom:title</i> (exactamente 1) <i>atom:link</i> con el valor del atributo igual a "self" (exactamente 1) <i>atom:update</i> (exactamente 1)
<atom:entry>	Representa una entry individual actuando como contenedor por las metadatos y datos asociados a la entrada. Puede aparecer como hijo de un elemento atom:feed, o como elemento raíz de un Atom Entry Document	<i>atom:author</i> (1 o más)* <i>atom:id</i> (exactamente 1) <i>atom:content</i> (máximo 1)** <i>atom:summary</i> (exactamente 1) <i>atom:title</i> (exactamente 1) <i>atom:update</i> (exactamente 1)
*a menos que contenga un elemento atom:source que contenga un elemento atom:author o, dentro de un Atom Feed Document, el elemento atom:feed contenga él mismo un elemento atom:author ** no es obligatoria la existencia de este elemento, si no está, entonces debe tener un elemento atom:link con el valor del atributo rel igual a alternate		
<atom:content>	Contiene o enlaza con el contenido de entry	
<atom:author>	Representa el autor del entry o feed	
<atom:id>	Expresa un identificador único y permanente (International Resource Identifier IRI) para un entry o feed	
<atom:link>	Define una referencia a un recurso Web. Debe tener un atributo href el valor del cual debe ser una referencia IRI	

<code>&lt;atom:summary&gt;</code>	Texto que provee de un resumen corto, abstracto, o uno extracte de un entry	
<code>&lt;atom:title&gt;</code>	Texto que proporciona un título legible por humanos de un entry o feed	
<code>&lt;atom:update&gt;</code>	Fecha que indica la más reciente modificación de un entry o un feed siempre y cuando el publisher considere esta modificación significativa de alguna manera	

**Tabla 42: Características generales Atom 1.0**





## **9.4 La Web Semántica y las Bases de Datos**

---

Los SGBD (Sistemas Gestores de Bases de datos) son un software específico, dedicados a servir de interfaz entre las bases de datos y las aplicaciones que las utilizan. Los SGBD incorporan una serie de servicios para la gestión de bases de datos: permiten el acceso a los datos de manera sencilla, gestionan los permisos de acceso a datos para múltiples usuarios, manipulan los contenidos de las bases de datos (inserción, supresión, modificación).

### **9.4.1 ¿Para qué sirven las SGBD?**

---

En general un SGBD incluye las características siguientes:

- Independencia: Los datos deben ser independientes, de forma que el nivel físico y lógico puedan modificarse sin que esto tenga consecuencias sobre las aplicaciones finales: los cambios deben ser transparentes para los usuarios.
- Tiempos de respuesta: el sistema debe dar respuestas rápidas a las consultas, esto implica el uso de algoritmos de búsqueda eficientes.
- Redundancia mínima: hace falta evitar en la medida de lo posible la existencia de informaciones redundantes, para evitar posibles errores y aprovechar el espacio de memoria.
- Integridad: hace falta garantizar la validez de los datos y estos deben guardar una coherencia entre ellos. Hace falta gestionar correctamente los datos que hacen referencia a otros datos.
- Administración centralizada: el administrador del SGBD debe poder gestionar las diferentes bases de datos de manera centralizada.
- Control de concurrencia: hace falta permitir el acceso simultáneo a la base de datos de múltiples usuarios, evitando que se produzcan inconsistencias en la información.
- Seguridad: hacen falta mecanismos de gestión de derechos de acceso de los usuarios a los datos. Son necesarios también sistemas de recuperación de datos en caso de ataques malintencionados o errores involuntarios.

## 9.4.2 SGBD existentes

---

Se presentan a continuación los Sistemas de Bases de datos más importantes, clasificándolos según si son “open source” o no.

<b>PostgreSQL</b>
<p>Es un SGBDR pensado principalmente para sistemas de explotación de tipo Linux y Unix (AIX, BSD, HP-UX, SGI IRIX, Mac HUESO X, Solaris, SunOS, Tru64), aunque también hay una versión por Windows (la v8.0.4). Incluye la mayoría de tipos de datos SQL92 y SQL99 (integer, numeric, boolean, char, varchar, dato, intervalo, timestamp).</p> <p>También soporta la grabación de ficheros binarios de gran capacidad (hasta 1Gb) como imágenes, sonido y vídeo. Incluye también utilidades por programar en C/C++, Java, Perl, Python, Ruby, Tcl, ODBC.</p> <p>El código fuente PostgreSQL está bajo la licencia BSD (Berkeley Software Distribution). Esta licencia da libertad al usuario por utilizar, modificar y distribuir PostgreSQL; cualquier cambio realizado por el usuario es propiedad del usuario. Medida máxima de una base de datos: ilimitado. Medida máxima de una mesa: 32TB.</p>
<b>MySQL</b>
<p>MySQL se ha convertido en el SGBD más popular actualmente. Desde la versión 3.23.19, MySQL está bajo la licencia GPL (General Public License), (tanto por Linux como por Windows). Esto significa que puede utilizarse de forma gratuita, pero teniendo en cuenta que obliga a distribuir cualquier producto derivado bajo la misma licencia.</p> <p>Si un desarrollador desea incorporar MySQL en su aplicación pero no quiere distribuirla bajo licencia GPL, puede adquirir la licencia comercial. MySQL colabora a menudo con PHP (es el caso de muchos CMS que hemos visto en el capítulo 5) para la publicación de datos a Internet.</p>
<b>Firebird</b>
<p>Firebird fue creado en 1981 y trabaja sobre Linux, Windows, y la mayoría de variantes Unix. Firebird fue programado en C y C++. Se basa en la versión 6 de Interbase106 y es de licencia pública. Se trata de un SGBD relativamente pequeño pero potente, con requerimientos de hardware bajos.</p>
<b>SQLite</b>
<p>Se trata de un pequeño conjunto de librerías C capaces de gestionar bases de datos y SQL. Toda una base de datos se graba en un solo fichero. Es un gestor de licencia pública del tipo “zero-configuration”: no necesita ser administrado ni configurado. Puede almacenar ficheros binarios grandes (imágenes, vídeo, música). Es más rápido que otros gestores más populares. Capacidad máxima de base de datos: 2TB.</p>
<b>Ingres</b>
<p>SGB open source. Ofrece escalabilidad, integración y flexibilidad multiplataforma (Linux, Unix, Windows). Soporta XML.</p>

**Tabla 43: SGBD de tipo Open Source**



<b>Oracle</b>
Es considerado uno de los SGBD más cumplidos, y también el más costes (tan por el precio de la licencia como por los requerimientos de hardware). Las primeras versiones se encontraban en Unix y Windows. A partir de la versión 8.0.5 también es disponible por Linux. Durante mucho tiempo a tenido el monopolio de los servidores empresariales, aun cuando los últimos años empieza a notar la competencia de otras gestores como SQL Server de Microsoft y los SGBD open source en general.  Incluye funcionalidades cómo: servidor de aplicaciones, encriptación, desarrollo de aplicaciones, creación de Intranets, Data Warehousing, ebusiness. Incorpora herramientas para trabajar con XML.
<b>Microsoft SQL Server</b>
Es la alternativa de Microsoft en el campo de los SGBD, y por lo tanto sólo trabaja en plataformas Windows. Es un sistema costes y potente (semejante a Oracle en este sentido), que incorpora múltiples posibilidades como: apoyo XML, creación de lugares Intranet, e-Business, autoconfigurable, monitorización de eventos, escalabilidad, Data Warehousing.
<b>Paradox</b>
Base de datos relacional básicamente por entornos Windows, desarrollada actualmente por Corel e incluida en el paquete ofimática WordPerfect Office. En los tiempos del MS-DOS era una base de datos muy popular. Más tarde apareció la versión por Windows, que no pudo competir con Microsoft Access.
<b>Interbase</b>
Borland InterBase está pensado por entornos zero-configuration (como SQLite). Es lo suficientemente compacto para ejecutarse en sistemas hardwares sencillos.
<b>Informix</b>
SGBD creado por Informix y actualmente propiedad de IBM. Antes de considerarse como un gestor de carácter general para todo tipo de plataformas, estaba especializado en aplicaciones de tipos GIS (datos geográficos), Datawarehouse y Datamining. Incorpora herramientas para la publicación de datos en páginas Web.
<b>DB2</b>
También propiedad de IBM, es un SGBD claramente de carácter general, para todo tipo de plataformas. Es flexible, escalable, fácil de utilizar y con un precio ajustado. Acostumbra a trabajar en equipo con WebSphere para la implementación de portales de contenidos.
<b>Sybase SQL Server</b>
Multiplataforma. Data Mining, Warehousing, desarrollo de aplicaciones. Soporta XML y Servicios Web.
<b>Fox Pro</b>
Producto de Microsoft. La última versión proporciona componentes XML y Web Services.
<b>MS Access</b>
SGBD de Microsoft (para Windows) de carácter personal; es decir, para gestionar bases de datos

pequeñas. No es muy adecuado para accesos simultáneos ni para gestionar la seguridad de usuarios.
<b>mSQL</b>
mSQL fue creado el año 1994 por Hughes Technologies. Su nombre viene de mini SQL. Es un gestor ligero con un uso muy eficaz de la memoria. Tiene ciertas limitaciones con respecto al uso de SQL.
<b>dBase</b>
Fue el primer SGBD en un entorno MS-DOS y fue el sistema más vendido hasta la aparición de Windows. Tras ser vendido por Borland a dataBased Intelligence Inc., salió en 2005 una versión para XP que incluye soporte para ficheros XML y entorno gráfico "Drag & Drop".
<b>Filemaker</b>
Solución para Macintosh y Windows. La última versión permitía la publicación de datos en la Web y soporte de Servicios Web. Permite el intercambio de datos con XML. Permite grabar ficheros multimedia.
<b>Pervasive SQL</b>
Motor de bases de datos que ofrece: integridad de los datos, alto rendimiento, flexibilidad, escalabilidad y bajo coste.

**Tabla 44: SGBD de tipo Comercial**

En resumen, los gestores de bases de datos más utilizados actualmente en el mercado son los producidos por Oracle, IBM, Informix y Sybase, por este orden. Todos ellos, junto con de otras, soportan la generación de documentos XML a partir de consultas SQL:

- Oracle 9i: Módulo Cartridge XML. Correspondencia entre documentos XML y tablas.
- SQL Server 2000: Correspondencia entre documentos XML y tablas. Graba metadatos XML.
- Sybase: Inserción de documentos XML en tablas.
- Informix: Tipos de datos específicos para grabar XML.
- DB2: XML Extender: búsqueda en documentos XML.

Producto	Propietario
Oracle	Oracle
SQL Server 2000	Microsoft
Sybase	Sybase
Informix	IBM
DB2	IBM
MySQL	MySQL AB
SQLite	Richard Hipp
PostgreSQL	Berkeley Computer Science Department
Access 2002	Microsoft
FileMaker	FileMaker
FoxPro	Microsoft

**Tabla 45: Algunos SGBD que sí soportan XML**



### 9.4.3 Lenguajes de consulta para RDF en relación con los SGBD

Los lenguajes de consulta para RDF deberían tener en cuenta (a modo de recordatorio) las características de RDF que se muestran a continuación:

**Modelo de datos abstracto.** La estructura subyacente de cualquier documento RDF es una colección de tripletas. Esta colección se suele llamar el Grafo RDF. Cada tripleta establece una relación entre dos nodos en el grafo. Este modelo abstracto de datos es independiente de la serialización de la sintaxis. Además, los lenguajes de consulta normalmente no proporcionan habilidad para hacer consultas sobre características específicas de la serialización, por ejemplo, sobre el orden de la serialización.

**Semántica formal e inferencia.** RDF tiene una semántica formal que proporciona una base dependiente para razonar sobre el significado de un grafo RDF. Este razonamiento se suele llamar “entailment”. Las reglas de “entailment” especifican que información implícita se puede inferir de información explícita.

**Soporte para tipos de datos XML Schema.** Los tipos de datos XML se pueden utilizar para representar valores de datos en RDF. XML Schema también proporciona un marco extensible para definir nuevos tipos de datos que se puedan usar en RDF. Estos tipos de datos deberían ser soportados en un lenguaje para consultas RDF.

**Soporte para hacer sentencias libres sobre recursos.** En general, no se puede asumir que toda la información sobre cualquier recurso esté disponible en una consulta RDF. Un lenguaje de consulta debería tener esto en cuenta, y debería tolerar información incompleta o contradictoria.

Además, las siguientes propiedades de cada lenguaje también deben tenerse en cuenta:

- **Expresividad.** Indica como se pueden formular consultas potentes en un lenguaje dado. Típicamente, un lenguaje debería como mínimo proporcionar los resultados ofrecidos por el álgebra relacional. Normalmente, la expresividad se restringe para mantener otras propiedades como seguridad y para permitir la ejecución eficiente y optimizable de consultas.
- **Cierre.** Esta propiedad requiere que los resultados de una operación sean de nuevo elementos del modelo de datos. Esto quiere decir que si un lenguaje de consulta opera en el modelo de datos de los grafos, los resultados de la consulta deberían ser de nuevo grafos.
- **Adecuación.** Un lenguaje de consulta se denomina adecuado y utiliza todos los conceptos del modelo de datos subyacente. Esta propiedad además complementa la propiedad del cierre: Para el cierre, el resultado de la consulta no puede salirse del modelo de datos, para la adecuación, se necesita explotar todo el modelo de datos.
- **Ortogonalidad.** La ortogonalidad de un lenguaje de consulta requiere que todas las operaciones puedan ser usadas independientemente del contexto de uso.
- **Seguridad.** Un lenguaje de consulta se considera seguro si cada consulta que es sintácticamente correcta devuelve un conjunto de resultados finito (sobre un conjunto de datos finito)

#### 9.4.4 Análisis de los lenguajes RDF concretos

---

##### RQL

---

Para una explotación total del conocimiento almacenado en datos RDF/RDFS, se requiere un lenguaje de consulta a nivel semántico (debe ser sensible a la semántica de las primitivas RDF/S). RQL es un lenguaje de consulta declarativo para RDF/S que explícitamente captura esta semántica en su diseño. RQL fue desarrollado en el instituto ICS-FORTH, y su potencia semántica está basada en la evaluación de caminos de expresiones sobre grafos RDF. Permite el uso de variables tanto para denotar nombres de nodos (es decir, clases), como arcos (es decir propiedades). Permite consultar esquemas RDF y descripciones RDF (es decir, instancias) en una misma consulta. RQL está definido por medio de un conjunto de consultas básicas e iteradores que permiten construir otras consultas a través de una composición funcional.

RQL se apoya sobre un modelo de gráfico formal que permite la interpretación de Descripciones de recursos por medio de uno o más esquemas. La novedad de RQL reside en su habilidad para combinar con facilidad consultas de esquemas y datos mientras explota las taxonomías de etiquetas y la clasificación múltiple de los recursos. RQL sigue una sintaxis similar a OQL. RQL es ortogonal, pero no es cerrado, y las consultas devuelven datos variables en vez de grafos. Sin embargo, la semántica RQL no es completamente compatible con la semántica RDF: Hay restricciones adicionales que se aplican a los modelos RDF para permitir consultas RQL.

##### SeRQL

---

SeRQL (Sesame RDF Query Lenguaje) es un lenguaje de consulta y transformación basado en diferentes lenguajes existentes, sobre todo RQL, RDQL y N3. Sus principales objetivos de diseño, son la unificación de las mejores prácticas para los lenguajes de consulta y proporcionar un lenguaje de consultas "ligero". Su sintaxis es similar a la de RQL, aunque se han hecho modificaciones para que el lenguaje sea más fácil de procesar. Al igual que RQL, SeRQL está basado en una interpretación formal de los grafos RDF, pero la interpretación formal de SeRQL está basada directamente en el modelo teórico de RDF.

SeRQL soporta expresiones path generalizadas y constantes booleanas, así como dos filtros básicos: select-from-where y construct-from-where. El primero devuelve una tabla variable, el segundo devuelve un subgrafo. Por lo tanto, las consultas construct-from-where proporcionan el cierre y la ortogonalidad, y además permite la composición de consultas. SeRQL no es seguro, ya que proporciona varias funciones recursivas.

##### TRIPLE

---

El término Triple denota tanto el lenguaje de consulta y sus reglas como la aplicación en si. El lenguaje deriva de F-Logic. Las tripletas RDF (S, P, O) están representadas como expresiones F-Logic S [P->O], las cuales pueden ser anidadas.

Por ejemplo, la expresión S[P1->O1, P2->O2[P3->O3]] se corresponde con las tripletas: (S,P1,O1), (S,P2,O2), y (O2,P3,O3).

Triple no distingue entre reglas y consultas, que son simplemente reglas sin encabezado, donde los resultados son los valores que se corresponden de las variables libres en la consulta. Por ejemplo:



FORALL X <- ( X[rdfs:label->"arbol"] )@default:In. Devuelve todos los recursos que contienen la etiqueta "arbol".

Debido a que el output es una tabla de variables y sus posibles vínculos, TRIPLE no cumple la propiedad del cierre. Además, tampoco es seguro, ya que permite reglas no seguras, como:

```
FORALL X ( X[rdfs:label->"arbol"] <- ( a[rdfs:label->"arbol"] )@default:In..
```

Mientras que TRIPLE es adecuado y cerrado para su propio modelo de datos, el mapeo de RDF a TRIPLE es no tiene pérdidas. Por ejemplo, los nodos anónimos RDF se hacen explícitos. TRIPLE es capaz de manejar varios modelos RDF simultáneamente, los cuales se identifican mediante el sufijo @model.

TRIPLE no codifica semánticas RDF fijas. Las semánticas tienen que ser especificadas mediante un conjunto de reglas a lo largo de la consulta. TRIPLE no soporta conjuntos de datos.

### RDQL

---

La sintaxis de RDQL sigue un patrón como SQL, donde una cláusula FROM se omite.

Por ejemplo `select ?p where (?p, <rdfs:label>, "arbol" )` devuelve todos los recursos con la etiqueta árbol en la variable libre `p`. La cláusula SELECT al principio de la query permite proyectar las variables. Las abreviaciones de los namespace pueden ser definidas en una consulta mediante una cláusula using independiente.

La información RDF Schema no es interpretada. Ya que la salida es una tabla de variables, y sus posibles correspondencias, RDQL no cumple por completo la propiedad del cierre y la ortogonalidad. RDQL es seguro y ofrece un soporte preliminar para tipos de datos.

### N3

---

Notation3 (N3) proporciona una sintaxis basada en texto para RDF. Sin embargo, el modelo de datos N3 se ajusta al modelo de datos RDF. Además, N3 permite definir reglas, las que se denotan utilizando una sintaxis especial, por ejemplo:

```
?y rdfs:label "arbol"  
?y a :QueryResult.
```

Estas reglas, aunque no son un lenguaje de consulta per se, pueden ser utilizadas con el propósito de consulta. Para este propósito las consultas tienen que ser almacenadas como reglas en un archivo dedicado, el cual se utiliza conjuntamente con los datos. Los comandos CWM Filter permiten automáticamente seleccionar los datos que son generados por reglas. Incluso aunque N3 cumple las propiedades de cierre, seguridad y ortogonalidad, la utilización de N3 como lenguaje de consulta es incómoda.

### Versa

---

Versa tiene una característica interesante, y es que el bloque principal del lenguaje es una lista de recursos RDF. Las tripletas RDF tienen un rol en las llamadas operaciones transversas, que tienen la forma `ListExpr - ListExpr -> BoolExpr`. Estas expresiones devuelven una lista de todos los objetos que cumplen las tripletas. Por ejemplo, la expresión trasversal `: all() - rdfs:label -> *`

Devolvería una lista conteniendo todas las etiquetas.

En una expresión transversal, podemos seleccionar alternativamente los sujetos poniendo una barra vertical al principio del símbolo flecha. Así: `all() |- rdfs:label -> eq("arbol")` devolvería como resultado todos los recursos que contengan la etiqueta "arbol". El hecho de que una expresión transversa se utiliza como una lista de expresiones, permite anidar expresiones para crear consultas más complejas. El árbol de estructura de datos y de expresiones hace que sea complicado proyectar varios valores de una vez. Versa utiliza el operador de la distribución para superar esta limitación. Crea una lista de listas, lo que permite seleccionar varias propiedades de una lista de recursos dada.

Versa ofrece soporte para reglas, ya que permite transversar predicados transitivamente. No tiene implementadas vistas, múltiples modelos y manipulación de datos. Sin embargo, versa cumple los criterios de ortogonalidad y seguridad.

## SPARQL

---

El protocolo SPARQL y el lenguaje de consulta SPARQL están actualmente bajo discusión en el W3C Working Draft. El pasado 28 de noviembre de 2005 han publicado un nuevo borrador. En estos momentos SPARQL es el lenguaje de consulta más potente para RDF/OWL. Además la tendencia es clara y buena parte de los sistemas de almacenamiento de RDF soportan ya SPARQL.

Está basado en anteriores lenguajes de consulta para RDF, como `dfDB`, `RDQL`, y `SeRQL`, y aporta nuevas características muy importantes. Existen varias implementaciones de SPARQL en diferentes plataformas y lenguajes, como `ARQ`, que implementa el lenguaje para Jena, y `SPARQL` para Sesame.

Proporciona las siguientes facilidades:

- Extrae información en forma de URIs, nodos en blanco y literales planos y tipados.
- Extrae subgrafos RDF
- Construye nuevos grafos RDF basados en la información de los grafos consultados.

Está basado en la correspondencia entre patrones de grafos. El patrón más simple de un grafo es el patrón triple, que es como una tripleta RDF pero con la posibilidad de una variable en cualquier posición en el sujeto, predicado u objeto. Con esta combinación se proporciona un patrón de gráfico básico.

Por ejemplo, un par de características especialmente interesantes son:

- `CONSTRUCT`: que permite construir un grafo de respuesta en vez de devolver la típica tabla de variables y valores de SQL.
- `DESCRIBE`: que permite recuperar todos los metadatos asociados a un recurso (URI).





### 9.4.5 Análisis de soluciones SGBD que implementan soporten RDF

#### ORACLE 10G

Oracle en su release 10.2 introduce un nuevo tipo de objeto (SDO\_RDF\_TRIPLE\_S) para almacenar datos RDF. Cada tripleta RDF <sujeito, propiedad, objeto> se trata como un único objeto de la base de datos. Como consecuencia, un único documento RDF con varias tripletas dará como resultado múltiples objetos de la base de datos. Las tripletas RDF son mapeadas en un grafo mediante el almacenamiento de los sujetos y objetos como nodos, y las propiedades como links. El almacenamiento de los datos RDF lo gestiona Oracle: todos los datos RDF se gestionan desde un esquema central, y se proporcionan funciones de acceso a nivel de usuario y constructores para hacer consultas y actualizar los datos RDF.

Las tripletas RDF se almacenan en una base de datos Oracle como una red lógica (utilizando Oracle Spatial Network Data Model - NDM), soportando grafos dirigidos y no dirigidos. NDM es la solución propuesta por Oracle para almacenar, manipular y analizar redes o grafos en la base de datos.

Las tripletas son parseadas y almacenadas en el sistema como entradas en las tablas rdf\_node\$ y rdf\_link\$. Los nodos en la red RDF se almacenan una sola vez y se reutilizan cuando se encuentran en nuevas tripletas. En las tablas de las aplicaciones definidas por el usuario solo se almacenan referencias en el objeto SDO\_RDF\_TRIPLE\_S para señalar a la tripleta almacenada en el esquema central.

Otra peculiaridad es que Oracle implementa la reificación (descripción de una tripleta RDF utilizando vocabulario RDF) utilizando Oracle XML DB DBUri, que referencia directamente las tripletas en la base de datos: sólo se almacena una nueva tripleta por cada reificación.

Las ventajas que se derivan de gestionar los datos RDF como objetos de la base de datos son las siguientes:

- Simplifica la modelización de aplicaciones RDF.
- Los datos RDF se pueden integrar fácilmente con otros datos de la empresa.
- La reusabilidad de los objetos RDF hace posible que se desarrollen nuevas aplicaciones con más facilidad.
- La abstracción de los objetos y la encapsulación de los comportamientos RDF facilita la comprensión y el mantenimiento de las aplicaciones.
- No se requieren mapeos entre los objetos RDF de la parte del cliente y las tablas y columnas de las bases de datos que contienen las tripletas.
- No se requieren configuraciones adicionales para almacenar datos RDF.

Oracle soporta los siguientes conceptos RDF: modelo de datos en grafos, vocabularios basados en URI, tipos de datos RDF y reificación. También soporta un subconjunto de consultas típicas en SQL.

#### Oracle Spatial RDF Data Model.

En Oracle Database 10g Release 2 se ha desarrollado un Nuevo modelo para almacenar los datos RDF y OWF. Esta funcionalidad reside en el Oracle Spatial Network Data Model (NDM). El modelo de datos RDF soporta tres tipos de objetos de base de datos: modelo, conjunto de reglas e índice de reglas.

#### Modelo

Hay solo un universo para todos los datos RDF almacenados en la base de datos.

Todas las tripletas RDF son parseadas y almacenadas en el sistema como entradas en tablas bajo el esquema MDSYS. Una tripleta RDF (sujeito, predicado, objeto) se trata como un objeto de la base de datos.

RDF\_MODEL\$ es una tabla de sistema creada para almacenar información en todos los modelos RDF y OWL de la base de datos. Cuando se crea un nuevo modelo RDF, se genera automáticamente un MODEL\_ID y se añade una entrada en la tabla del modelo.

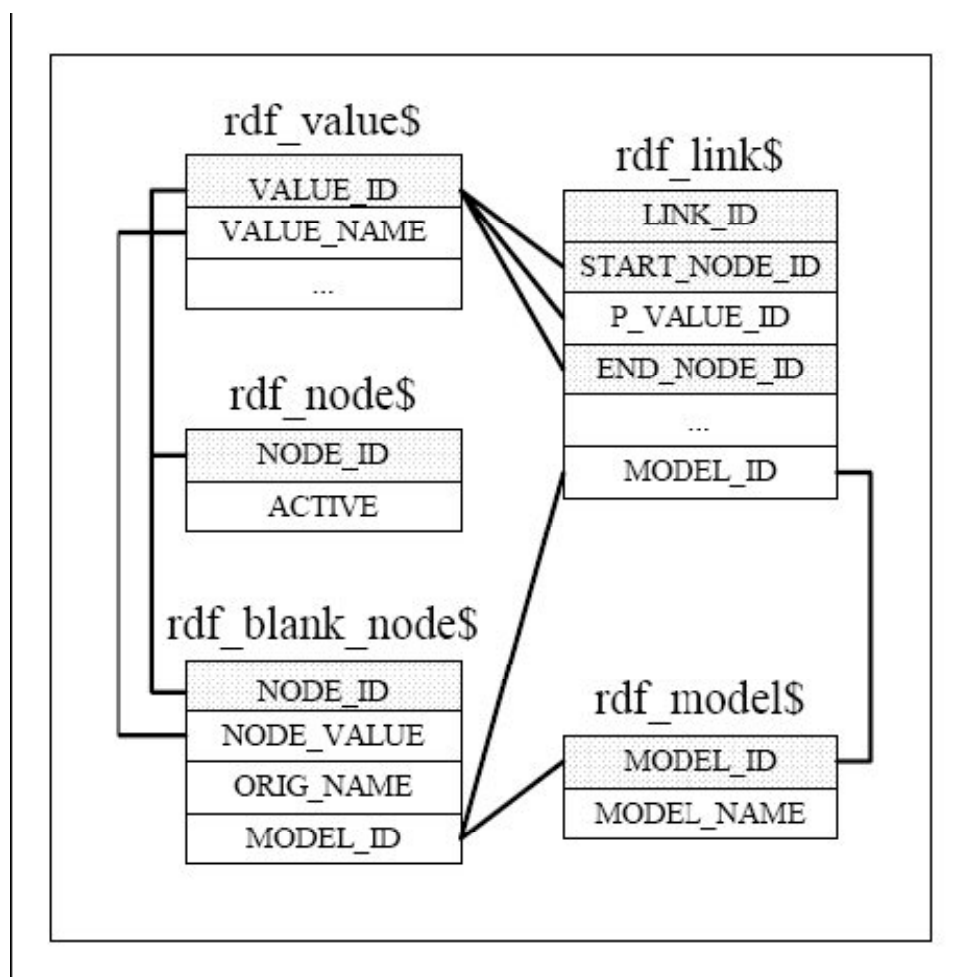
La tabla RDF\_NODE\$ almacena el VALUE\_ID para valores de texto que participan en sujetos u objetos de las sentencias. El NODE\_ID es el mismo que VALUE\_ID. Los valores NODE\_ID son almacenados una

sola vez, independientemente del número de sujetos u objetos en los que participe. La tabla de nodos permite a los datos RDF ser expuestos a todas las funciones analíticas y APIS disponibles en NDM.

La tabla LINKS\$ almacena las tripletas para todos los modelos de la base de datos.

Además, el MODEL\_ID lógicamente particiona la tabla RDF\_LINK\$. Seleccionando todos los links de un MODEL\_ID específico, devuelve la red RDF de ese modelo en particular.

La tabla RDF\_VALUE\$ almacena los valores de texto, es decir, el URI o los literales para cada parte de la tripleta. Cada valor de texto se almacena una sola vez, y un único VALUE\_ID se genera para las entradas de texto. URIs, nodos en blanco, literales planos y literales tipados son diferentes posibles entradas de VALUE\_TYPE.



**Fig. 21:** RDF\_VALUE\$, RDF\_NODE\$, RDF\_LINK\$ y RDF\_MODEL\$ son las tablas necesarias para almacenar RDF en el Oracle RDF Data Model

Los nodos en blanco se utilizan para representar objetos desconocidos, y cuando la relación entre un nodo sujeto y otro nodo es n-aria. Los nuevos nodos en blanco se generan automáticamente si se encuentran en las tripletas. Sin embargo, es posible que los usuarios reutilicen nodos en blanco, por ejemplo, cuando se insertan datos en contenedores o colecciones. La tabla RDF\_BLANK\_NODE\$ almacena los nombres originales de los nodos en blanco que se reutilizan cuando se encuentran en las tripletas.

Para representar una sentencia reificada, se crea un recurso utilizando el LINK\_ID de la tripleta. El recurso puede ser utilizado como el sujeto o el objeto de una sentencia. Para procesar una sentencia reificada, en primer lugar, en el sujeto de una nueva tripleta se inserta el valor del recurso de la sentencia, como propiedad rdf:type y como objeto: rdf:Statement. Entonces, una nueva tripleta se introduce por cada aseveración sobre la sentencia reificada. Sin embargo, cada sentencia reificada tendrá



solo un rdf:type asociado a un rdf:Statement, independientemente del numero de aserciones hecho utilizando este recurso.

El modelo de datos RDF de Oracle soporta contenedores y colecciones. Un contenedor o colección tendrá un rdf:container\_name o rdf:collection\_name asociado a un rdf:type, y un LINK\_TYPE de un RDF\_MEMBER.

Dos nuevos tipos de objetos se han definido para los datos RDF. SDO\_RDF\_TRIPLE sirve como la representación de la tripleta de un dato RDF, SDO\_RDF\_TRIPLE\_S se define para almacenar datos persistentemente en la base de datos. La función GET\_RDF\_TRIPLE() se puede utilizar para devolver un tipo SDO\_RDF\_TRIPLE.

#### **Conjunto de reglas (Rulebase)**

Cada rulebase RDF consiste en un conjunto de reglas. Cada regla es identificada por un nombre, i consiste en un patrón 'IF' para los antecedentes, una condición de filtro opcional que además restringe los subgrafos, y un patrón THEN para los consecuentes.

Cuando se aplica una regla a un modelo RDF puede producir tripletas adicionales.

Un modelo RDF aumentado con un conjunto de reglas es equivalente al conjunto de tripletas original además de las tripletas inferidas mediante la aplicación del rulebase al modelo. Las reglas de un rulebase pueden aplicarse al rulebase en si mismo para generar tripletas adicionales.

Oracle soporta tanto un rulebase RDF que implementa las reglas que conlleva RDF, y un rulebase RDF Schema (RDFS), que implementa las reglas de RDFS. Ambos rulebases se crean cuando se añade soporte RDF a la base de datos. También es posible crear un rulebase adicionales para habilidades de inferencia especializadas.

Para cada rulebase, se crea una tabla de sistema, además de una vista de sistema de la rulebase. Esta vista se utiliza para insertar, borrar y modificar reglas en la rulebase.

#### **Índice de Reglas.**

Un índice de reglas es un objeto que contiene tripletas pre-procesadas que pueden ser inferidas aplicando un conjunto de rulebases a un conjunto específico de modelos. Si una consulta se refiere a cualquier rulebase, debe existir un índice de reglas para cada combinación de rulebase-modelo en la consulta.

Cuando se crea un índice de reglas, una vista se crea también en las tripletas RDF asociadas con el índice bajo el esquema MDSYS. Esta vista es visible únicamente por el propietario de del índice de reglas, y para los usuarios que tengan este privilegio. La información sobre todos los índices de reglas e mantiene en la vista de índice de reglas. La información sobre todos los objetos de la base de datos, como modelos y rulebases, relacionada con los índices de reglas se mantienen en la vista Rule Index Datasets.

#### **Consultas**

La utilización de la función de tabla SDO\_RDF\_MATCH permite embeber una consulta de grafos en una consulta SQL. Esta función tiene la habilidad de buscar patrones contra los datos RDF, incluyendo inferencia, datos RDF, RDFS y reglas definidas por el usuario. La función SDO\_RDF\_MATCH se ha diseñado para cumplir la mayoría de los requerimientos identificados por W3C en SPARQL para consultas de grafos.

También se proporciona una API Java para representaciones de grafos y análisis de grafos, que incluyen la posibilidad de encontrar un path entre dos recursos, o de encontrar un path entre dos recursos cuando los links son de un tipo determinado.

**Tabla 46: Análisis de Oracle 10g**

## Jena2

Jena es un framework Java para construir aplicaciones para la Web Semántica.

Proporciona un entorno de programación para RDF, RDFS y OWL, incluyendo un motor de inferencia basado en reglas. El código de Jena es libre, y ha ido evolucionando a partir de trabajo con HP Labs Semantic Web Programme.

El framework de Jena incluye:

- Una API RDF
- Lectura y escritura de RDF en formato RDF/XML, N3 y N-Triples.
- Una API OWL.
- Almacenamiento en memoria y persistente.
- Consultas a través de RQL.

Jena es el toolkit más importante de los programadores de la Web Semántica.

Actualmente, va por la segunda generación del producto. Jena2 está diseñado para solucionar algunos de los problemas de Jena1. Específicamente, el diseño del esquema se ha modificado para mejorar problemas de rendimiento debido a demasiadas uniones de tablas y el incremento del almacenamiento resultante de la implementación de la reificación.

Jena1 almacena las tripletas de una forma normalizada. Una tabla de sentencias almacena referencias al sujeto, predicado y objeto, y el valor real de las URIS y de los literales se almacena en dos tablas adicionales. Este diseño es eficiente en espacio, porque los valores de texto se almacenan una única vez, independientemente del número de veces que aparezcan en las tripletas. Sin embargo, una unión a tres bandas se requiere para operaciones de búsqueda.

Además, la tabla de sentencias no es escalable para grandes bases de datos.

Jena 2 utiliza una visión desnormalizada, multi-modelo y de triple almacenamiento.

Los modelos se almacenan en tablas separadas, y cada modelo almacena las sentencias afirmadas en una tabla y las sentencias reificadas en otra, aunque este comportamiento por defecto puede ser alterado mediante la configuración. La tabla de sentencias confirmadas almacena los valores textuales reales de las tripletas en columnas de sujeto, predicado y objeto. Los valores textuales se almacenan además redundantemente, si el mismo valor textual aparece en diferentes tripletas.

Jena2 además consume mas espacio de almacenamiento que Jena 1; sin embargo, el número de joins de las tablas requerido se reduce en tiempo de consulta, mejorando el rendimiento.

Las sentencias reificadas se almacenan en una tabla de propiedades, que tiene columnas StmtURI (que representan la sentencia reificada), rdf:subject, rdf:predicate, rdf:object and rdf:type. Una fila simple con todos los atributos presentes representa una tripleta reificada.

Además de las tablas para sentencias afirmativas y reificadas, Jena2 puede configurarse para incluir tablas de propiedades en la creación de grafos. Estas tablas almacenan pares de sujeto-valor para predicados específicos. La tabla columnar incluye el sujeto y cada predicado que va a ser representado en la tabla.

Por ejemplo, una tabla de propiedades Dublin Core puede incluir las columnas sujeto, dc:title, dc:publisher, y dc:description, donde una fila simple almacena los valores del predicado para un sujeto común. Las tablas de propiedades están diseñadas para proporcionar una ligera reducción del almacenamiento, ya que los predicados URIs no se almacenan. Se intenta agrupar propiedades que son accedidas conjuntamente y de esta forma incrementar el rendimiento.

Estas son las tablas básicas que componen el esquema de Jena2, aunque hay otras tablas de soporte.

Jena2, así como otros sistemas que implementan almacenamiento persistente en base de datos relacionales, utiliza tablas relacionales planas.

El modulo de bases de datos de Jena proporciona una implementación de la interface del modelo Jena, pero con la capacidad de almacenar y recuperar datos RDF utilizando una base de datos.

Actualmente, soporta MySQL, Oracle y PostgreSQL para almacenamiento persistente, y se ejecuta bajo Windows y Windows XP.

El subsistema de persistencia soporta la capacidad Fastpath para las consultas RDQL que genera



consultas SQL dinámicamente para mejorar el rendimiento de las consultas RDQL tanto como sea posible en el motor de la base de datos.

El algoritmo Fastpath trabaja dividiendo las variables y constantes de una consulta en uno o más grupos, llamados stages, donde cada stage consiste en variables que pueden ser procesadas juntas. Stages que consisten en más de una variable son procesadas mediante una consulta simple generada dinámicamente. Las stages que consisten justamente en una variable son procesadas mediante una operación Find.

Esto se considera más eficiente que generar una consulta SQL. El algoritmo Fastpath también determina el orden de la ejecución para intentar minimizar el tiempo de ejecución.

También hay que mencionar ARQ, que es un motor de consulta para Jena que soporta el lenguaje SPARQL.

Algunas de las características de ARQ son:

- Soporta varios lenguajes de consulta: SPARQL, RDQL, ARQ (el lenguaje propio del motor de base de datos).
- Múltiples motores de consulta: Motor de propósito general, motores de acceso remoto y traductor a SQL.

**Tabla 47: Análisis de Jena 2**



## SESAME

Sesame es una arquitectura modular basada en Java y de código libre para almacenar y hacer consultas sobre RDF y RDF Schema. Soporta tres lenguajes de consulta (RQL, SeRQL y RDQL) y puede usar la memoria principal de la máquina o una base de datos relacional para el almacenamiento de los datos RDF.

En la mayoría de los casos, Sesame depende de una base de datos relacional para almacenar los datos RDF. Actualmente, Sesame soporta PostgreSQL, MySQL, Oracle (9i o superior) y SQL Server. Tanto PostgreSQL y MySQL se distribuyen con licencia Open Source. En general, Sesame rinde más con MySQL que con PostgreSQL. No hay estudios disponibles sobre el rendimiento sobre Oracle o SQLServer.

Si se desea utilizar Sesame utilizando el almacenamiento en memoria, no se necesita la instalación de un sistema de gestión de base de datos.

### La Necesidad de Lenguaje de consulta para RDFS

Los documentos RDF y los esquemas RDF pueden considerarse sobre tres niveles diferentes de abstracción:

- En el nivel sintáctico, son documentos XML.
- En el nivel de estructura, que considera un conjunto de tripletas.
- En el nivel semántico, que constituye uno o más grafos con semánticas parcialmente predefinidas.

#### a. Consultas a nivel Sintáctico.

El modelo RDF puede ser descrito mediante la notación XML, por lo que parece razonable pensar que se pueden realizar consultas RDF empleando para ello lenguajes de consulta para XML, como por ejemplo Xquery. Pero este enfoque no considera el hecho que de XML maneja una estructura en árbol mientras que RDF es un grafo etiquetado. Este no es el único inconveniente, también existe el problema que la serialización de RDF en XML se puede hacer de diferentes formas.

#### b. Consultas a Nivel de Estructura.

Un documento RDF se representa como un conjunto de tripletas; cada tripleta representa una sentencia de la forma sujeto-predicado-objeto. Diferentes lenguajes de consulta han sido propuestos e implementados teniendo en cuenta que los documentos RDF son un conjunto de tripletas, y que estas tripletas pueden ser consultadas de diferentes formas.

Aplicando un lenguaje de consulta como Squish, se podría formular la siguiente consulta:

```
SELECT ?x
FROM somesource
WHERE (rdf:type ?x Algo)
```

La gran ventaja de esta consulta es que directamente referencia el modelo RDF y es independiente de la sintaxis empleada para la representación de los datos.

Sin embargo, una desventaja de cualquier lenguaje de consulta a éste nivel es que interpretan cualquier modelo RDF como un conjunto de tripletas, inclusive aquellos elementos que tienen una semántica especial en RDFS.

#### c. Consultas a nivel Semántico.

Un requerimiento fundamental es que se necesitan consultas a nivel semántico, de modo que permitan obtener todo el conocimiento representado a través de un modelo RDFS. Se pueden considerar dos opciones para alcanzar este objetivo:

- Procesar y almacenar la clausura del grafo dado como base para la consulta.
- Permitir un procesador de consultas que infiera nuevas sentencias de acuerdo a la necesidad.

### La arquitectura de SESAME

En la siguiente figura se muestra la arquitectura de Sesame. Uno de sus componentes principales es SAIL (Storage And Inference Layer) que es el nivel que se encarga de interactuar con los manejadores



de bases de datos (DBMS) permitiendo de este modo mantener la arquitectura de Sesame independiente del DBMS. SAIL es una API que ofrece métodos RDF específicos a sus “clientes” y traduce estos métodos a llamadas de un DBMS específico. Los módulos funcionales de Sesame son “clientes” de SAIL. Actualmente existen tres módulos: EL motor de consultas RQL, el Módulo de Administración RDF y el módulo de exportación RDF.

Dependiendo del ambiente en el cual se implante Sesame ofrece diversas formas de comunicación, sea vía HTTP para contexto Web, o Remote Method Invocation (RMI) o Simple Object Acces Protocol (SOAP).

### Módulos Funcionales de Sesame

#### a. Módulo de Consulta RQL

La versión de RQL de Sesame está caracterizada por cumplir con los estándares de la W3C, sin embargo no soporta la tipificación de datos. El proceso de consulta es el siguiente: después del parsing y la construcción del árbol de consulta, pasa a un optimizador de consultas que transforma la consulta en un modelo equivalente pero más eficiente. La optimización consta de un conjunto de heurísticas, éstas pre-evaluaciones de optimización no dependen de ningún método de almacenamiento en particular.

EL modelo optimizado es subsecuentemente evaluado siguiendo la estructura de árbol generada, cada objeto representa una unidad básica en la consulta original y se evalúa a si mismo extrayendo información por medio de SAIL cuando lo necesita. La gran ventaja de este enfoque es que los resultados se retornan de la misma forma, en vez de construir primero la consulta completa en memoria.

El grueso de las consultas son resueltas por el motor RQL, por ejemplo cuando una consulta contiene una operación de semi-join, cada sub-consulta es evaluada y la operación de semi-join es ejecuta entonces por el motor de consultas sobre los resultados.

Otra alternativa sería aprovechar la optimización a partir del DBMS, pero esto haría que el proceso de consulta fuera dependiente de un DBMS específico.

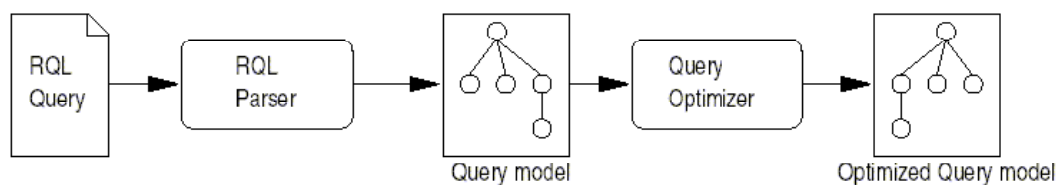


Fig. 22: Parsing de las consultas y optimización del modelo

#### b. Módulo de Administración.

Para posibilitar la inserción de datos RDF e información de esquemas en un repositorio, Sesame utiliza el módulo de administración, ofreciendo dos operaciones fundamentales:

- Adición incremental de datos y esquemas RDF.
- Limpiar un repositorio.

Hay que señalar que una opción borrado (por sentencias) funcional aun no esta disponible.

El módulo extrae la información de una fuente RDF(S), generalmente en XML serializado, y se hace un parsing usando un parser RDF (por ejemplo, el ARP RDF parser que es parte del Toolkit de Jena), que entrega información al módulo de administración de la forma básica (objeto, atributo, valor), y luego al repositorio por medio de la capa SAIL reportando los errores que pudieron haber ocurrido.

#### c. Módulo de Exportación RDF.

Permite exportar los contenidos de un repositorio en formato XML serializado, el objetivo de este modulo es combinar Sesame con otras herramientas RDF, dado que la mayoría reconoce el formato. Además de permitir la exportación de datos, de esquemas, o de ambos, dependiendo si es necesario la semántica o no.

Tabla 48: Análisis de Sesame

### **RDF GATEWAY**

RDF Gateway es una plataforma diseñada para la Web Semántica. Es un servidor de aplicaciones, un servidor Web, y un servidor de bases de datos RDF.

El motor de base de datos de RDF Gateway no es como los de los Sistemas de Gestión de Bases de datos Relacionales (RDBMS). Soporta un lenguaje de comandos similar a SQL, múltiples bases de datos, tablas almacenadas en memoria o disco, cursores y transacciones.

La principal diferencia entre RDF Gateway y una base de datos relacional es como se representa la semántica de un modelo de datos. Una base de datos relacional codifica la semántica implícitamente utilizando elementos de almacenamiento como tablas, columnas, y claves ajenas. Las aplicaciones cliente tienen que ser desarrolladas para comprender la semántica de la información relacional. Esto hace que sea difícil compartir información entre sistemas que no comparten un esquema relacional idéntico. RDF proporciona un estándar abierto para especificar la semántica de un modelo de datos. Además, RDF Gateway no necesita codificar la semántica utilizando tablas, columnas y claves ajenas. Con RDF Gateway todas las tablas tienen el mismo formato y las claves ajenas no son necesarias.

Ya que cualquier modelo puede ser descrito utilizando tripletes RDF, todas las tablas tienen el mismo formato. Cada tabla tiene cuatro columnas: Contexto, predicado, sujeto y objeto.

Con RDF Gateway las tablas no se asocian con una clase simple de objeto en el modelo de datos, ya que no pueden alojar objetos de muchas clases diferentes.

Una tabla de RDF Gateway puede incluso mantener tablas de varios esquemas diferentes. Las tablas de RDF no son más que fuentes de datos.

Alguna de sus características referentes a la gestión de los datos RDF son las siguientes:

#### **Motor de Base de datos Nativo.**

- Utiliza comandos similares a SQL para realizar consultas de datos RDF.
- Razonamiento deductivo vía reglas de inferencia (librerías para OWL y RDFS)
- Alto rendimiento en disco y tablas de memoria.
- Transacciones con niveles aislados.
- Tablas virtuales que integran fuentes de datos externas (páginas Web, documentos locales, bases de datos relacionales, etc.)
- Consultas federadas a través de múltiples fuentes de datos internas y externas.
- Búsquedas de texto.
- Encriptación de datos
- Soporte para la serialización RDF/XML, N3 y NTriples
- Seguridad a nivel de tabla y de sentencias.
- Acceso a datos desde otras aplicaciones vía http, Microsoft ADO o Sun Microsystems JDBC.

#### **Arquitectura:**

RDF Gateway es un servidor de alto rendimiento que combina el conjunto de características de un sistema de gestión de base de datos y un servidor Web. El objetivo de este diseño es proporcionar una aplicación completa para agrupar, consultar, transformar y entregar información. RDF Gateway proporciona un lenguaje muy rico de script denominado RDFQL. Está basado en Java Script, con extensiones de consulta para realizar búsquedas federadas entre diferentes fuentes de recursos. Las páginas RDFQL pueden escribirse y utilizarse para crear aplicaciones muy completas llamadas packages.





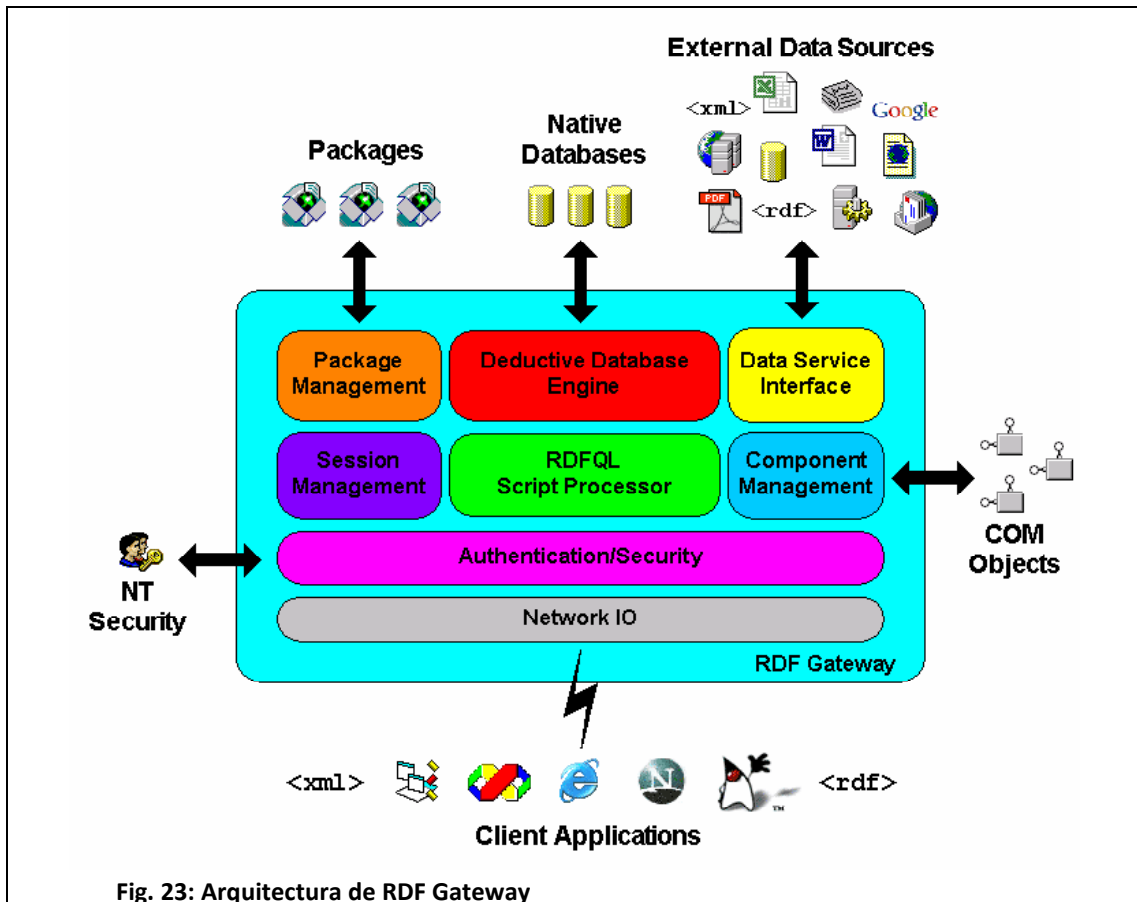


Fig. 23: Arquitectura de RDF Gateway

**RDFQL Script Processor**

El procesador de RDFQL script es una máquina virtual que compila, cachea y ejecuta RDFQL scripts. RDFQL integra extensiones de consulta similares a SQL para proporcionar acceso a la base de datos. También RDFQL permite que las páginas contengan una combinación de scripts y contenido estático similar a Microsoft Active Server Pages (ASP).

**Base de datos deductiva**

RDF Gateway tiene un motor de base de datos que ha sido diseñada desde el principio para soportar RDF. Su diseño semiestructurado es ideal para consultar y almacenar tripletes RDF. El motor deductivo de la base de datos realiza una evaluación de las consultas que es federada a lo largo de todas las fuentes de datos especificadas. Las capacidades de inferencia lógica del motor proporcionan soporte para la sintaxis declarativa de RDFQL. Implementa si propia persistencia en archivos, eliminando la necesidad de un sistema de almacenamiento de datos externo.

**Interfaz de servicio de datos.**

La interfaz de servicio de datos permite a fuentes de datos externas integrarse con RDF Gateway. Un proveedor de servicio de datos es un módulo que implementa su interface y expone el contenido de un tipo concreto de datos como datos RDF.

RDFQL permite que se realicen consultas federadas sobre múltiples servicios de datos Esta interfaz abierta hace posible utilizar cualquier proveedor de servicio de datos disponible actualmente o desarrollar un proveedor personalizado para una fuente de datos.

**Autenticación / Seguridad**

RDF Gateway tiene un modelo de seguridad basado en derechos y permisos para controlar el acceso al servidor y a los recursos de la base de datos. Soporta sus propios usuarios y roles, además de grupos y usuarios de NT. Un usuario de NT se autentica siempre utilizando las credenciales NT para la cuenta. Esta utilización de grupos y usuarios de NT hace posible que se gestione una administración de seguridad externa.

**Network IO**

RDF Gateway tiene un interfaz de red optimizado para el rendimiento y la escalabilidad. Soporta tanto http como un protocolo propietario basado en TCP/IP.

La capa de red IO soporta esquemas de autenticación de red seguros como NT Challenge/Response (NTLM). RDF Gateway tiene drivers clientes para Microsoft ADO y Sun Microsystems JDBC. Esto permite a RDF Gateway soportar una gran cantidad de clientes, como navegadores Web, aplicaciones Windows, aplicaciones Java y clientes basados en XML o RDF.

**Package Management**

RDF Gateway permite que las aplicaciones se desarrollen y se distribuyan como paquetes. Un paquete consiste en páginas RDF Server, páginas HTML, imágenes o cualquier otro tipo de archivo.

**Component Management**

RDFQL soporta COM en sus scripts para el servidor. Esto permite que la funcionalidad de RDF se extienda para que las aplicaciones se integren con RDF Gateway.

**Tabla 49: Análisis de DRF Gateway**



## 9.5 Portales Web

---

Un portal Web es un lugar de Internet que proporciona información relacionada con un dominio concreto (por ejemplo [marmiton.org](http://marmiton.org), un portal de recetas de cocina).

Un portal Web ofrece a sus usuarios una serie de servicios cómo:

- recibir noticias
- comunicación entre usuarios (blogs)
- construcción de comunidades
- encontrar enlaces hacia de otras recursos web de interés.

El éxito de un portal web recaerá básicamente en la capacidad de indexación de los contenidos y en la sindicación. Una gran parte de la información de un portal es enviada por los miembros de la comunidad, que queda grabada en una de las subcategorías del portal (por ejemplo, recetas de cocina vegetarianas).

Pero un índice de contenidos no es una herramienta lo suficiente potente para la búsqueda de contenidos: se hace necesaria la existencia de una ontología relacionada con la temática del portal, que defina los conceptos relacionados con el dominio y los posibles vínculos entre conceptos.

Ejemplos de portales basados en ontologías son OntoWeb y The Open Directory Project.



## 9.6 Colecciones multimedia

---

Gracias a las ontologías se pueden asignar anotaciones semánticas a las colecciones de imágenes, audio o vídeo. Ya se han comentado las dificultades que tienen las máquinas actualmente para comprender un texto en lenguaje natural: es obvio que extraer el significado semántico de un fichero multimedia pone todavía más problemas.

Por esta razón, estos tipos de recursos se indexan con cabeceras y metadatos. El problema es que diferentes usuarios pueden describir estos objetos de maneras diferentes. Idealmente, las ontologías capturan informaciones adicionales sobre el dominio de busca y mejoran la recopilación de imágenes.

Hay dos tipos de ontologías multimedia:

- relativas al medio: contienen descripciones de propiedades y taxonomías de diferentes tipos de media (por ejemplo, la duración de un vídeo)
- relativas al contenido: describen el dominio al cual corresponde un recurso.

Existen diferentes protocolos para entornos multimedia: SMIL (Synchronised Multimedia Integration), SVG (Scalable Vector Graphics), VoiceXML.

Un ejemplo sería una colección de imágenes publicadas en el sitio Web de un museo: La Universidad de Helsinki lleva a cabo un proyecto de portal semántico de los museos finlandeses.



## **9.7 Gestión de sitios Web corporativos**

---

Las grandes compañías (por ejemplo Atos Origin) acostumbran a tener una pila de páginas web relacionadas con temas diferentes:

- catálogos de productos
- inserciones en prensa
- ofertas de trabajo
- e-bussiness
- e-learning
- búsqueda y tecnologías, etc.

Las ontologías pueden ayudar a indexar todos estos documentos y proporcionar una extracción de información más precisa. Hay empresas que disponen de su propia taxonomía por organizar su información, pero una sola ontología acostumbra a ser insuficiente puesto que sólo muestra un punto de vista del dominio (el de la emprendida en cuestión, sin tener en cuenta otras entidades del mismo campo).

Se hace necesaria la posibilidad de trabajar con múltiples ontologías, puesto que no tendrán el mismo punto de vista de un mismo concepto un ingeniero de sistemas de la compañía A y un comercial de zona de la emprendida B.



## 9.8 Agentes Inteligentes

---

La Web Semántica proporciona a los agentes la posibilidad de comprender e integrar varias fuentes de información. Un ejemplo de agente semántica utilizando servicios web sería “un planificador de actividades”, capaz de inferir las preferencias de un usuario (como tipo de películas, platos preferidos, tipos de lectura) y organizar un fin de semana en función de toda esta información.

La efectividad de esta tarea dependerá de la cantidad y calidad de la ofrecida existente en el entorno y en la capacidad de definición del perfil de usuario (que almacena sus preferencias). Durante el proceso de busca de estas actividades se accederá a servicios específicos, que pueden incluir sistemas de puntuación (por ejemplo un portal dedicado al cine como <http://www.allocine.fr>, con una valoración de las películas más votadas por el público y por la prensa especializada).

Agentcities es un ejemplo de iniciativa que estudia el uso de agentes en un entorno de servicios web distribuidos. Pretende crear una red de agentes que representen una ciudad y los servicios que puede ofrecer esta ciudad. En una primera etapa se definen servicios orientados al ocio y el consumo, pero más adelante se puede aplicar a servicios de comercio electrónico (business to business y business to enterprise).

Así, los agentes de software, denominados en el ámbito anglosajón softbots o killer apps, constituyen una de las áreas clave en el desarrollo de la inteligencia artificial, aunque intervienen otras áreas del conocimiento como la sociología, la informática, la lógica, la lingüística y las telecomunicaciones, entre otras, ello ha llevado a proponer la creación de una nueva disciplina dentro de la Ingeniería de Software, la Ingeniería de Software Orientada a Agentes ( Agent-Based Software Engineering ), cuyo fundamento es solucionar el problema que supone la heterogeneidad de las situaciones en las que puede encontrarse un agente de software y solucionar este problema mediante la comunicación y la cooperación.

En la inteligencia artificial ha surgido un nuevo paradigma conocido como “paradigma de agente”, que ha tomado un gran auge entre los investigadores en el diseño y desarrollo de entidades que puedan actuar de forma autónoma y razonada. Los investigadores en el campo de los agentes computacionales han aportado varias definiciones al término, cada uno desde su perspectiva particular, fundamentada básicamente en la línea de investigación en la que trabajan (inteligencia artificial, ingeniería de software, sistemas autónomos). No obstante, la aplicación de los agentes de software es del interés para especialistas en organización de información por las perspectivas que abren en lo que se refiere a la localización, identificación, relación, mantenimiento y selección de recursos de información.

Un agente de software puede definirse como “un programa autocontenido capaz de controlar su proceso de toma de decisiones y de actuar, basado en la percepción del ambiente, en la persecución de uno o varios objetivos”. En esencia, de la definición anterior es posible extraer algunas características que deben tener los agentes:

- Deben ser parte de un ambiente.
- Deben censar su entorno y actuar sobre él.
- Deben responder según los objetivos para los que se diseñaron.

Para ello, según Jennings y Wooldridge, son necesarios una serie de atributos o propiedades que lo definen como agente:

- **Autonomía:** Capacidad de actuar sin intervención directa de una persona u otro agente, y controlar sus propias acciones y estado interno.



- **Habilidad social:** Un agente debe ser comunicativo, incorporar habilidades para interactuar con otros agentes o incluso con alguna persona, para solicitar información o bien para exponer los resultados obtenidos de la ejecución de las tareas encargadas.  
Reactividad: Se refiere al hecho de que un agente debe poder censar el estado del ambiente dentro del cual se encuentra inmerso y, en función de esto, actuar para responder de manera adecuada a los cambios que produzcan.
- **Orientación por objetivos:** Un agente no sólo debe actuar ante los cambios detectados en el medio, sino que, además, debe trabajar en función de los objetivos para los cuales se diseñaron y las tareas que les fueron delegadas en cada momento.
- **Continuidad temporal:** Un agente es un proceso temporalmente continuo. A diferencia de un programa convencional del cual se conoce su inicio y fin, un agente debe ejecutarse hasta que se alcance el conjunto de objetivos solicitados, o bien, mientras su ciclo perdure y su usuario no desee detenerlo.

El ciclo de vida de un agente depende de sus características, de las tareas que realice y de los deseos de su usuario en cuanto al tiempo durante el cual debe ejecutarse. De ahí, la movilidad como capacidad del agente para viajar por toda la red, nodo a nodo, en busca de recursos que cumplan con su estrategia de búsqueda. La movilidad no es una propiedad indispensable para un agente, sino que modifica la forma en que cumple con sus objetivos; en este caso, puede recurrir a los recursos que ofrece una red de computadoras, y aportar una nueva forma de computación distribuida. Según esto, Berney propone una clasificación de agentes de acuerdo con sus líneas de investigación y desarrollo:

- **Agentes de interfaz:** Un agente de interfaz es un software cuasi-inteligente que asiste a un usuario cuando interactúa con una o más aplicaciones. La motivación es que se les pueda delegar tareas complejas y laboriosas. Son asistentes personales que reducen el trabajo ante la sobrecarga de información, por ejemplo, el filtrado de los mensajes de correo electrónico o la recuperación de archivos en Internet.
- **Agentes de colaboración:** Constituyen sistemas multiagentes, es decir, existe más de un agente dedicado a satisfacer los requerimientos de sus usuarios. Para ello, es necesario contar con esquemas de comunicación entre agentes que posibiliten la cooperación y el intercambio de conocimiento.
- **Agentes móviles:** Los agentes móviles son aplicaciones capaces de viajar por una red de computadoras, interactuando con servidores externos y recolectar información en nombre de su dueño y retornar luego de completar las tareas establecidas previamente.  
**Agentes de recuperación de información:** El objetivo principal de los agentes dedicados específicamente a la recuperación de información es obtener información para el usuario. La motivación para su construcción es que con el crecimiento vertiginoso de Internet, la cantidad de información accesible supera el tiempo de disponible para analizarla.

*"El papel del agente inteligente en el proceso de recuperación semántica de información no debe confundirse con la de un buscador inteligente".* Un buscador inteligente se aprovechará del enriquecimiento semántico de los recursos web para mejorar, principalmente, en la precisión y la recuperación de información. , aunque su funcionamiento se basará, como los actuales buscadores, en la previa indización de todos aquellos recursos susceptibles de recuperarse. Por otra parte, el agente inteligente recorrerá el web por medio de enlaces entre recursos (taxonomías, tesauros, ontologías) en busca de aquella información que le sea solicitada y puede además interactuar con el entorno para el cumplimiento de tareas encomendadas mediante la utilización de esquemas de metadatos codificados en esquemas RDF ( RDF Schema ) que pudiesen procesar y reutilizar ontologías.

La web semántica proyecta como uno de sus fundamentos el desarrollo de sistemas de información basados en el conocimiento de agentes inteligentes para lograr la interoperabilidad, no sólo sintáctica y semántica, sino entre aplicaciones independientemente de la acción de los usuarios de la red. Sustentados sobre la base de la heterogeneidad, constituyen nuevos métodos para la organización y la recuperación de información en entornos distribuidos.

## 9.9 Agentes y Servicios Web

---

El costo de desarrollar software siempre es muy alto y la diversidad de las plataformas es una realidad desde el inicio de la Informática. De hecho, conforme más complejas fueron las aplicaciones que las empresas demandaban, más caras era desarrollarlas. Para enfrentarse a esta situación, se han seguido diferentes líneas de implementación, todas ellas dirigidas a reutilizar las aplicaciones desarrolladas.

Una de estas propuestas fue y es la estandarización de lenguajes de programación de forma que si cualquier desarrollador escribía un programa en "C", sólo necesitaría un compilador de C en la plataforma específica en la que quisiera ejecutar la aplicación. Esto, en realidad, es difícil de hacer funcionar, porque surgieron pequeñas diferencias y extensiones de C que hacían difícil transportar una aplicación entre diferentes plataformas. C, un lenguaje de programación desarrollado en 1972 por el estadounidense Dennis Ritchie en los Laboratorios Bell, considerado por muchos un lenguaje ensamblador más independiente de la máquina que un lenguaje de alto nivel, su estrecha asociación con el sistema operativo UNIX, su enorme popularidad y su homologación por el American National Standards Institute (ANSI) lo han convertido tal vez en lo más cercano a un lenguaje de programación estandarizado en el sector de las microcomputadoras y estaciones de trabajo.

Otra posibilidad fue la que ha desarrollado Sun Microsystems con "Java". Se programa para una plataforma, pero para una plataforma virtual, en este caso, para la máquina virtual Java ( Java Virtual Machine ), y en cada plataforma real se implementa una máquina virtual de Java, que será la encargada de ejecutar las aplicaciones escritas en Java.

Java es lenguaje de programación orientado a objetos desarrollado por la empresa Sun Microsystems en 1995 y que se ha extendido ampliamente en el web. Es un lenguaje de alto nivel y propósito general similar a C++, con marcadas características de seguridad y transportabilidad. Este lenguaje define una máquina virtual independiente de la plataforma donde se ejecuta, que procesa programas, llamados Applets, descargados desde el servidor web. Además, debido al modo de ejecución de los Applets, este lenguaje es muy seguro frente a la presencia y ataque de virus informáticos.

Las implementaciones son específicas de cada plataforma, pero al ser todas las máquinas virtuales exactamente la misma, los programas escritos en Java deben poder ejecutarse sin ningún problema en todas las plataformas que tengan una máquina virtual de Java.

Esta apuesta ha demostrado ser muy útil y funcionar bastante bien. Sin embargo: ¿qué pasa si se tienen varias aplicaciones desarrolladas en lenguajes propietarios o en plataformas específicas y que interactúen entre ellas? El costo de elegir un único lenguaje o plataforma y migrarlo al mismo es descabellado en la mayoría de las situaciones, y es aquí donde los Web Services, así como otras tecnologías, pueden ser de gran utilidad.

Con los Web Services se pueden reutilizar desarrollos empleados sin importar la plataforma en la que funcionan o el lenguaje en el que están escritos. Los Web Services se constituyen en una capa adicional a estas aplicaciones de forma que pueden interactuar entre ellas, a partir del uso de tecnologías estándares para comunicarse, desarrolladas en el contexto de Internet por organismos importantes de estandarización como: ANSI, IEEE ( Institute of Electric and Electronics Engineers .), ISO y W3C.

Actualmente, gran parte de los presupuestos destinados al diseño y mantenimiento de sistemas de información se dedican a resolver problemas de integración entre aplicaciones, de forma que puedan comunicarse independientemente de su plataforma operativa en tiempo real. Muchas de las compañías líderes del mercado de software han concebido un nuevo método mediante el cual





solucionar el problema de la heterogeneidad e interoperabilidad entre aplicaciones web mediante el diseño y mantenimiento de los Web Services. "Un Web Service es un componente de software que se comunica con otras aplicaciones mediante la codificación de los mensaje en XML y el envío de estos mensaje con protocolos estándares de Internet como el Hyper -Text Transfer Protocol (HTTP)". 41

El desarrollo de servicios web o Web Services como aplicaciones en entornos distribuidos y descentralizados, básicamente utiliza las siguientes tecnologías:

- Un formato que describa la interfaz del componente (sus métodos y atributos) basado en XML. Por lo general, este formato es el WSDL (Web Service Description Language ).
- Un protocolo de aplicación basado en mensajes y que permita que una aplicación interactúe (use, llame, ejecute) con el Web Service. Por lo general, este protocolo es SOAP ( Simple Object Access Protocol ).
- Estructuras parecidas a las bases de datos que organizan los servicios web; así como especificaciones que definen un camino para publicar y descubrir información sobre los servicios Web mediante el estándar UDDI (Universal Description, Discovery and Integration).
- Un protocolo de transporte que se encargue de transportar los mensajes por Internet. Por lo general, este protocolo de transporte es HTTP (Hyper-Text Transfer Protocol ) que es exactamente el mismo que se usa para navegar por el web.

Los Web Services no son, por tanto, aplicaciones con una interfaz gráfica con los que las personas puedan interactuar, sino que son software accesibles en Internet o en redes privadas que utilizan tecnologías de Internet en lugar de otras aplicaciones. De esta forma, pueden desarrollarse aplicaciones que hagan uso de otras, que estén disponibles en Internet e interactúen con ellas.

En este sentido, un aspecto de suma importancia para la puesta en marcha y mejora de la web semántica es la existencia de herramientas de soporte y la re-utilización de ontologías que puedan utilizar los Web Services para recuperarlas, mediante una plataforma que una entornos de desarrollo de aplicaciones para la web semántica y un módulo de conexión a servicios web para la búsqueda y recuperación de ontologías.

Definido sobre esta base, empresas como Hewlet Packard y la National University of Ireland trabajan desde el año 2002 en una plataforma orientada al descubrimiento, descripción y mediación entre Web Services en el marco de la web semántica conocida como Semantic Web Enabled Web Services . Además, el Computer Science and Electrical Engineering Department de la Universidad de Maryland, desarrolla un proyecto de investigación iniciado en el año 2004 con un período de duración de dos años, que consiste en un buscador temático llamado Swoogle, con una interfaz de consulta muy sencilla que recupera ontologías codificadas en RDF, OWL y DAML+OIL mediante esta plataforma de intercambio de Web Services.

### ***9.10 Informática presente en la vida diaria***

---

El conjunto de pequeños dispositivos electrónicos sin hilos y portables forman el paradigma de la informática omnipresente. Constantemente aparecen nuevos aparatos como teléfonos portables multimedia con conexión a Internet, reproductores mp3, discos duros externos, aparatos de fotos digitales y en general todo tipo de dispositivos relacionados de alguna manera con los ordenadores. La tendencia (el ejemplo más claro son los teléfonos de última generación) es conseguir utilizar toda la potencia de los servicios que ofrece Internet desde cualquier punto dónde se encuentre la usuario.

Esto implica la necesidad de intercomunicar diferentes dispositivos, de fabricantes diferentes, con propósitos diferentes y utilizando tecnologías diferentes. Hará falta pues una ontología que describa las características de cada dispositivo, el tipo de acceso que utiliza, las reglas de uso definidas por el propietario y otros requerimientos técnicos relacionados con la conexión de un elemento determinado a la red ad hoc.

En este sentido se está trabajando en esquemas RDF para dispositivos como CC/PP (Composite Capability/Preference Profile) y UAProf (WAP User Agent Profiling Specification), junto con protocolos de servicios web semánticos (DAML-S).

W3C ha creado un grupo de trabajo denominado Mobile Web Initiative encargado de establecer un estándar para crear contenidos y lugares Web para dispositivos portables.



## **9.11 Buscadores semánticos**

---

Los buscadores semánticos son un ejemplo más de aplicaciones basadas en Web Semántica. El objetivo en este caso es satisfacer las expectativas de busca de información por parte de un usuario que necesita respuestas precisas.

Ejemplos de buscadores semánticos son:

- **Swoogle:** busca documentos empleando RDF y OWL, fue diseñado por la UMBC (University of Maryland Baltimore County)
- **Kartoo:** motor de busca más orientado a gráficos que a texto creado por dos primos franceses (Laurent y Nicholas Baleyrier), presenta la información en gráficos semánticos
- **Ujiko:** el mismo equipo de Kartoo presentó en 2004 el buscador Ujiko, que genera un perfil de usuario a medida que se realizan buscas permitiendo definir lugares preferidos y lugares indeseables basándose en un sistema de puntos.



## 9.12 Firmas digitales y redes de confianza

---

Uno de los componentes fundamentales de la web semántica serán las firmas digitales. Estas se definen como “bloques de datos cifrados que las computadoras y agentes de software podrán utilizar para verificar que la información adjunta fue proporcionada por una fuente fiable”.<sup>42</sup> Al aplicar un algoritmo de cifrado sobre un documento o mensaje a firmar, se obtiene un bloque de datos que representa un resumen del documento o mensaje. La firma digital se obtiene mediante un Sistema de criptografía de clave pública, que se basa en el empleo de dos tipos de claves: pública y privada, todo lo que se cifre mediante una clave pública solo puede ser descifrado a por medio de su correspondiente clave privada y viceversa.

Las ventajas que confiere la firma digital a los recursos, documentos o mensajes del web, son principalmente:

- **Identificación:** Posibilidad de determinar la identidad del emisor o autor del recurso.
- **Integridad:** Facilidad para detectar la manipulación o alteración ilícita del recurso.

La privacidad de los datos personales que se manejan en Internet es una preocupación constante para empresas, gobiernos, medios de comunicación y el público en general. Para solucionar este problema, surgieron grupos de trabajo e iniciativas dirigidas a estandarizar estos problemas como los grupos de trabajo XML Encryption y XML Signature, este último fruto del esfuerzo conjunto entre el W3C y el IETF (Internet Engineering Task Force), quienes han desarrollado una especificación, con sintaxis XML, para la representación de firmas digitales en recursos web que alcanzó el grado de Recomendación del W3C en febrero de 2002.

XML Signature asegura la integridad de partes de documentos XML transportados, así como proporciona la autenticación de mensajes y servicios de autenticación de firma para datos de cualquier tipo, tanto si se encuentra en el XML que incluye la firma o en cualquier otra parte. Puede aplicarse a cualquier contenido digital (objeto de datos), incluido XML por medio de la asociación de claves con los datos de consulta. XML Signature representa un sistema que mediante una firma digital permite ofrecer autenticidad de los datos. Con la firma digital se confirma la identidad del emisor, la autenticidad del mensaje e integridad.

En el caso de XML Encryption, es un lenguaje cuya función principal es asegurar la confidencialidad de partes de documentos XML por medio de la encriptación parcial del documento transportado. XML Encryption puede aplicarse a cualquier recurso web, incluido contenido que no es XML.

Otro de los esfuerzos orientados a la seguridad es la conocida como P3P o Plataforma de Preferencias de Privacidad (Platform for Privacy Preferences), que nace ante la necesidad de garantizar la privacidad en una web cada vez más extensa, que alcanzó el grado de Recomendación por el W3C en abril de 2002. P3P es un lenguaje estándar que ofrece a los usuarios una forma sencilla y automatizada de controlar en mayor medida el uso que se hace de su información personal en los sitios web que visitan. A su vez posibilita:

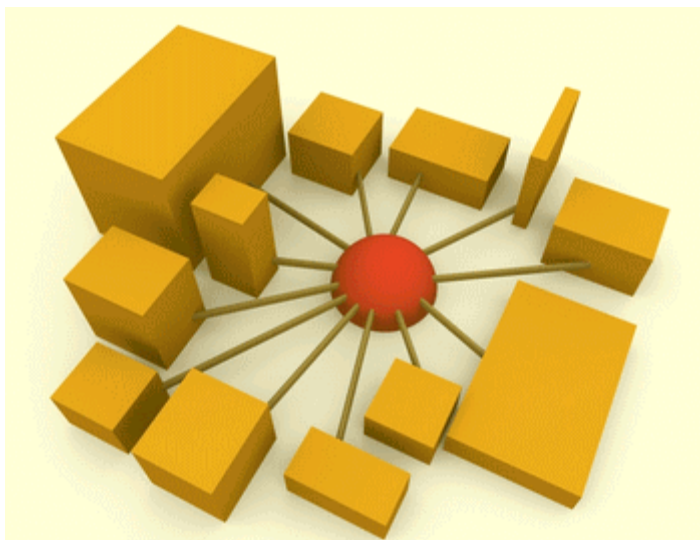
Desarrollar herramientas y servicios que ofrezcan a los usuarios un mayor control sobre la información personal que se maneja en Internet y, al mismo tiempo, aumentar la confianza entre los servicios web y los usuarios.

Mejorar el control del usuario al colocar políticas de privacidad donde los usuarios pueden encontrarlas, en un formato en el que los usuarios pueden entender y, lo más importante, con la posibilidad de que el usuario actúe sobre lo que ve.



A los usuarios web, facilidad y regularidad a la hora de decidir si quieren o no, y bajo qué circunstancia, revelar información personal.

Permite a los sitios web trasladar sus prácticas de privacidad a un formato estandarizado y procesable por dispositivos -basado en XML - que puede ser recuperado de forma automática y que además puede interpretarse fácilmente por los navegadores web. Sin embargo, “aunque las firmas digitales resultan de utilidad para describir la autenticidad de las relaciones de autoría entre individuos y recursos, no describen la confianza de los usuarios y agentes inteligentes sobre dichos autores y, por lo tanto, sobre el contenido de sus recursos”. 43



**Fig. 24: Representación de la interacción de las Redes sociales**

La solución puede estar en el concepto de una Web de Confianza que basa su funcionamiento sobre la base de la confianza o fiabilidad entre usuarios y agentes de software en la red, definido como “el grado en que un agente considera un aserto como verdadero para un contexto dado”. 44 Cada usuario, identificado por un URI, expresaría su grado de confianza o desconfianza sobre otros individuos, que a su vez harían lo mismo sobre otros, lo que daría como resultado extensas e interoperables redes de confianza procesables por agentes inteligentes.

Para la web semántica se han puesto en práctica aplicaciones que propician el análisis de redes sociales que toman como premisa un vocabulario RDF para describir información sobre los individuos y sus relaciones en una red social, conocido como FOAF (Friend OF A Friend), que se extiende con el objetivo de expresar el nivel de confianza entre los individuos de la red para finalmente mostrar su implementación práctica en áreas como el correo electrónico y la mensajería instantánea.

El uso conjunto de firmas digitales y redes de confianza, aplicadas a las reglas de inferencia con la cuales operan los agentes de software, pueden solucionar los problemas de autenticación y validación de permisos de usuarios, así como entre estos y los computadoras. El problema de la centralización de los mecanismos de seguridad puede resolverse mediante el uso generalizado de redes de confianza y bloques de datos cifrados, que harían innecesario el empleo de criptografía en la mayoría de los casos y, consecuentemente, de las Autoridades de Certificación, en un entorno caracterizado por la descentralización en el que pretende establecerse la web semántica.

### 9.13 Topic Maps

---

Un Topic Map tiene como finalidad normalizar los elementos y la notación utilizada para estructurar la información mediante la construcción de una red de enlaces semánticos que relacione diferentes recursos informativos. Tienen su origen en el grupo Davenport, un foro destinado a productores de libros electrónicos surgido a principios de la década de los 90. En 1993, se propuso la creación de una norma cuyo principal objetivo fuera posibilitar la fusión de índices impresos, que posteriormente evolucionó hacia otras estructuras (como tesauros y ontologías), hasta llegar a ser una herramienta considerada en el web para la organización, representación y gestión del conocimiento. Teóricamente, un Topic Map equivale a un índice creado para una colección de recursos disponibles en formato electrónico, cuyos elementos son:

- Una serie de tópicos o materias que se utilizan para describir el contenido de los recursos y facilitar su recuperación.
- Relaciones que establecen entre los tópicos.
- Una serie de recursos que se indizarán o describirán mediante los tópicos anteriores -ello equivale a decir que el recurso trata de o contiene información sobre ese tópico.

En tal sentido, un Topic Map sería similar a un catálogo de materias con los siguientes elementos:

- Un conjunto de encabezamientos que se pueden utilizar como descriptores.
- Relaciones explícita entre encabezamientos.
- Referencias a los libros a los que se han asignado las materias.

En la práctica, la implementación de los Topics Maps se diseñó para arquitectura SGML con notación de "HyTime" - la DTD del estándar ISO/IEC 13250:2000 perteneciente al diseño de Topics Maps se ha mantenido en HyTime-, no obstante, esta notación ha caído en desuso tras la aparición de XML. La Norma ISO /IEC 10744:1992 Information technology - Hypermedia Time-Based Structuring Language (Hytime) define el lenguaje y modelo subyacente para la representación de hiperdocumentos que enlazan y sincronizan información estática y dinámica (dependiente del tiempo) contenida en documentos convencionales o multimedia y otros objetos de información. En este caso, las causas del desplazamiento de la notación de HyTime se deben a la creación de una DTD para crear Topic Maps en XML, denominada XTM (XML Topic Map ) y al auge que ha obtenido este. En cualquier caso, actualmente pueden encontrarse ejemplos de ambos lenguajes en la web.

Un topic es el elemento principal de un Topic Map, es el término que expresa determinado concepto o idea (o subject, como se denomina en el estándar). Ejemplos de topics pueden ser "Europa", "persona", "idea" o "continente". Básicamente, los topics tienen tres características principales: su denominación (names), sus apariciones u ocurrencias (occurrences) y su función en las asociaciones (role associations). Un topic puede tener varias denominaciones, pero debe estar representado por una forma base (base name). La denominación es un elemento obligatorio y representa la forma normal de hacer mención al topic. Las apariciones son enlaces a recursos informativos, como puede ser una página web. Son elementos opcionales que son relevantes a un topic determinado, es decir, un topic puede tener cero o muchas ocurrencias.

Los recursos informativos pueden ser de muchos tipos: una cita de un texto, una definición, una ley, un artículo, etcétera. Cada uno de los diferentes tipos documentales puede agruparse mediante roles de ocurrencia (diccionario, página web, imagen). Es importante que la mayoría de estas ocurrencias suelen ser externas al Topic Map, una situación análoga a la que siempre ha existido entre un tesauro y la indización documental realizada con ese tesauro.



La asociación es un enlace que establece una relación entre dos o más topics. Una forma de verlo es tomar una frase y considerar los sustantivos de esas frases como topics y emplear los términos de unión entre los topics para denominar la asociación como por ejemplo: "las piñas están formadas por piñones"

Ahora bien, la especificación Topic Map se completa con un mecanismo de seriado, que se refiere a cómo codificarlo y escribirlo por medio de un formato legible y procesable por máquina. En este sentido, se han propuesto dos alternativas. Inicialmente, la norma ISO 13250 definió una DTD en SGML para su codificación, intercambio y procesamiento. Posteriormente, la generalización de XML hizo que se adaptase la DTD inicial a este lenguaje.



Fig. 25: Representación gráfica del metabuscador Kartoo

El resultado de este trabajo fue la especificación XML Topic Map, publicada en febrero de 2001 y, a su vez, la norma ISO 13250 fue revisada para incluir, como parte del estándar, la versión XML de la DTD. Ambos proyectos se centran en la definición conceptual de los Topics Maps y en su representación con lenguajes de marcas. "Actualmente, se encuentran en fase de desarrollo algunos estándares para la consulta y representación de los Topic Maps":

- La especificación TMQL (Topic Map Query Language): tiene como propósito definir un lenguaje para interrogar el contenido de los Topics Maps.
- La especificación TMCL (Topic Map Constraint Language): permitirá definir restricciones sobre el contenido expresado en el Topic Map.

El estado actual de estas especificaciones se puede consultar en: [www.isotopicmaps.org](http://www.isotopicmaps.org)

"Los Topic Maps constituyen una de las principales propuestas para la visualización de la web semántica". Existe una problemática obvia en este tema cuando se tiene en cuenta que un Topic Map puede tener cientos de miles de asociaciones de diferentes tipologías. Actualmente, entre las representaciones propuestas destacan los árboles, los browsers o navegadores y los gráficos. A la hora

de analizar estas representaciones, se debe considerar que muestren, tanto información local de los topics que interesan al usuario, como información sobre la localización de esos topics en su conjunto. En el caso de los gráficos, por ejemplo, se muestra un nodo principal y los nodos más próximos.

Por otra parte, los Topic Maps se proponen para modelar redes semánticas. Por ejemplo, si se establece una relación entre un Topic Map y un tesoro se puede apreciar que los tesauros no son más que un caso particularmente simple del primero, donde tan solo existen tres tipos de relaciones (jerarquía, sinonimia y relacionado).

Al definir el modelado inicial del Topic Map y un tesoro, las semejanzas no son tan evidentes. Una de las diferencias con los tesauros está más en el momento histórico en que cada propuesta tuvo lugar, así los tesauros son normalmente productos incorporados a una única organización de forma centralizada y cuya presentación en formato electrónico es sólo una característica extra, pero no un requisito. Los Topics Maps son productos descentralizados que mejoran con la cooperación de distintas organizaciones y que tienen esencialmente un formato electrónico.

Asimismo, los Topic Maps tienen una estrecha conexión con RDF porque, según Le Grand y otros autores, ambos están destinados a ser complementarios. La diferencia es que un Topic Map es independiente de que existan recursos de información o no, y permite en ocasiones una navegación entre conceptos. Por otra parte RDF, según Sigel, está más centrado en la descripción de recursos de información y no trata la semántica de las relaciones entre los meta-datos. Por lo tanto, las ventajas que tienen los Topics Maps para la representación del conocimiento en la web semántica, se resumen de la siguiente manera:

- Los Topic Maps pueden dar una semántica a elementos que están en el web, al organizarlos y describirlos, pero sin modificarlos.
- Presencia de perfiles de usuarios, p ermiten adaptarse a distintas comunidades para compartir recursos de información.
- Navegabilidad e inferencia mediante estructuras semánticas: Mejora no sólo la recuperación de información, sino también la gestión del conocimiento y el mantenimiento de los Topic Maps. En este punto, es también interesante recordar la independencia de los recursos de información que tiene un Topic Map, al poder navegar por nodos que no tengan ningún recurso asociado
- Fusión con otras estructuras de conocimiento, que permite una gestión descentralizada.
- Buena escalabilidad y compatibilidad para adaptarse al creciente número de recursos de información.

Entre las posibles desventajas, residen más en la poca madurez del estándar que en otros aspectos. Así se han revisado distintos aspectos como:

- La necesidad de mejorar la capacidad de integrar las propiedades de las dentro del esquema de definición de Topic Maps.
- La línea de investigación emergente sobre la forma de asegurar la consistencia de los Topic Maps.
- Las restricciones necesarias relativas a su validación y corrección.

El esfuerzo que supone la creación de un Topic Map realmente se compensa si se piensa en las posibilidades que ofrecen en cuanto a capacidad de indización a texto completo delegados a aplicaciones informáticas de carácter distribuido. Si bien estas aplicaciones son poco conocidas en nuestro ámbito académico, constituyen buenos ejemplos del potencial que ofrecen para la organización y la recuperación de información.





Algunas de ellas, como

- Ontopia (<http://www.ontopia.net>)
- Empolis (<http://www.empolis.com>)
- Mondeca ( <http://www.mondeca.com>)
- Infoloom ( <http://www.infoloom.com>)

comparten características y funcionalidades; en todos los casos los Topics Maps se utilizan como una interfaz para indizar, interrogar y navegar repositorios de información.



## 9.14 Otro tipo de aplicaciones

---

Se pueden encontrar otros ejemplos de aplicaciones basadas en la Web Semántica en el sitio Web de W3C: SWAD-Europe: Semantic web applications - analysis and selection.

También se recomienda consultar el WSIndex (Web Services Directory): un portal de información sobre los nuevos servicios web que incluye múltiples enlaces relacionados con XML, Web Semántica, Weblogs, Web Services, etc.

<b>Catalogue/thesaurus management</b>
Catalogue Management
Catalogue integration
DMOZ - Directory Mozilla - open directory
Thesaurus management
<b>Data dependent agents</b>
Financial Assistant
ITTalks
Jema
Shopping assistants
Virtual Travel Agent
<b>Data integration</b>
B2B web service mediation
Catalogue integration
Database integration example
Financial Portals
Gene Ontology
Mozilla
eScience Data Grids
<b>Knowledge formation</b>
Assumption tracker
Bibliography workbench
ClaiMaker/Scholonto
Community Arkive
Community formation
DMOZ - Directory Mozilla - open directory
Ideas workbench
SWAP - semantic web and peer-to-peer
<b>Knowledge management</b>
Community formation
Community portals
Helpdesk support



OntoShare - community of practice support
PatMan
PlanetOnto
Sun GKE
ePerson
<b>Metadata for annotation and enrichment</b>
Annotea
Assumption tracker
Bibliography workbench
Community Arkive
Community bookmarking
Distributed topic portals
EARL
Gene Ontology
MIT/HP SIMILE project
<b>Metadata for discovery and selection</b>
B2B trading market-places
DCMI registry
Edutella
HP Portal
MIT/HP SIMILE project
MUSE
Recommendation Networks
Scholnet
SeLeNe
Semantic tagging
Sun GKE
Web service description and discovery
<b>Metadata for media and content</b>
Adobe XMP
Arkive internal
MIT/HP SIMILE project
<b>Ontology Management</b>
APECKS
Gene Ontology
KAON
SWAP - semantic web and peer-to-peer
<b>Personal information management</b>
Bibliography workbench
Community bookmarking

Event tracking
Genealogy assistant
Haystack
Ideas workbench
Jema
Mozilla
ePerson
<b>Semantic indexing</b>
Community Arkive
Community portals
Context aware links
Curriculum Online
Distributed topic portals
HP Portal
ITTalks
Museum portals
MusicBrainz
PlanetOnto
Score
Semantic tagging
TAP semantic search
<b>Syndication Category</b>
Event tracking
Rich Site Summary/RDF Site summary
Syndication

**Tabla 50: Listado General de aplicaciones RDF según la base SWAD-Europe**



## 10 Análisis de las tecnologías de Web Semántica

---

### 10.1 Búsquedas en la Web Semántica

---

#### 10.1.1 La cuestión de la búsqueda en la Web Semántica

---

Cuando se busca información en sistemas de ayuda existentes, sistemas de gestión documental o en Internet, no se obtienen respuestas como se esperaría, sino documentos. Los sistemas actuales recuperan los documentos que pueden o no contener las respuestas a las preguntas formuladas, basándose en apariciones de palabras clave, sus frecuencias, u otras heurísticas propias del área de recuperación de información. Actualmente, en una sociedad cada vez más basada en la información y en el conocimiento, estos sistemas tienden a cambiar hacia soluciones que encuentren y formulen respuestas concretas, que aprendan de los usuarios u ofrezcan la información solicitada de manera visual y fácil de entender. Existe una gran demanda, sobre todo por parte de profesionales no informáticos, de sistemas que les permitan localizar y manipular información concisa, sin tener que expresar sus preguntas y criterios de búsquedas en lenguajes crípticos.

Esta revolución en el campo de recuperación o extracción de la información viene alimentada por las mejoras en el hardware informático, capaz de manejar grandes cantidades de datos de manera efectiva, por la aparición de tecnologías de gestión de conocimiento que permiten comprender la manera en la que los humanos manejan y buscan información así como por tecnologías que permiten que los ordenadores interpreten nuestro lenguaje y a su vez aprendan a adaptarse a nuestras necesidades de manera dinámica.

La tecnología actual empleada en sistemas de gestión documental permite almacenar grandes cantidades de documentos que se clasifican por unas pocas características predefinidas. Estos sistemas pueden incluir un servicio de publicación del contenido en formato de la WWW. La funcionalidad principal ofrecida a los usuarios finales es el listado de los documentos de acuerdo a criterios de clasificación y su recuperación por palabras claves. La calidad de este servicio depende en gran medida en el esfuerzo invertido en la codificación o clasificación previa y manual del documento. Queda como responsabilidad del usuario codificar el contenido del documento de acuerdo a los metadatos y palabras clave disponibles.

Los buscadores simples son ya un estándar y no requieren grandes esfuerzos ni en codificación ni tampoco en adecuación y provisión del contenido. Algunos buscadores avanzados, como Google [Google], tienen también en cuenta la estructura hipertextual de la red y la usan para ordenar los resultados de acuerdo a una reputación o calculada para cada documento. En la Ilustración 52 se puede observar cuál es la relación entre el esfuerzo invertido en la preparación de los contenidos (su codificación) y el esfuerzo en la codificación de las propias aplicaciones. Las aplicaciones están a su vez ordenadas por el grado de sofisticación en sus funcionalidades. Se puede observar, por ejemplo, los buscadores por palabras clave a penas necesitan gastar esfuerzo en la preparación del contenido ofreciendo un nivel bajo en cuanto a funcionalidades. De la misma manera Google que ofrece mejores resultados de búsqueda tiene aumentada la necesidad de esfuerzo en codificación de la aplicación ya que el contenido se tiene que procesar (algoritmo de reputación de páginas: PageRank<sup>TM</sup>). El caso de la aplicación de bibliografía en el ámbito científico CiteSeer [CiteSeer] es más pronunciado. Procesa publicaciones científicas en diversos formatos y extrae información sobre los autores, bibliografía, palabras clave, etc. El esfuerzo en implementación de una heurística de análisis de documentos es grande.

Si embargo, en el marco de la Web Semántica, con todo el contenido generado de acuerdo a modelos semánticos consensuados el esfuerzo requerido se centrará en la lógica de negocio de la aplicación en vez de su provisión.

Los nuevos sistemas de gestores documentales semánticos que están emergiendo constituyen soluciones construidas sobre sistemas de análisis y comprensión de lenguaje muy cercano al humano, y que permiten extraer el significado de las preguntas formuladas, contrastándolas contra un modelo lógico que delimita el dominio con información semántica y lingüística.

### 10.1.2 Métrica definible para la Web Semántica

---

La búsqueda semántica ha sido uno de los beneficios esperados de la Web Semántica desde su emergencia a finales de los 90. Una forma de entender un motor de búsqueda semántica es como una herramienta que recibe consultas basadas en ontologías (p.e. en RDQL, RQL, SPARQL, etc.), las ejecuta contra una base de conocimiento, y devuelve tuplas que satisfacen la consulta.

Estas técnicas típicamente utilizan modelos booleanos de búsqueda, basados en una visión ideal del espacio de información, consistentes en piezas formales de conocimiento ontológico sin ambigüedad ni redundancia.

Los llamados portales semánticos típicamente proporcionan funcionalidades sencillas de búsqueda que más podrían caracterizarse como recuperación semántica de datos que como recuperación semántica de información. Las búsquedas devuelven instancias de una ontología más que documentos y por lo general no se proporciona un método de ranking. En algunos sistemas, se añaden enlaces a documentos referenciados por las instancias, junto a cada instancia devuelta en la respuesta a la consulta, pero ni las instancias ni los documentos están ordenados por relevancia.

Junto con TAP, KIM es una de las propuestas más completas publicadas hasta la fecha, en nuestro conocimiento, para la construcción de Bases de Conocimientos y la anotación automática a gran escala. Existen otros trabajos complementarios al de KIM y TAP basados en un algoritmo de ranking específicamente diseñado para un modelo de recuperación basado en ontologías, utilizando un sistema de indexado semántico basado en la ponderación de las anotaciones.

Ahondando en esta perspectiva, un elemento de conocimiento es una respuesta o bien correcta o bien incorrecta procedente de una petición de información, y por ende los resultados de la búsqueda se suponen siempre 100% precisos, de forma que no se contempla la noción de respuesta aproximada a una necesidad de información. Si bien esta concepción de la búsqueda semántica aporta ventajas fundamentales. Otra forma de ver la recuperación de información en la Web Semántica, un motor de búsqueda devuelve documentos, más que (o además de) valores exactos, en respuesta a las consultas del usuario. Más aún, y como requisito clave para la escalabilidad hacia fuentes masivas de información, el motor debe ordenar los documentos de acuerdo con criterios de relevancia basados en las ontologías.

Un modelo de recuperación basado en ontologías puramente booleano tiene sentido cuando el corpus de información puede ser completamente representado como una base de conocimiento formal, de manera que los resultados de las búsquedas consisten en entidades de la ontología. Pero, como es bien conocido, existen límites respecto al punto hasta donde el conocimiento se puede formalizar de este modo. En primer lugar, debido al enorme volumen de información disponible hoy día en forma de texto y contenidos multimedia no estructurados, convertir esta cantidad de información en



conocimiento ontológico con un coste viable es un problema sin resolver en general. En segundo lugar, los documentos tienen un valor por sí mismos, y no son equivalentes a la suma de sus partes. Aunque es útil descomponer documentos en unidades de información menores que puedan ser reutilizadas y ensambladas para diferentes del modelo vectorial clásico de recuperación de información, adecuada para una representación basada en ontologías, sobre la cual se puede definir un algoritmo de ranking. El rendimiento de nuestro modelo está en relación directa con la cantidad y calidad de la información en la BC sobre la que opere.

Los últimos avances en la automatización del poblado de ontologías y la anotación semiautomática de textos son prometedores. Mientras, si es que algún día sucede, las ontologías y metadatos (y la propia Web Semántica) llegan a ser un recurso de disponibilidad común, la falta o incompletitud de las ontologías y Bases de Conocimientos disponibles será una limitación que muy probablemente tendremos que admitir a medio plazo. En consecuencia, la tolerancia a Bases de Conocimientos incompletas se establece como un importante requisito en nuestra propuesta.

### **10.1.3 Resumen del estado del arte del análisis de la Web Semántica**

---

Típicamente, no existe una unicidad de criterio en el modelo de análisis de ranking o de estadística de la Web Semántica.

Los algoritmos o las métricas desarrollados al caso son numerosos y, si bien el objetivo de este TFC no es analizar la bondad, la complejidad o la eficiencia de los mismos, sí se evalúa su implantación, de forma que se puede indicar que la capacidad para evaluar los datos de accesos y demás modelos estadísticos desarrollables, están no en función de parámetros generales estandarizados, sino en función de las características implementadas por cada aplicación en uso en Internet.

Es por esta razón que resultaría casi inviable en el estado actual proceder a una valoración correcta y objetiva de los resultados a nivel global, ya que los rankings encontrados o bien son de utilidad muy limitada o bien son estadísticas basadas en el acceso únicamente a la Base de Conocimiento disponible solo por esa aplicación. Estos resultados no son homologables, ya que requeriría un análisis de la compatibilidad de los datos, de los algoritmos de acceso y de las valoraciones asignadas a cada acceso a enlace, frase o palabra.

Finalmente, y por estas razones anteriormente indicadas, se ha considerado solamente la captura de datos formales de los sitios más relevantes en la Red, de forma que, al menos, se pueda tener una idea clara del nivel estadístico actual y posibilitar una pequeña comparativa, al nivel permitido por los datos que proporcionan dicha aplicaciones.

#### 10.1.4 Diversos Análisis Realizados sobre la Web Semántica

##### PTSW (Ping the Semantic Web)

Este es el número de 'typed entities' definidas en cada recurso rdf conocido por el servicio PTSW (Ping the Semantic Web) [tabla 51]. Por ejemplo, si un recurso RDF tiene 4 foaf:Person definidos, entonces el valor 4 será añadido al contador. Si la misma entidad (entity-> URI) es definida en dos recursos diferentes RDF, el tipo de la entidad (entity type) será calculado dos veces.

Es preciso tomar estos valores calculados como una aproximación buena, pero no con una fiabilidad absoluta.

Entity types (6755 known entity types)	Count
<a href="http://xmlns.com/foaf/0.1/Person">http://xmlns.com/foaf/0.1/Person</a>	18.213.663
<a href="http://xmlns.com/foaf/0.1/Document">http://xmlns.com/foaf/0.1/Document</a>	4.562.658
<a href="http://blogs.yandex.ru/schema/foaf/Posts">http://blogs.yandex.ru/schema/foaf/Posts</a>	583.494
<a href="http://www.w3.org/2003/01/geo/wgs84_pos#Point">http://www.w3.org/2003/01/geo/wgs84_pos#Point</a>	149.746
<a href="http://rdfs.org/sioc/ns#Post">http://rdfs.org/sioc/ns#Post</a>	137.728
<a href="http://semantic-mediawiki.org/swivt/1.0#Subject">http://semantic-mediawiki.org/swivt/1.0#Subject</a>	128.996
<a href="http://xmlns.com/foaf/0.1/Agent">http://xmlns.com/foaf/0.1/Agent</a>	116.508
<a href="http://rdfs.org/sioc/ns#Forum">http://rdfs.org/sioc/ns#Forum</a>	102.997
<a href="http://rdfs.org/sioc/ns#Site">http://rdfs.org/sioc/ns#Site</a>	101.647
<a href="http://rdfs.org/sioc/ns#Usergroup">http://rdfs.org/sioc/ns#Usergroup</a>	101.536
<a href="http://rdfs.org/sioc/types#ArgumentativeDiscussion">http://rdfs.org/sioc/types#ArgumentativeDiscussion</a>	101.139
<a href="http://www.w3.org/2002/07/owl#Class">http://www.w3.org/2002/07/owl#Class</a>	85.870
<a href="http://xmlns.com/foaf/0.1/OnlineAccount">http://xmlns.com/foaf/0.1/OnlineAccount</a>	85.790
<a href="http://www.w3.org/2002/07/owl#AnnotationProperty">http://www.w3.org/2002/07/owl#AnnotationProperty</a>	69.356
<a href="http://www.w3.org/2002/07/owl#ObjectProperty">http://www.w3.org/2002/07/owl#ObjectProperty</a>	65.180
<a href="http://www.w3.org/2002/07/owl#DatatypeProperty">http://www.w3.org/2002/07/owl#DatatypeProperty</a>	56.484
<a href="http://usefulinc.com/ns/doap#Version">http://usefulinc.com/ns/doap#Version</a>	41.489
<a href="http://smw.ontoware.org/2005/smw#Thing">http://smw.ontoware.org/2005/smw#Thing</a>	34.659
<a href="http://www.w3.org/2002/07/owl#Ontology">http://www.w3.org/2002/07/owl#Ontology</a>	30.948
<a href="http://purl.org/rss/1.0/item">http://purl.org/rss/1.0/item</a>	27.819
<a href="http://xmlns.com/foaf/0.1/PersonalProfileDocument">http://xmlns.com/foaf/0.1/PersonalProfileDocument</a>	23.008
<a href="http://usefulinc.com/ns/doap#Project">http://usefulinc.com/ns/doap#Project</a>	16.888
<a href="http://rdfs.org/sioc/ns#User">http://rdfs.org/sioc/ns#User</a>	15.904
<a href="http://xmlns.com/foaf/0.1/Image">http://xmlns.com/foaf/0.1/Image</a>	13.032
<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property">http://www.w3.org/1999/02/22-rdf-syntax-ns#Property</a>	10.967
<a href="http://usefulinc.com/ns/doap#CVSRepository">http://usefulinc.com/ns/doap#CVSRepository</a>	10.695
<a href="http://www.geonames.org/ontology#Feature">http://www.geonames.org/ontology#Feature</a>	9.168





<a href="http://atomowl.org/ontologies/atomrdf#Entry">http://atomowl.org/ontologies/atomrdf#Entry</a>	8.914
<a href="http://xmlns.com/foaf/0.1/Group">http://xmlns.com/foaf/0.1/Group</a>	8.422
<a href="http://purl.org/dc/dcmitype/Image">http://purl.org/dc/dcmitype/Image</a>	7.444
<a href="http://atomowl.org/ontologies/atomrdf#Feed">http://atomowl.org/ontologies/atomrdf#Feed</a>	7.070
<a href="http://82.204.47.68:80/resource/vocab/survey.observations">http://82.204.47.68:80/resource/vocab/survey.observations</a>	6.273
<a href="http://usefulinc.com/ns/doap#SVNReposito...">http://usefulinc.com/ns/doap#SVNReposito...</a>	5.783
<a href="http://www.techpresentations.org/Special:URIResolver/Category-3ARFC">http://www.techpresentations.org/Special:URIResolver/Category-3ARFC</a>	5.076
<a href="http://www.w3.org/2002/07/owl#Restriction">http://www.w3.org/2002/07/owl#Restriction</a>	4.739
<a href="http://atomowl.org/ontologies/atomrdf#Link">http://atomowl.org/ontologies/atomrdf#Link</a>	4.518
<a href="http://rdfs.org/sioc/types#Comment">http://rdfs.org/sioc/types#Comment</a>	4.477
<a href="http://purl.org/rss/1.0/channel">http://purl.org/rss/1.0/channel</a>	4.462
<a href="http://www.w3.org/2000/01/rdf-schema#Class">http://www.w3.org/2000/01/rdf-schema#Class</a>	3.884
<a href="http://mobi.yaco.es/andalucia.rdf/andalucia-tourism.owl#Alojamiento">http://mobi.yaco.es/andalucia.rdf/andalucia-tourism.owl#Alojamiento</a>	3.526
<a href="http://www.w3.org/2004/12/q/contentlabel#ContentLabel">http://www.w3.org/2004/12/q/contentlabel#ContentLabel</a>	3.517
<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq">http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq</a>	3.350
<a href="http://purl.org/vocab/bio/0.1/Birth">http://purl.org/vocab/bio/0.1/Birth</a>	3.181

**Tabla 51: Numero de 'entities types' conocidos en el servicio PTSW**

Namespaces (535 know namespaces)	Number of documents
<a href="http://xmlns.com/foaf/0.1/">http://xmlns.com/foaf/0.1/</a>	958.665
<a href="http://blogs.yandex.ru/schema/foaf/">http://blogs.yandex.ru/schema/foaf/</a>	583.150
<a href="http://www.w3.org/2003/01/geo/wgs84_pos#">http://www.w3.org/2003/01/geo/wgs84_pos#</a>	150.194
<a href="http://rdfs.org/sioc/ns#">http://rdfs.org/sioc/ns#</a>	113.917
<a href="http://rdfs.org/sioc/types#">http://rdfs.org/sioc/types#</a>	102.534
<a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a>	35.063
<a href="http://semantic-mediawiki.org/swivt/1.0#">http://semantic-mediawiki.org/swivt/1.0#</a>	26.830
<a href="http://usefulinc.com/ns/doap#">http://usefulinc.com/ns/doap#</a>	16.746
<a href="http://www.geonames.org/ontology#">http://www.geonames.org/ontology#</a>	5.198
<a href="http://smw.ontoware.org/2005/smw#">http://smw.ontoware.org/2005/smw#</a>	4.248
<a href="http://sydneydirectory.org/index.php/Special:URIResolver/">http://sydneydirectory.org/index.php/Special:URIResolver/</a>	4.217
<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>	3.490
<a href="http://purl.org/rss/1.0/">http://purl.org/rss/1.0/</a>	3.457
<a href="http://purl.org/vocab/bio/0.1/">http://purl.org/vocab/bio/0.1/</a>	3.170
<a href="http://www.w3.org/2004/12/q/contentlabel#">http://www.w3.org/2004/12/q/contentlabel#</a>	2.867
<a href="http://ontoworld.org/wiki/Special:URIResolver/">http://ontoworld.org/wiki/Special:URIResolver/</a>	2.802
<a href="http://semanticweb.org/id/">http://semanticweb.org/id/</a>	2.310
<a href="http://wecowi.net/view/Spezial:URIResolver/">http://wecowi.net/view/Spezial:URIResolver/</a>	1.880
<a href="http://http://wiki.creativecommons.org/Special:URIResolver/">http://http://wiki.creativecommons.org/Special:URIResolver/</a>	1.581
<a href="http://web.resource.org/cc/">http://web.resource.org/cc/</a>	1.200
<a href="http://www.naturheilkraut.com/index.php/Spezial:URIResolver/">http://www.naturheilkraut.com/index.php/Spezial:URIResolver/</a>	1.173
<a href="http://atomowl.org/ontologies/atomrdf#">http://atomowl.org/ontologies/atomrdf#</a>	1.124
<a href="http://fr.jurispedia.org/index.php/Special:URIResolver/">http://fr.jurispedia.org/index.php/Special:URIResolver/</a>	830
<a href="http://www.daml.org/2001/10/html/airport-ont#">http://www.daml.org/2001/10/html/airport-ont#</a>	713
<a href="http://www.icra.org/rdfs/vocabularyv03#">http://www.icra.org/rdfs/vocabularyv03#</a>	630
<a href="http://dbpedia.org/class/yago/">http://dbpedia.org/class/yago/</a>	626
<a href="http://xmlns.com/wordnet/1.6/">http://xmlns.com/wordnet/1.6/</a>	544
<a href="http://www.techpresentations.org/Special:URIResolver/">http://www.techpresentations.org/Special:URIResolver/</a>	543
<a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>	490
<a href="http://purl.org/dc/dcmitype/">http://purl.org/dc/dcmitype/</a>	449
<a href="http://www.w3.org/1999/">http://www.w3.org/1999/</a>	430
<a href="http://www.w3.org/2004/02/skos/core#">http://www.w3.org/2004/02/skos/core#</a>	379
<a href="http://spiele.j-crew.de/wiki/Spezial:URIResolver/">http://spiele.j-crew.de/wiki/Spezial:URIResolver/</a>	367
<a href="http://purl.org/stuff/rev#">http://purl.org/stuff/rev#</a>	320
<a href="http://www.w3.org/2002/12/cal/ical#">http://www.w3.org/2002/12/cal/ical#</a>	307
<a href="http://acronym.deri.org/schema#">http://acronym.deri.org/schema#</a>	305
<a href="http://norman.walsh.name/knows/taxonomy#">http://norman.walsh.name/knows/taxonomy#</a>	289



<a href="http://owl.mindswap.org/2003/ont/owlweb.rdf#">http://owl.mindswap.org/2003/ont/owlweb.rdf#</a>	289
<a href="http://www.mindswap.org/~glapizco/technical.owl#">http://www.mindswap.org/~glapizco/technical.owl#</a>	289
<a href="http://www.holygoat.co.uk/owl/redwood/0.1/tags/">http://www.holygoat.co.uk/owl/redwood/0.1/tags/</a>	288
<a href="http://en.jurispedia.org/index.php/Special:URIResolver/">http://en.jurispedia.org/index.php/Special:URIResolver/</a>	285
<a href="http://www.isi.edu/webcrawler/bibtex.o.daml#">http://www.isi.edu/webcrawler/bibtex.o.daml#</a>	243
<a href="http://ontoworld.org/index.php/">http://ontoworld.org/index.php/</a>	210

Tabla 52: Numero de 'namespaces' conocidos en el servicio PTSW

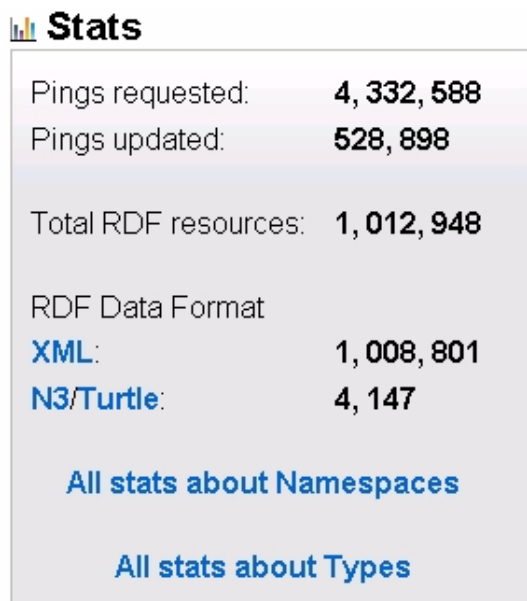


Fig. 26: Estadísticas generales de PTSW

El servicio Pingthesemanticweb.com permite mucha mayor versatilidad en el servicio de gestión de estadísticas y solicitudes de consulta a su base de datos semántica. Para ello tiene un sistema API para la construcción de consultas especializadas [<http://pingthesemanticweb.com/api.php>]:

```
POST /ping/xmlrpc HTTP/1.0
User-Agent: Java
Host: rpc.pingthesemanticweb.com
Content-Type: text/xml
Content-length: 250
<?xml version="1.0"?>
<methodCall>
  <methodName>weblogUpdates.ping</methodName>
  <params>
    <param>
      <value>Some title page</value>
    </param>
    <param>
```

```
<value>http://www.talkdigger.com/conversations/some_conversation.html</value>
</param>
</params>
</methodCall>
```

**Fig. 27: Ejemplo de XML-RPC Request**

```
HTTP/1.1 200 OK
Date: Fri, 11 Aug 2006 15:48:54 GMT
Server: Apache/1.3.36 (Unix)
X-Powered-By: PHP/5.1.4
Connection: close
Content-Length: 383
Content-Type: text/xml

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ferror</name>
            <value>
              <boolean>0</boolean>
            </value>
          </member>
          <member>
            <name>message</name>
            <value>Thanks for pinging PingtheSemanticWeb.com</value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

**Fig. 28: Ejemplo de XML-RPC Response**



## Swoogle

Swoogle Today		
admin_dt	31/05/2008 23:58	Datetime Watched
url_total	7.502.770	Number of URLs being discovered
url_pinged	4.097.505	Number of URLs being pinged
total_swd	2.558.045	Number of Semantic Web Documents (regardless of embedded or containing some errors) be confirmed.
total_swd_strict	1.444.221	Number of error-free pure Sematic Web Documents
total_swd_embed	900.849	Number of documents (except SWDs, PDF, and JPEG) embedding Semantic Web Data
triple_total	645.013.694	Number of triples could be parsed from all Semantic Web Documents.

**Tabla 53: Estadísticas generales del servicio Swoogle para la Web Semántica**

## SemanticWeb.org

There are 7,032 total pages in the database. This includes "talk" pages, pages about semanticweb.org, minimal "stub" pages, redirects, and others that probably do not qualify as content pages. Excluding those, there are 4,130 pages that are probably legitimate content pages.

177 files have been uploaded.

There have been a total of 5,350,755 page views, and 42,058 page edits since semanticweb.org was setup. That comes to 5.98 average edits per page, and 127.22 views per edit.

The job queue length is 0.

### User statistics

There are 1,256 registered users, of which 15 (or 1.19%) have Sysops rights.

### Most viewed pages

- Main Page (280.280)
- Semantic MediaWiki (102.460)
- Events (25.167)
- Help:Editing (23.391)
- Semantic Wiki State Of The Art (22.787)
- Help:Inline queries (22.593)
- Help:Semantics (22.522)
- People (20.816)
- Category:Person (19.875)
- Help:Annotation (19.524)

**Tabla 54: Estadísticas generales del servicio SemanticWeb.org**

### 10.1.5 Estadísticas de la Web Semántica en www.redwebsemantica.es

Sumario					
	Primera visita	Sumario			Última visita
	19 Feb 2007 - 12:18	Año 2007			31 Dic 2007 - 19:08
	Visitantes distintos	Número de visitas	Páginas	Solicitudes	Ancho de banda
Trafico visto *	<= <b>2356</b> Valor exacto no disponible en la vista por años	<b>3164</b> (1.34 Visitas/Visitante)	<b>4748</b> (1.5 páginas/visita)	<b>8170</b> (2.58 solicitudes/visita)	<b>30.60 MB</b> (9.9 KB/visita)
Trafico no visto *			<b>14802</b>	<b>15602</b>	<b>4.20 MB</b>

Fig. 29: Sumario accesos año 2007

Sumario					
	Primera visita	Sumario			Última visita
	01 Ene 2008 - 01:23	Año 2008			02 Jun 2008 - 02:57
	Visitantes distintos	Número de visitas	Páginas	Solicitudes	Ancho de banda
Trafico visto *	<= <b>2290</b> Valor exacto no disponible en la vista por años	<b>2784</b> (1.21 Visitas/Visitante)	<b>4023</b> (1.44 páginas/visita)	<b>4902</b> (1.76 solicitudes/visita)	<b>10.29 MB</b> (3.78 KB/visita)
Trafico no visto *			<b>57865</b>	<b>57944</b>	<b>17.60 MB</b>

Fig. 30: Sumario accesos año 2008

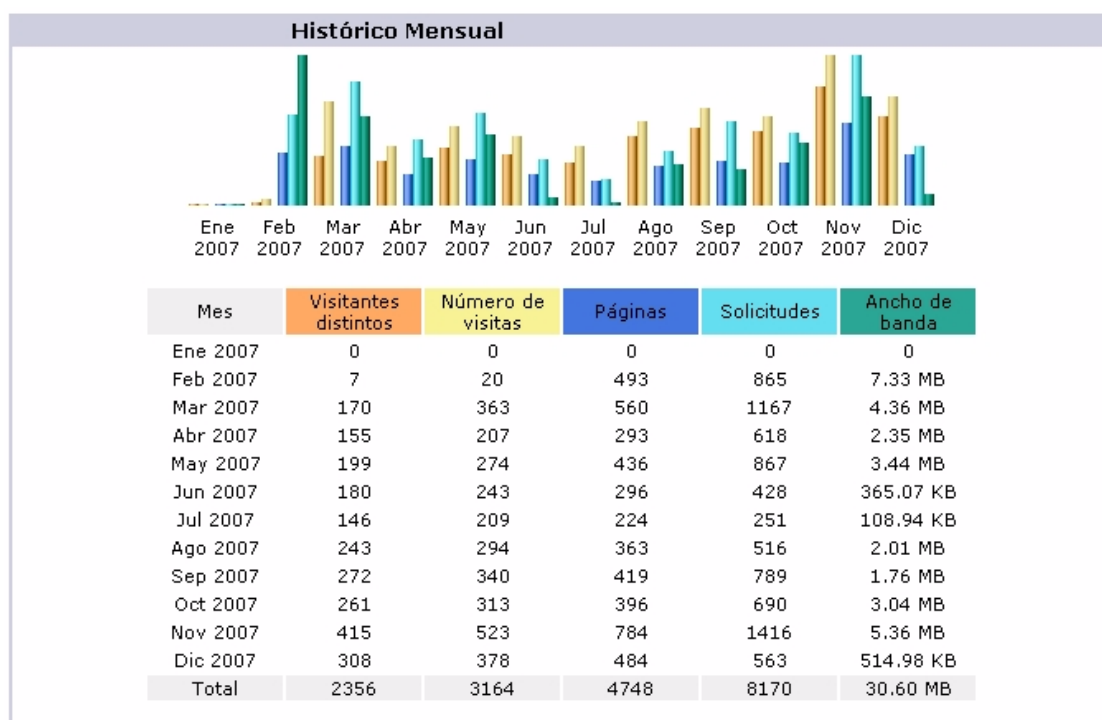


Fig. 31: Histórico mensual 2007



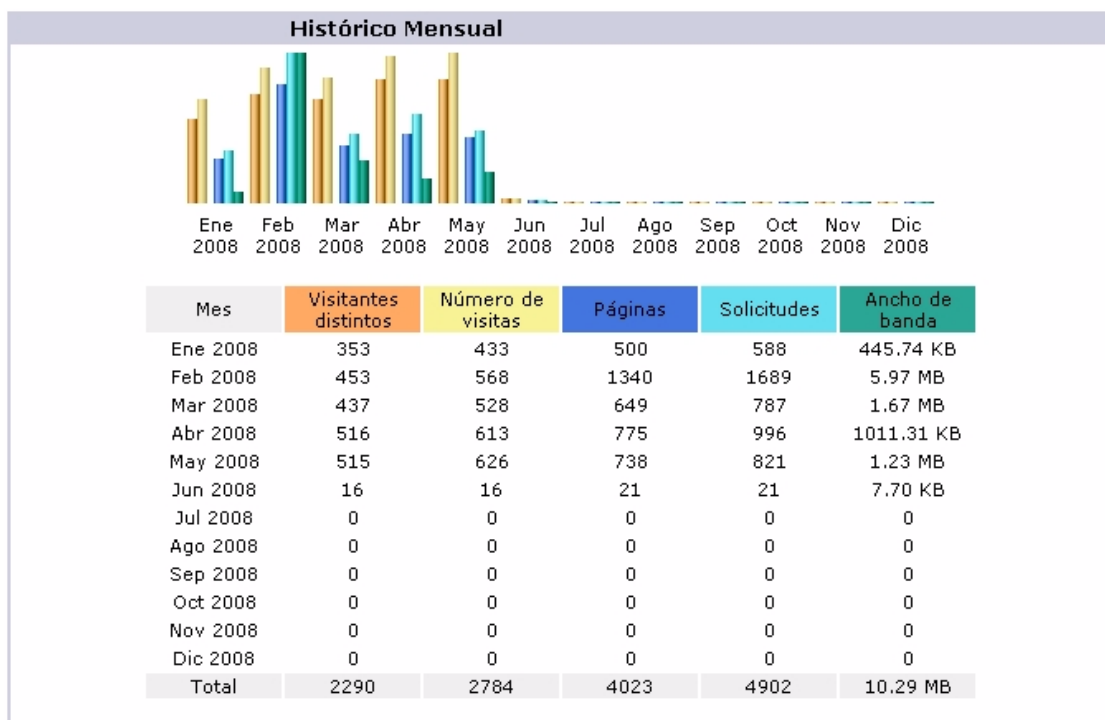


Fig. 32: Histórico mensual 2008

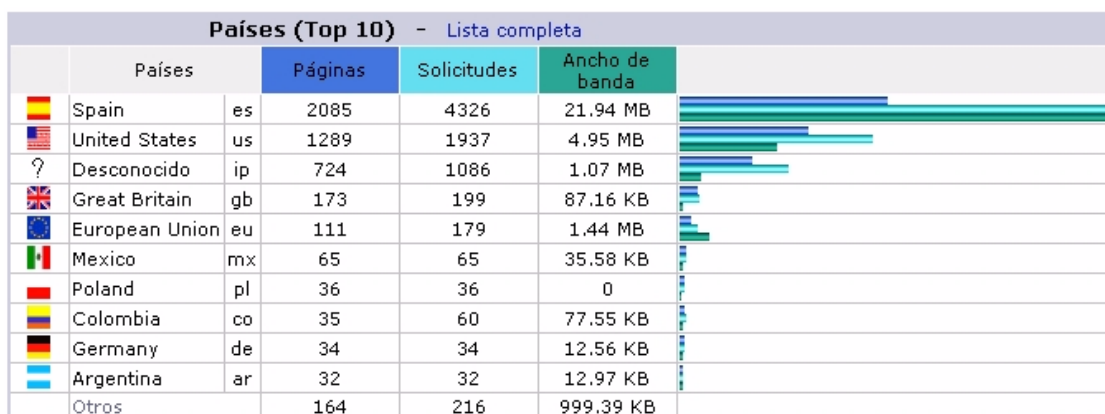


Fig. 33: Listado de principales países en acceso en 2007



Fig. 34: Listado de principales países en acceso en 2008

Servidores (Top 10) - <a href="#">Lista completa</a> - <a href="#">Última visita</a> - <a href="#">Dirección IP no identificada</a>				
Servidores : 1998	Páginas	Solicitudes	Ancho de banda	Última visita
aluche.dia.fi.upm.es	645	1185	10.65 MB	19 Dic 2007 - 10:08
66.55.151.2	113	113	0	29 Dic 2007 - 20:47
halliday.dia.fi.upm.es	61	122	418.48 KB	02 Oct 2007 - 16:37
eubank.netcraft.com	60	60	0	11 Sep 2007 - 18:08
fuencarral.dia.fi.upm.es	46	64	356.90 KB	23 Nov 2007 - 17:32
236.pool85-50-2.dynamic.orange.es	43	104	737.10 KB	11 Nov 2007 - 13:14
duero.dit.upm.es	41	77	169.45 KB	12 Sep 2007 - 13:32
jeison.deusto.es	40	401	702.61 KB	24 May 2007 - 12:46
dial.net.pl	36	36	0	07 May 2007 - 22:15
lager.netcraft.com	32	32	0	27 Dic 2007 - 12:06
Otros	3631	5976	17.62 MB	

Fig. 35: Listado de principales servidores en 2007

Servidores (Top 10) - <a href="#">Lista completa</a> - <a href="#">Última visita</a> - <a href="#">Dirección IP no identificada</a>				
Servidores : 2119	Páginas	Solicitudes	Ancho de banda	Última visita
63.80.56.40	409	574	4.25 MB	06 Feb 2008 - 08:06
207.43.224.72	242	308	1.21 MB	06 Feb 2008 - 04:26
66.55.151.2	75	75	0	01 Jun 2008 - 03:39
lager.netcraft.com	69	69	0	23 May 2008 - 00:59
26-204-34-66.cust.propagation.net	30	30	12.16 KB	30 May 2008 - 22:44
195.2.114.31	25	25	10.13 KB	01 Abr 2008 - 09:29
78.157.143.200	15	15	6.08 KB	23 Abr 2008 - 09:17
server.minimedia.co.uk	13	13	5.27 KB	09 Abr 2008 - 11:20
213.201.115.248	12	12	4.05 KB	05 May 2008 - 14:27
dv3.brianhallinc.com	12	12	4.86 KB	29 Feb 2008 - 20:02
Otros	3121	3769	4.78 MB	

Fig. 36: Listado de principales servidores en 2008

Visitas de Robots/Spiders (Top 10) - <a href="#">Lista completa</a> - <a href="#">Última visita</a>			
16 Visitas de Robots*	Solicitudes	Ancho de banda	Última visita
MSNBot	169	68.49 KB	30 Dic 2007 - 13:11
Inktomi Slurp	164	197.21 KB	30 Dic 2007 - 05:31
Googlebot (Google)	91	104.29 KB	29 Dic 2007 - 04:56
Voila	44	17.83 KB	29 Dic 2007 - 11:10
Unknown robot (identified by 'crawl')	42	17.02 KB	23 Dic 2007 - 11:05
EchO!	30	50.00 KB	19 Nov 2007 - 20:38
Jeeves	26	10.54 KB	30 Dic 2007 - 00:14
Netcraft Web Server Survey	13	0	31 Dic 2007 - 17:37
Unknown robot (identified by 'robot')	9	3.65 KB	23 Nov 2007 - 05:41
Unknown robot (identified by 'spider')	6	2.43 KB	16 Nov 2007 - 08:18
Otros	14	5.67 KB	

Fig. 37: Listado de los principales robot/spiders 2007





Visitas de Robots/Spiders (Top 10) - Lista completa - Última visita			
16 Visitas de Robots*	Solicitudes	Ancho de banda	Última visita
Inktomi Slurp	140	279.25 KB	30 May 2008 - 14:48
MSNBot	129	52.28 KB	01 Jun 2008 - 16:15
Googlebot (Google)	52	115.23 KB	01 Jun 2008 - 20:59
Jeeves	42	17.02 KB	30 May 2008 - 16:41
Voila	34	13.78 KB	01 Jun 2008 - 03:07
Unknown robot (identified by 'crawl')	31	12.56 KB	29 May 2008 - 05:05
GigaBot	9	3.65 KB	23 May 2008 - 14:40
Netcraft Web Server Survey	5	0	01 Jun 2008 - 17:58
Unknown robot (identified by 'spider')	4	830 Bytes	10 Abr 2008 - 12:38
Alexa (IA Archiver)	3	1.22 KB	22 Mar 2008 - 06:26
Otros	9	3.65 KB	

Fig. 38: Listado de los principales robot/spiders 2008

Duración de las visitas		
Número de visitas: 3164 - Media: 140 s	Número de visitas	Porcentaje
0s-30s	2804	88.6 %
30s-2mn	59	1.8 %
2mn-5mn	59	1.8 %
5mn-15mn	87	2.7 %
15mn-30mn	70	2.2 %
30mn-1h	73	2.3 %
1h+	12	0.3 %

Fig. 39: Duración de las visitas en 2007

Duración de las visitas		
Número de visitas: 2784 - Media: 81 s	Número de visitas	Porcentaje
0s-30s	2584	92.8 %
30s-2mn	41	1.4 %
2mn-5mn	34	1.2 %
5mn-15mn	60	2.1 %
15mn-30mn	26	0.9 %
30mn-1h	32	1.1 %
1h+	6	0.2 %
Desconocido	1	0 %

Fig. 40: Duración de las visitas en 2008









Tipo de fichero						
Tipo de fichero			Solicitudes	Porcentaje	Ancho de banda	Porcentaje
	html	HTML or XML static page	3720	45.5 %	1.07 MB	3.4 %
	gif	Image	2815	34.4 %	6.40 MB	20.9 %
	jsp	Dynamic Html page or Script file	948	11.6 %	18.86 MB	61.6 %
	css	Cascading Style Sheet file	196	2.3 %	1.29 MB	4.2 %
	png	Image	184	2.2 %	130.27 KB	0.4 %
	jpg	Image	168	2 %	167.26 KB	0.5 %
	Desconocido		80	0.9 %	1.18 MB	3.8 %
	js	Javascript file	59	0.7 %	1.51 MB	4.9 %

Fig. 41: Índice de ficheros accedidos en 2007









Tipo de fichero						
Tipo de fichero			Solicitudes	Porcentaje	Ancho de banda	Porcentaje
	html	HTML or XML static page	3344	68.2 %	1.26 MB	12.2 %
	gif	Image	737	15 %	2.11 MB	20.4 %
	Desconocido		525	10.7 %	1.59 MB	15.4 %
	jsp	Dynamic Html page or Script file	151	3 %	3.63 MB	35.2 %
	jpg	Image	48	0.9 %	103.27 KB	0.9 %
	css	Cascading Style Sheet file	45	0.9 %	425.52 KB	4 %
	png	Image	45	0.9 %	739.29 KB	7 %
	js	Javascript file	4	0 %	144.27 KB	1.3 %
-	db		3	0 %	337.00 KB	3.1 %

Fig. 42: Índice de ficheros accedidos en 2008

Páginas/URLs (Top 10) - Lista completa - Pagina de entrada - Salida					
261 páginas diferentes	Accesos	Tamaño medio	Página de entrada	Salida	
/	3650	300 Bytes	3065	3013	
/home.jsp	256	7.29 KB	52	60	
/SemWeb/sewView/header.jsp?_origin=%2FsewView%2Fframes.jsp	97	9.88 KB			
/SemWeb/sewView/frames.jsp	95	271 Bytes	1		
/SemWeb/home.jsp	95	7.44 KB		2	
/home.jsp?_origin=%2Fhome.jsp&content=%2Fintroduction.jsp	30	43.74 KB	1	8	
/home.jsp?content=/sew/viewTerm&_origin=%2Fhome.jsp&_sew_concept...	21	131.43 KB		2	
/home.jsp?content=/sew/viewTerm&_origin=%2Fhome.jsp&_sew_concept...	16	67.11 KB			
?mosConfig_absolute_path=http://sagasufx.com/rulez.txt?	15	415 Bytes	10	10	
/semweb/	11	105 Bytes	4		
Otros	462	25.61 KB	31	69	

Fig. 43: Lista de páginas más accedidas en 2007

Páginas/URLs (Top 10) - Lista completa - Pagina de entrada - Salida					
755 páginas diferentes	Accesos	Tamaño medio	Página de entrada	Salida	
/	2564	369 Bytes	2249	2226	
/home.jsp	69	7.94 KB	19	23	
?mosConfig_absolute_path=http://www.gumgangfarm.com/shop/data/i...	32	415 Bytes	22	24	
//?mosConfig_absolute_path=http://www.cdpm3.com/id.txt?	31	415 Bytes	9	9	
//?mosConfig_absolute_path=http://ljmirc.com.ar/id.txt?	24	415 Bytes	11	11	
//?mosConfig_absolute_path=http://forum.ivc.com.ua/language/lang...	20	415 Bytes	10	10	
?mosConfig_absolute_path=http://12.30.229.109/images/.../di??	14	415 Bytes	7	7	
?mosConfig_absolute_path=http://www.vacancespreference.com/edit...	12	415 Bytes	5	4	
?mosConfig_absolute_path=http://www.cdpm3.com/id.txt?	9	415 Bytes	6	6	
//?mosConfig_absolute_path=http://www.airmatrix.net/id.txt?	9	415 Bytes	5	5	
Otros	1239	4.39 KB	441	457	

Fig. 44: Lista de páginas más accedidas en 2008



<b>Buscadores de frases clave (Top 10)</b>		
Lista completa		
216 Frases clave diferentes	Búsquedas	Porcentaje
red semantica	304	27.3 %
web semantica	213	19.1 %
web semántica	67	6 %
semantica	45	4 %
red web semantica	44	3.9 %
red semántica	26	2.3 %
red web semántica	25	2.2 %
semantic web	23	2 %
redwebsemantica	12	1 %
red web	12	1 %
Otras cadenas de búsqueda	339	30.5 %

Fig. 45: Lista principales frases usadas en búsquedas 2007

<b>Buscadores de frases clave (Top 10)</b>		
Lista completa		
217 Frases clave diferentes	Búsquedas	Porcentaje
web semantica	274	24.8 %
red semantica	261	23.6 %
web semántica	69	6.2 %
semantica	57	5.1 %
red semántica	41	3.7 %
semantic web	34	3 %
red tematica	20	1.8 %
red web semantica	14	1.2 %
tematica	11	0.9 %
semantica web	10	0.9 %
Otras cadenas de búsqueda	313	28.3 %

Fig. 46: Lista principales frases usadas en búsquedas 2008

<b>Buscadores de palabras clave (Top 10)</b>		
Lista completa		
148 Palabras diferentes	Búsquedas	Porcentaje
semantica	732	27.5 %
red	570	21.4 %
web	551	20.7 %
semántica	166	6.2 %
la	44	1.6 %
es	43	1.6 %
que	41	1.5 %
semantic	41	1.5 %
de	39	1.4 %
tematica	36	1.3 %
Otras palabras	396	14.8 %

Fig. 47: Lista principales palabras usadas en búsquedas 2007

<b>Buscadores de palabras clave (Top 10)</b>		
<a href="#">Lista completa</a>		
134 Palabras diferentes	Búsquedas	Porcentaje
semantica	749	28.4 %
web	560	21.2 %
red	489	18.5 %
semántica	154	5.8 %
tematica	66	2.5 %
semantic	62	2.3 %
de	48	1.8 %
que	34	1.2 %
es	33	1.2 %
español	32	1.2 %
Otras palabras	407	15.4 %

**Fig. 48:** Lista principales palabras usadas en búsquedas 2008



## 10.2 Algunos usos actuales representativos de la Web Semántica

Huyendo de los tópicos de las aplicaciones o herramientas de la Web Semántica que se suelen indicar en todos los trabajos, se mencionan a continuación algunas de las aplicaciones de vanguardia (que en algunos casos puede resultar su clasificación como Web Semántica algo difusa), más punteras y con mejor recorrido:

[Freebase](#) es una base de datos de distintos tipos de contenidos, cercano a las definiciones enciclopédicas. Cualquier persona puede ingresar información nueva, es “abierta” y tiene su API. Tiene mucho de la wikipedia pero con un agregado que la diferencia, como las sugerencias que brinda a los usuarios al realizar las búsquedas, la organización es por tipo y con “etiquetado semántico”. Además es posible dejar comentarios y comenzar discusiones. Más sobre su funcionamiento en este video.



[PowerSet](#). Quizá el buscador más esperado para que salga de su beta cerrada para comprobar si lo bien que se habla del servicio es cierto o más bien una buena promoción. Existe un video y algún pantallazo de los resultados, poco más. El “buscador de lenguaje natural”, así de autodenominan, no se basa en búsquedas por palabras o términos (keywords) sino con frases naturalmente dichas. Un ejemplo visto es “Qué senadores escribieron un libro?” y los resultados muestran los nombres de esos senadores y no textos que contengan esa frase. Se supone que es (o será, veremos) la gran competencia a Google. Más información en su blog.

[Twine](#). Proclamada por ellos mismos como la primera aplicación principal de la Web Semántica. También en beta cerrada, el servicio propone ser “una inteligente ayuda para compartir, organizar y encontrar información con gente en quien confiás”. A medida que el usuario va utilizando el servicio e ingresa información Twine va “aprendiendo” de sus preferencias que se completará con una “Semántica Gráfica” y recogerá el etiquetado que el mismo usuario realice en otras redes. Para ser sincero, hasta no probarlo no me queda muy claro su funcionamiento. Pueden pedir su registro en el sitio principal.

[AdaptiveBlue](#) ofrece 2 servicios, que sintetizados a continuación. Smartkinks, un plugin para nuestros blogs que agrega un pop-up a los enlaces para dar información complementaria. Por ejemplo, si enlazamos a un cantante se agregará un pequeño cuadrado que al pulsarlo abrirá una ventana con más data sobre ese músico de una gran cantidad de sitios. BlueOrganizer es una extensión para el navegador que facilita la organización de la información en redes sociales. Por ejemplo, al pulsar sobre una imagen el menú contextual nos da la posibilidad de guardarlo en nuestro del.icio.us, o en Flickr, o en Last.fm según nuestras preferencias. Ambos tienen la particularidad de estructurar información que aun no lo está, organizándola.



[Hakia](#). Se trata de otro buscador semántico, que interpreta las búsquedas del lenguaje natural. Puede responder preguntas escritas directamente. Según dijo su creador funciona rastreando contenidos de texto en la web para luego analizarla semánticamente.



[Talis](#) es una compañía británica de software de más de 40 años que ha creado una plataforma de aplicaciones web semántica. Es diferente a los servicios descritos aquí, se trata de una plataforma y no de un sólo producto. Es una combinación de Web 2.0 y Web Semántica en el sentido que permite a los desarrolladores crear aplicaciones para compartir, remezclar y reutilizar datos. La empresa hace hincapié en “la importancia del contexto, la función, la intención y la atención de

manera significativa en el comportamiento de seguimiento de toda la web.”

[True Knowledge](#) es otro buscador que combina al análisis del lenguaje natural, una base de conocimientos internos y base de datos externos para ofrecer respuestas inmediatas a las diversas preguntas. En lugar de simplemente llevar a los usuarios a las páginas webs del motor de búsqueda, este servicio además cree que puede encontrar la respuesta explícita. En el video explicativo muestran por ejemplo los resultados ante la pregunta si Jennifer López está soltera. En la mayoría de los buscadores te llevan o a wikipedia u otro sitio popular. Pero aquí directamente te dicen NO, gran cartel explicativo ofreciendo la respuesta. Debajo sí están los resultados de sitios webs. Tiene una API y blog propio.



[Tripit](#) es una aplicación para gestionar y planificar nuestros viajes. Al registrarnos e indicarles el lugar donde iremos y recibiremos información por correo con artículos de wikipedia, fotos de flickr, artículos relacionados y eventos. También es posible sincronizar nuestra agenda con la de nuestros acompañantes para que entren en el plan de viaje.

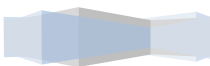


[ClearForest](#), adquirida hace algunos meses por Reuters, tiene un servicio web y una extensión de Firefox llamada Gnosis que le “permite identificar a las personas, empresas, organizaciones, productos y geografías en la página que está viendo”. Basta un click en el menú para visualizar diversos tipos de anotaciones que Gnosis aporta.

[Spock](#) es un “buscador de personas” (del cual ya hemos hablado) del cual se ha dicho que es “uno de los mejores motores de búsqueda verticales semántico construido hasta la fecha.” Su creador dice que las características que lo diferencian son: la búsqueda centralizada en la persona, un set de atributos que caracterizan a las personas (la geografía, fecha de nacimiento, ocupación, etc), el uso de etiquetas como vínculos y relaciones entre las personas, mecanismo de corrección de los usuarios a través del bucle.

**Tabla 55: Aplicaciones vanguardia en Web Semántica**





## Índice de Figuras

---

Fig. 1: Diagramas de Gantt con la programación temporal del desarrollo del TFC...	17
Fig. 2: Diagrama de análisis, proceso y formateo de un documento XML.....	21
Fig. 3: Grafo de declaración RDF.....	24
Fig. 4: Serialización en XML de sentencias RDF.....	25
Fig. 5: Ejemplo de uso de un contenedor Bag.....	26
Fig. 6: Jerarquía de clases del modelo RDFS.....	29
Fig. 7: Versiones del Lenguaje OWL.....	31
Fig. 8: Variantes de WSMML.....	33
Fig. 9: Estructura en capas de la Web Semántica.....	36
Fig. 10: Diagrama de llamada a Procedimiento.....	38
Fig. 11: Diagrama de invocación RMI.....	39
Fig. 12: Diagrama de interacción de Servicios Web.....	40
Fig. 13: Esquema de desarrollo e influencia hasta llegar a los Servicios Web Semánticos.....	43
Fig. 14: Ontología de alto nivel OWL-S.....	44
Fig. 15: Visión general de funcionamiento de WMSO.....	45
Fig. 16: Anotación semántica de los elementos de WSDL.....	47
Fig. 17: La Web Actual frente a la Web Semántica.....	62
Fig. 18: De la Web actual a la Web Semántica.....	64
Fig. 19: Ejemplo de codificación en FOAF.....	145
Fig. 20: Ejemplo de identificación de personas en FOAF.....	146
Fig. 21: RDF_VALUE\$, RDF_NODE\$, RDF_LINK\$ y RDF_MODEL\$ son las tablas necesarias para almacenar RDF en el Oracle RDF Data Model.....	168
Fig. 22: Parsing de las consultas y optimización del modelo.....	173
Fig. 23: Arquitectura de RDF Gateway.....	175
Fig. 24: Representación de la interacción de las Redes sociales.....	187
Fig. 25: Representación gráfica del metabuscador Kartoo.....	189
Fig. 26: Estadísticas generales de PWSW.....	201
Fig. 27: Ejemplo de XML-RPC Request.....	202
Fig. 28: Ejemplo de XML-RPC Response.....	202
Fig. 29: Sumario accesos año 2007.....	204
Fig. 30: Sumario accesos año 2008.....	204
Fig. 31: Histórico mensual 2007.....	204
Fig. 32: Histórico mensual 2008.....	205
Fig. 33: Listado de principales países en acceso en 2007.....	205
Fig. 34: Listado de principales países en acceso en 2008.....	205
Fig. 35: Listado de principales servidores en 2007.....	206
Fig. 36: Listado de principales servidores en 2008.....	206
Fig. 37: Listado de los principales robot/spiders 2007.....	206
Fig. 38: Listado de los principales robot/spiders 2008.....	207
Fig. 39: Duración de las visitas en 2007.....	207
Fig. 40: Duración de las visitas en 2008.....	207





<b>Fig. 41: Índice de ficheros accedidos en 2007</b> .....	207
<b>Fig. 42: Índice de ficheros accedidos en 2008</b> .....	208
<b>Fig. 43: Lista de páginas más accedidas en 2007</b> .....	208
<b>Fig. 44: Lista de páginas más accedidas en 2008</b> .....	208
<b>Fig. 45: Lista principales frases usadas en búsquedas 2007</b> .....	209
<b>Fig. 46: Lista principales frases usadas en búsquedas 2008</b> .....	209
<b>Fig. 47: Lista principales palabras usadas en búsquedas 2007</b> .....	209
<b>Fig. 48: Lista principales palabras usadas en búsquedas 2008</b> .....	210



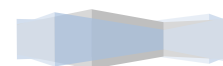
## Índice de Tablas

---

Tabla 1: Cronología de desarrollo del TFC.....	15
Tabla 2: Diferencias entre HTML y XML .....	20
Tabla 3: Características fundamentales del lenguaje XML.....	22
Tabla 4: Ejemplos de deficiencias en la búsqueda dentro de la Web Actual.....	60
Tabla 5: Diferencias básicas entre la Web Actual y la Web Semántica.....	65
Tabla 6: Herramientas de creación de ontologías .....	69
Tabla 7: Herramientas de anotación externa.....	70
Tabla 8: Herramientas de autor .....	71
Tabla 9: Lenguajes de codificación de ontologías.....	72
Tabla 10: Esquema 1, anotaciones externas y de autor .....	74
Tabla 11: Esquema 2, creación de ontologías .....	76
Tabla 12: Características de Cohse .....	79
Tabla 13: Características de Amaya .....	82
Tabla 14: Características de Anozilla .....	85
Tabla 15: Características de Yawas.....	88
Tabla 16: Características de Annotation System.....	91
Tabla 17: Características de Trellis Web.....	95
Tabla 18: Características de MnM .....	99
Tabla 19: Características de OntoMat-Annotizer.....	102
Tabla 20: Características de SHOE Knowledge Annotator.....	106
Tabla 21: Características de SMORE .....	109
Tabla 22: Características de Melita.....	113
Tabla 23: Características de GATE .....	116
Tabla 24: Características de LinkFactory .....	119
Tabla 25: Características de OIEd.....	122
Tabla 26: Características de OntoEdit .....	125
Tabla 27: Características de OntolinguaServer .....	129
Tabla 28: Características de Protégé .....	133
Tabla 29: Características de WebODE .....	137
Tabla 30: Comparativa de herramientas de anotación externas (I).....	138
Tabla 31: Comparativa de herramientas de anotación externas (II).....	139
Tabla 32: Comparativa de herramientas de anotación autor (I).....	140
Tabla 33: Comparativa de herramientas de anotación autor (II).....	141
Tabla 34: Comparativa de herramientas de creación de ontologías (I) .....	142
Tabla 35: Comparativa de herramientas de creación de ontologías (II).....	143
Tabla 36: Comparativa de herramientas de creación de ontologías (III) .....	144
Tabla 37: Características de container .....	154
Tabla 38: Características de Entry.....	155
Tabla 39: Características Rich Site Summary (RSS 0.91).....	155
Tabla 40: Características RDF Site Summary (RSS 1.0) .....	156
Tabla 41: Características Really Simple Syndication (RSS 2.0) .....	156
Tabla 42: Características generales Atom 1.0 .....	158



<b>Tabla 43: SGBD de tipo Open Source</b> .....	160
<b>Tabla 44: SGBD de tipo Comercial</b> .....	162
<b>Tabla 45: Algunos SGBD que sí soportan XML</b> .....	162
<b>Tabla 46: Análisis de Oracle 10g</b> .....	169
<b>Tabla 47: Análisis de Jena 2</b> .....	171
<b>Tabla 48: Análisis de Sesame</b> .....	173
<b>Tabla 49: Análisis de DRF Gateway</b> .....	176
<b>Tabla 50: Listado General de aplicaciones RDF según la base SWAD-Europe</b> .....	194
<b>Tabla 51: Numero de 'entities types' conocidos en el servicio PTSW</b> .....	199
<b>Tabla 52: Numero de 'namespaces' conocidos en el servicio PTSW</b> .....	201
<b>Tabla 53: Estadísticas generales del servicio Swoogle para la Web Semántica</b> .....	203
<b>Tabla 54: Estadísticas generales del servicio SemanticWeb.org</b> .....	203
<b>Tabla 55: Aplicaciones vanguardia en Web Semántica</b> .....	212



## Glosario

---

### A

#### **Aidministrador**

Empresa holandesa dedicada a la ingeniería del conocimiento. Esta empresa desarrollo Sesame (véase Sesame).

#### **Anotación**

El proceso de asociar a un recurso metadatos que lo describen en todo o en parte.

#### **Anotación semántica**

Anotación en la que los metadatos están definidos formalmente y son procesables por ordenador.

#### **Apache**

Servidor de páginas Web desarrollado por la Apache Software Foundation, organización formada por miles de voluntarios que colaboran para la creación de software de libre distribución.

#### **API**

Interfaz de programación de aplicaciones (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones, interfaz de programación de la aplicación): una serie de funciones que están disponibles para realizar programas para un cierto entorno.

Representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software.

#### **ASP**

Active Server Pages. Una especificación que permite crear dinámicamente páginas Web mediante HTML, scripts, y componentes de servidor ActiveX reutilizables.

### B

#### **Benchmark**

Programa especialmente diseñado para evaluar el rendimiento de un sistema, de software o de hardware.

#### **Bugs**

Un error de software o computer bug, que significa bicho de computadora, es el resultado de una falla de programación introducida en el proceso de creación de programas de ordenador. El término bug fue acreditado erróneamente a Grace Murray Hopper, una pionera en la historia de la computación, pero Thomas Edison ya lo empleaba en sus trabajos para describir defectos en sistemas mecánicos por el año 1870.

### C

#### **CDA**

Estándar para modelizar información en el dominio de la medicina.



## **CGI**

Common Gateway Interface (en inglés "Pasarela de Interfaz Común", abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (explorador Web) solicitar datos de un programa ejecutado en un servidor Web. CGI especifica un estándar para transferir datos entre el cliente y el programa.

## ***D***

### **DAML**

DARPA Agent Markup Language. Es un lenguaje para modelar ontologías creado con una extensión de RDF. Está siendo desarrollado por el DARPA en colaboración con W3C.

### **DAML+OIL**

DARPA Agent Markup Language + Ontology Inference Layer. Es un lenguaje de codificación de ontologías.

### **DOM**

Document Object Model, literalmente "Modelo de Objetos de Documento". Interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts acceder y actualizar dinámicamente los contenidos, la estructura y el estilo de los documentos HTML y XML

### **DTD**

Document Type Definition. Un documento que describe la estructura de una página Web escrita en XML.

## ***E***

### **ebXML**

Estándar para modelizar información en el dominio de las finanzas.

## ***F***

### **FLASH**

Software de Macromedia para crear pequeñas animaciones vectoriales reproducidas en la Web. El navegador de un usuario necesita el plug-in Flash Player para interpretar las animaciones Flash.

### **FpXML**

Estándar para modelizar información en el dominio de las finanzas.

### **Framework**

Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, librerías y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.



## **G**

### **GPL**

La GNU General Public License (Licencia Pública General) es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre.

### **Grafo**

Estructura de datos utilizada en algunos lenguajes de programación, en la cual cada elemento puede tener uno o varios predecesores y uno o varios sucesores.

## **H**

### **Hipermedia**

Enfoque para manejar y organizar información, a partir de una red de nodos conectados por enlaces. Estos nodos pueden contener textos, gráficos, imágenes, audio, animaciones y video. El usuario tiene la flexibilidad de acceder a dichos nodos de manera no secuencial lo que genera la noción de navegación.

### **HTML**

Acronimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

### **HTTP**

Es el protocolo de la Web (WWW), usado en cada transacción. Las letras significan Hyper Text Transfer Protocol, es decir, protocolo de transferencia de hipertexto. El hipertexto es el contenido de las páginas Web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla

## **I**

### **IEEE**

IEEE corresponde a las siglas del Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos, una asociación estadounidense dedicada a la estandarización. Es una asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros de telecomunicaciones, ingenieros electrónicos, Ingenieros en informática, etc.

### **IEEE LOM**

Estándar para modelizar información en el dominio de la enseñanza.

### **IsaViz**

Entorno para la visualización y edición de modelos RDF.



## **J**

### **J2EE**

J2EE son las siglas de Java 2 Enterprise Edition que es la edición empresarial del paquete Java creada y distribuida por Sun Microsystems. Comprenden un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales.

Debido a que J2EE no deja de ser un estándar, existen otros productos desarrollados a partir de ella aunque no exclusivamente.

### **Java**

Java es una plataforma de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales.

### **JavaBeans**

JavaBeans es un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java.

### **JavaScript**

Lenguaje desarrollado por Netscape. Aunque es parecido a Java se diferencia de él en que los programas están incorporados en el archivo HTML.

### **Jena**

Framework desarrollado por HP Labs para manipular metadatos desde una aplicación Java.

### **Joseki**

Servidor para la publicación, en la Web, de modelos RDF.

### **JSP**

Java Server Pages (JSP) es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java

## **K**

### **Kaon**

Entorno gráfico para visualizar y construir ontologías.

## **L**

## **M**

### **Metadato**

Los metadatos son datos sobre los datos, esto es, información sobre la información misma. En esencia, intentan responder a las preguntas quién, qué, cuándo, cómo, dónde y porqué, sobre cada una de las facetas relativas a los datos que se documentan.

## **MySQL**

MySQL es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto.

## **N**

### **N3**

Lenguaje de programación equivalente a RDF.

### **Netscape**

Navegador Web, el cual estaba basado, en un principio, en el programa Mosaic. Ha crecido en sus características rápidamente y ahora se reconoce mundialmente como el mejor y más popular navegador. La corporación Netscape también produce otro tipo de software.

### **NewsML**

Estándar para modelizar información en el dominio del periodismo.

### **NLM Medline**

Estándar para modelizar información en el dominio de la medicina.

## **O**

### **OCML**

Lenguaje de programación.

### **Ontologia**

Hace referencia al intento de formular un exhaustivo y riguroso esquema conceptual dentro de un dominio dado, con la finalidad de facilitar la comunicación y la compartición de la información entre diferentes sistemas.

### **Oracle**

Sistema de administración de base de datos (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

### **OWL**

Acronym de "Web Ontology Language", un lenguaje de marcado para la publicación y compartición de datos usando ontologías en la Web.

## **P**

### **Parser**

Software especializado en reconocer marcas en un documento.

### **PHP**

PHP (acrónimo recursivo de "PHP: Hypertext Preprocessor", originado inicialmente del nombre PHP Tools, o Personal Home Page Tools) es un lenguaje de programación interpretado. Aunque fue concebido en el tercer trimestre de 1994 por Rasmus Lerdorf. Se utiliza entre otras cosas para la programación de páginas Web activas, y se destaca por su capacidad de mezclarse con el código HTML





## **Plugin**

Un plugin (o plug-in) es un programa de ordenador que interactúa con otro programa para aportarle una función o utilidad específica, generalmente muy específica. Los plugins típicos tienen la función de reproducir determinados formatos de gráficos, reproducir datos multimedia, codificar/decodificar emails, filtrar imágenes de programas gráficos, etc.

## **PostgreSQL**

PostgreSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

## **PRISM**

Estándar para modelizar información en el dominio del periodismo.

## **R**

### **RDF**

RDF son las siglas de Resource Description Framework, la especificación de un modelo de metadatos, (normalmente implementado como una aplicación de XML) que ha sido desarrollada por el World Wide Web Consortium (W3C).

### **RDF Schema**

Lenguaje de definición de ontologías para la Web Semántica estándar del W3C. Tiene menos poder expresivo que OWL.

Recurso: cualquier cosa que pueda ser interesante describir en una aplicación de Web Semántica. Una página web, un correo electrónico, un fichero, ... son recursos, pero también lo son una persona, un coche, incluso una idea. Los recursos se identifican por medio de URIs.

### **RDFS**

Lenguaje para describir vocabularios RDF.

### **RDQL**

Lenguaje de programación para manejar datos RDF.

### **Recurso**

Cualquier cosa que pueda ser interesante describir en una aplicación de Web Semántica. Una página web, un correo electrónico, un fichero,... son recursos, pero también lo son una persona, un coche, incluso una idea. Los recursos se identifican por medio de URIs.

### **Reificación**

Sentencias realizadas sobre otras sentencias.

### **RIXML**

Estándar para modelizar información en el dominio del periodismo.

### **Rollback**

Comando para descartar los cambios realizados en una base de datos y que aun no han sido confirmados.

## **RQL**

Lenguaje de programación para manejar datos RDF.

## **S**

## **SAIL**

Acónimo de “Storage And Inference Layer”, es el nivel que se encarga de interactuar con los manejadores de bases de datos (DBMS) permitiendo de este modo mantener la arquitectura de Sesame independiente del DBMS. Es una API que ofrece métodos RDF específicos a sus “clientes” y traduce estos métodos a llamadas a un DBMS específico.

## **SCIPHON**

Estándar para modelizar información en el dominio de la medicina.

## **SCORM**

Estándar para modelizar información en el dominio de la enseñanza.

## **SeRQL**

Lenguaje de programación para manejar datos RDF.

## **Servlet**

Objetos que corren dentro del contexto de un servidor de aplicaciones (por ejemplo Tomcat) y extienden su funcionalidad.

## **Sesame**

Sesame fue desarrollado por la empresa holandesa Administrator. Tiene una arquitectura desarrollada para un eficiente almacenamiento y consulta de grandes cantidades de metadatos en RDF y RDF Schema.

## **SGBD**

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

## **Shell**

Parte fundamental de un sistema operativo encargada de ejecutar las órdenes básicas para el manejo del sistema. Suelen incorporar características tales como control de procesos, redirección de entrada/salida y un lenguaje de órdenes para escribir programas por lotes o (scripts).

## **SPARQL**

Lenguaje de programación para manejar datos RDF.

## **SQL**

El Lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla.



## **T**

### **Tomcat**

Tomcat ( Jakarta Tomcat ) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation.

Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Se le considera un servidor de aplicaciones.

### **TopicMaps**

Lenguaje de programación.

### **Triple RDF**

Es el elemento básico de un modelo RDF. También llamada sentencia RDF. Está compuesta por un sujeto, un predicado y un objeto. Una tripla RDF indica que el recurso identificado por el sujeto tiene una propiedad indicada por el predicado, cuyo valor es el objeto.

## **U**

### **URI**

Acrónimo de "Uniform Resource Identifier", literalmente "Identificador Uniforme de Recurso". Conjunto de caracteres que identifica un recurso de red, mediante un nombre o utilizando sus datos de localización

### **URL**

Acrónimo de "Universal Resource Locator" (Localizador Universal de Recursos / Identificador Universal de Recursos). Sistema unificado de identificación de recursos en la red. Es el modo estándar de proporcionar la dirección de cualquier recurso en Internet.

## **V**

### **Versa**

Lenguaje de programación para manejar datos RDF.

## **W**

### **W3C**

El World Wide Web Consortium, abreviadamente W3C, es una organización que produce estándares para la World Wide Web (o Telaraña Mundial). Está dirigida por Tim Berners-Lee, el creador original de URL (Uniform Resource Locator, Localizador Uniforme de Recursos), HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de Hipertexto) y HTML (Lenguaje de Marcado de Hipertexto) sobre el que se basa la Web, y por extensión su inventor.

### **WebODE**

Lenguaje de programación.

### **WWW**

La World Wide Web (del inglés, Telaraña Mundial), la Web o WWW, es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador Web para

extraer elementos de información (llamados "documentos" o "páginas Web") de los servidores Web (o "sitios") y mostrarlos en la pantalla del usuario.

## **X**

### **XBRL**

Estándar para modelizar información en el dominio de las finanzas.

### **XHTML**

XHTML, acrónimo inglés de eXtensible Hyper Text Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.

### **XML**

XML es el acrónimo del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C).

### **XMLNews**

Estándar para modelizar información en el dominio del periodismo.



## Bibliografía

---

- Antoniou G., van Harmelen F. (2004). A semantic web primer. The MIT Press.
- Baclawski, K.; Kokar, M.; Kogut, P.; Hart, L.; Smith, J.; Holmes, W.; Letkowski, J.; Aronson, M. and Emery, P.; 2002. "Extending the UML for Ontology Development", SOSYM 2002, Software System Model (2002) 1: 1-15, Springer-Verlag.
- Bechhofer S., van Harmelen F., Hendler J., Horrocks I., McGuinness D., Patel-Schneider P. and Stein L., 2004. "OWL Web Ontology Language Reference", W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>.
- Benjamins, V. R. and Gomez-Perez A., 1996. "Knowledge-System Technology: Ontologies and Problem-Solving Methods". Disponible en <http://www.swi.psy.uva.nl/usr/richard/pdf/kais.pdf>.
- Berners Lee T, Handler J, Lassila O. The Semantic Web. Scientific American 2001;284(5).
- Berners Lee T, Miller E. The Semantic Web lifts off, W3C. ERCIM News 2002;(51). Disponible: <http://www.w3.org/2002/03/key-free-trust>.
- Berners-Lee, T.; Hendler, J.; Lassila, O., 2001, "The Semantic Web". Scientific American, <http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>.
- Berney B. Software Agents. A Review. Manchester: Manchester Metropolitan University, 1996.
- Blazquez, M.; Fernandez, M.; Garcia-Pinar, J.; Gomez-Perez, A. 1998. "Building Ontologies at the Knowledge Level using the Ontology Design Environment". Knowledge Acquisition Workshop (KAW'98).
- Branchman RJ. Structural Paradigm for Representing Knowledge. Harvard: Harvard University, 1977.
- Bray T.; Paoli J. and Sperberg-Mcqueen C., 1998. "Extensible Markup Language (XML) 1.0.". W3C (World-Wide Web Consortium) Recommendation 10 February 1998. <http://www.w3.org/TR/REC-xml>.
- Broekstra J. and Kampman A.; 2001. "Sesame: A generic Architecture for Storing and Querying RDF and RDF Schema". Deliverable 9, On-To-Knowledge project. <http://www.ontoknowledge.org/down/del10.pdf>.
- Burnage G, Dunlop D. Encoding the British National Corpus. Disponible en: <http://www.natcorp.ox.ac.uk/using/papers/Burnage93a.htm>.
- Burnard L, Light R. Three SGML metadata formats: TEI, EAD, and CIMI. A Study for BIBLINK Work Package 1.1. 1996. Disponible en: <http://hosted.ukoln.ac.uk/biblink/wp1/sgml/>.
- Carlson, D.; 2001. "Modeling XML Applications with UML", Addison-Wesley.
- Castells P. Aplicación de técnicas de la web semántica. Disponible en: <http://giig.ugr.es/~mgea/coline02/Articulos/pcastells.pdf>.
- Chang, W.; 1998. "A Discussion of the Relationship Between RDF-Schema and UML", <http://www.w3.org/TR/1998/NOTE-rdf-uml-19980804/>.
- Consortium, Feb. 2004, <http://www.w3.org/TR/2004/REC-rdf-nt-20040210/>.
- Corcho O, Fernández López M, Gómez Pérez A. Methodologies, tools and languages for building ontologies: where is their meeting point? Data & Knowledge Engineering 2003;46(1):41-64.
- Corcho, O.; Fernández-López, M. and Gomez-Perez A. 2001. "Technical Roadmap v 1.0". Deliverable 1.1.1 del Consorsio OntoWeb, disponible en: [http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/deliverable\\_view](http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/deliverable_view)
- Cranefield, S. 2001. "Networked knowledge representation and exchange using UML and RDF". Journal of Digital Information, 1(8),. <http://jodi.ecs.soton.ac.uk/>.

- D. Fensel, J. Hendler, H. Lieberman, W. Wahlster. Spinning the Semantic Web. MIT Press, 2002.
- Daconta MC, Obrst Leo J, Smith KT. The Semantic Web: A guide to the future of XML, Web Services and Knowledge Management. New York:: Wiley, 2003 p. 145.
- Daconta MC, Obrst LJ, Smith KT. The Semantic Web: A guide to the future of XML. Web Services and Knowledge Management New York: Wiley, 2003.
- daml+oil directory site <http://www.daml.org/2001/03/daml+oil-index.html>
- Davies J., Fensel D., van Harmelen F. (2003), Towards the Semantic Web. John
- Díaz O, Iturroz J, Ibáñez F. Integración, navegación y presentación: experiencias utilizando XML. Novática 2000;146(4):12-9.
- Dietel Fensel. Ontologies: a Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, 2001.
- Ding Y, Foo S. Ontology research and development. Part 1. A review of ontology generation. Journal of Information Science 2002;28(2):123-36.
- Ding Y, Foo S. Ontology research and development. Part 2. A review of ontology mapping and evolving. Journal of Information Science 2002;28(5):375-88.
- E.R. Harold, W.S. Means. XML in a Nutshell. Ed. O'Reilly, 2001.
- Fensel D, Bussler C. Semantic Enabled Web Services. 2002. Disponible en:  
<http://www.swsi.org/resources/wsmf.pdf>.
- Fernández M, Simeón J, Walder P. XML Query Language: Experiences and Examples. Disponible en:  
<http://www.w3.org/1999/09/ql/docs/xquery.html>.
- García Jiménez A. Instrumentos de representación del conocimiento: tesauros versus ontologías. Anales de la Documentación 2004;(7):7.
- García Jiménez A. Instrumentos de representación del conocimiento: tesauros versus ontologías. Anales de la Documentación 2004;(7):7.
- García Sánchez, F. Sistema basado en tecnologías del conocimiento para entornos de servicios web semánticos, julio 2007, Universidad de Murcia
- Gómez Pérez A. (coord.). A Survey on Ontology Tools, OntoWeb deliverable 1.3.2002. Disponible en:  
[http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb\\_Del\\_1-3.pdf](http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-3.pdf).
- Gómez-Pérez, M. Fernández-López, O. Corcho. Ontological Engineering. Springer Verlag, 2004.
- Gruber T. and Gerbaux F., 1994. "Frame ontology". <ftp://ftp.ksl.stanford.edu/pub/knowledge-sharing/ontologies/html/frame-ontology/index.html>
- Gruber T. R. and Olsen G. R. 1994. "An Ontology for Engineering Mathematics", Knowledge Systems Laboratory, Stanford University disponible en: <http://www.ksl.stanford.edu/knowledge-sharing/papers/engmath.html>.
- Guarino N. Formal ontology and information systems. En: Guarino N (ed.). Formal Ontology in Information Systems . Amsterdam: IOS Press, 1998.
- Guerrero Bote V, Lozano Tello A. Vínculos entre las Ontologías y la Biblioteconomía y Documentación. En: La Representación y la Organización del Conocimiento en sus distintas perspectivas: su influencia en la Recuperación de la Información. Actas del IV Congreso ISKO-España EOCONSID'99, abril 22-24, Granada, España. Granada: ISKO-Facultad de Biblioteconomía y Documentación, 1999. p. 25-31.
- Hans Bergsten. JavaServer Pages. O'Reilly & Associates, Inc., 2001.
- Hayes P., McBride B.: RDF Semantics. W3C Recommendation, World-Wide Web
- Hendler J, Golbeck J, Parsia B. Trust Networks on the Semantic Web. 2003. Disponible en:  
<http://www.mindswap.org/papers/CIA03.pdf>.



- Hendler J, Golbeck J, Parsia B. Trust Networks on the Semantic Web. En: Proceedings of Cooperative Intelligent Agents, August 27-29, Helsinki, Finland. Helsinki: s.e, 2003.
- Irons Walch V. Standard for Archival Descriptions: A Handbook. Disponible en: <http://www.archivists.org/catalog/stds99/chapter3.html>.
- ISO. ISO 8879. Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML). Geneva: ISO, 1986.
- J. Davies, D. Fensel, F. Van Harmelen. Towards the Semantic Web. John Wiley & Sons, Ltd., 2003.
- Jennings N, Wooldridge M. Agent Technology - Foundations, Applications, and Markets, New York: Springer-UNICOM, 1998.
- Klyne G.; Reynolds F.; Woodrow C.; Ohto H.; Hjelm J.; Butler M. and Tran L., 2004. "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0". W3C Recommendation 15 January 2004, <http://www.w3.org/TR/2004/REC-CCPPstruct-vocab-20040115/>.
- Kogut, P.; Cranefield, S.; Hart, L.; Dutra, M.; Baclawski, K.; Kokar, M. and Smith, J.; 2003. "UML for Ontology Development". <http://www.sandsoft.com/docs/ker4.pdf>.
- Lassila O. and Swick R. R. (editores). "Resource description framework (RDF): Model and syntax specification", 1999. W3C Recommendation 1999-02-22, <http://www.w3.org/TR/REC-rdf-syntax>.
- Le Grand B and Soto M. Visualisation of the Semantic Web: Topic Maps Visualisation. En: IEEE Computer Society. Proceedings of Six International Conference on Information Visualisation, July 10-12, London, England. London: IEEE Computer Society, 2002.
- M. Birbeck et. al. Professional XML. Ed. Wrox Press, 2000.
- M. Kay. XSLT, 2nd edition. Ed. Wrox Press, 2001.
- Maller P, Dahaes V. Web Semántica: el salto evolutivo de la Web. Disponible en: <http://www.proyectoweb.org/boletines/034-marz03.html>.
- Markup Language (XML) 1.0 (Third Edition), World-Wide Web Consortium, Feb. 2004. <http://www.w3.org/TR/2004/REC-xml-20040204/>
- Matthews B, Wilson M, Brickley D. Semantic Web Advanced Development in Europe. ERCIM News 2002; ( 51):15-6.
- Merriam- Webster Online. Disponible en: [www.m-w.com](http://www.m-w.com).
- Miller E. An Introduction to the Resource Description Framework. D-Lib Magazine 1998;(5). Disponible en: <http://www.dlib.org/dlib/may98/miller/05miller.html>.
- Minsky ML. Semantic Information Processing Cambridge: MIT Press, 1968.
- Navarro, C. White, L. Burman. Mastering XML. Ed. BYBEX, 2000.
- Neches R. Enabling technology for knowledge sharing. AI Magazine 1991;12(3):36-56.
- Newcomer, E.; 2002, "Understanding Web services: XML, WSDL, SOAP, and UDDI". Addison-Wesley, Boston, MA.
- Novak JD, Gowin DB. Learning How to Learn . New York: Cambridge University Press, 1984.
- Noy NF, McGuinness DL. Ontology Development 101: A Guide to Creating Your First Ontology. 2000. Disponible en: <http://www.cs.man.ac.uk/~carole/old/GGF%20Tutorial%20Stuff/ontology101.pdf>.
- Oil directory site. <http://www.ontoknowledge.org/oil/>.
- Paling S. Converting a controlled vocabulary into an ontology. Information Research 2001;6(2). Disponible en: <http://informationr.net/ir/6-2/paper94.html>.
- Peis E, Ruíz Rodríguez AA. EAD (Encoded Archival Description): Desarrollo, estructura, uso y aplicaciones. Disponible en: <http://www.hipertext.net/web/pag223.htm>.

- Powers, S. (2003). Practical RDF. O'Reilly
- R. Baeza-Yates, B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley Pub Co., 1999.
- RDF Data Access group: <http://www.w3.org/2001/sw/DataAccess/>
- Reagle JM. Key Free Trust in the Semantic Web: Finding Bacon's Key. Disponible en:  
<http://www.w3.org/2002/03/key-free-trust>.
- Roberts N. Historical studies in documentation: pre-history of information retrieval thesaurus. Journal of Documentation 1984;40(4):271-85.
- Rusty Harold E, Means W. XML. Madrid: Anaya Multimedia, 2005.
- Senso JA. Herramientas para trabajar con RDF. El profesional de la Información 2003;12(2):132-8.
- Stein, L. A.; Connolly, D.; McGuinness, D. (editors). "DAML Ontology language specification", released in October 2000. <http://www.daml.org/2000/10/daml-ont.html>.
- Studer R, Benjamins R, Fensel D. Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering 1998;25(1/2):161-97.
- The Dublin Core Metadata Initiative. Available from <http://www.dublincore.org>
- University of Chicago. The Chicago Manual of Style. 15 th ed. Chicago: University of Chicago Press, 2005.
- V. Geroimenko and C. Chen. Visualizing the Semantic Web. Springer Verlag, 2003.
- Van Herwijnen E. Practical SGML. 2 ed. Boston: Kluwer Academic Publishers, 1994.
- W3C, WWW Consortium, 2001, Semantic Web, <http://www.w3.org/2001/sw/>
- Web Ontology Working Group of W3C, WWW Consortium, 2001,  
<http://www.w3.org/2001/sw/WebOnt>
- WordNet site, <http://www.cogsci.princeton.edu/~wn/>
- World-Wide Web Consortium, RDF: <http://www.w3.org/RDF>
- World-Wide Web Consortium, Semantic Web: <http://www.w3.org/2001/sw/>





## Páginas Web de referencia

---

<http://alt1040.com/2008/02/la-web-semantica-explicada-en-6-minutos/>  
<http://annotation.semanticweb.org>  
<http://annotation.semanticweb.org/tools/ontomat>  
<http://Annozilla.mozdev.org>  
<http://apollo.open.ac.uk/>  
<http://arantxa.ii.uam.es/~castells/docencia/semanticweb/>  
<http://arantxa.ii.uam.es/~castells/docencia/semanticweb/links.html>  
<http://blogs.zdnet.com/semantic-web/?p=115>  
[http://bvs.sld.cu/revistas/aci/vol13\\_5\\_05/aci01505.htm](http://bvs.sld.cu/revistas/aci/vol13_5_05/aci01505.htm)  
[http://bvs.sld.cu/revistas/aci/vol13\\_6\\_05/aci030605.htm](http://bvs.sld.cu/revistas/aci/vol13_6_05/aci030605.htm)  
[http://bvs.sld.cu/revistas/aci/vol13\\_6\\_05/acisu0605.htm](http://bvs.sld.cu/revistas/aci/vol13_6_05/acisu0605.htm)  
<http://cohse.semanticweb.org>  
<http://delicias.dia.fi.upm.es/webODE/>  
<http://dis.um.es/~jfernand/jfernand.htm>  
[http://en.wikipedia.org/wiki/Semantic\\_Web](http://en.wikipedia.org/wiki/Semantic_Web)  
[http://en.wikipedia.org/wiki/Semantic\\_Web\\_Services](http://en.wikipedia.org/wiki/Semantic_Web_Services)  
[http://en.wikipedia.org/wiki/Social\\_Semantic\\_Web](http://en.wikipedia.org/wiki/Social_Semantic_Web)  
<http://enealcubo.com/blog/c/?p=62>  
[http://es.geocities.com/metadatosxmlorganizacion/documentos\\_rdf.html](http://es.geocities.com/metadatosxmlorganizacion/documentos_rdf.html)  
<http://es.wikipedia.org/wiki/FOAF>  
[http://es.wikipedia.org/wiki/Web\\_sem%C3%A1ntica](http://es.wikipedia.org/wiki/Web_sem%C3%A1ntica)  
<http://esw.w3.org/topic/CommercialProducts>  
<http://esw.w3.org/topic/ConverterToRdf>  
<http://esw.w3.org/topic/SemanticWebTools>  
<http://esw.w3.org/topic/SemWebSpain>  
<http://gate.ac.uk>  
<http://giig.ugr.es/~mgea/coline02/Articulos/pcastells.pdf>  
<http://infomesh.net/2001/swintro/#itWorks>  
<http://kmi.open.ac.uk/projects/akt/MnM/documents.html>  
<http://knowledgeweb.semanticweb.org/semanticportal/home.jsp>  
<http://maude.sip.ucm.es/alberto-verdejo/tesis/index.html>  
<http://misapuntessobrewebsemntica.blogspot.com/2008/01/lassila-programming-semantic-web.html>  
<http://nets.ii.uam.es/index.html>  
<http://nlp.shef.ac.uk/melita/>  
<http://oiled.man.ac.uk>  
<http://ontologias-metadatos.50webs.com/>  
<http://ontoweb.aifb.uni-karlsruhe.de/>  
<http://pingthesemanticweb.com/api.php>  
<http://pingthesemanticweb.com/stats/namespaces.php>  
<http://protege.stanford.edu>  
<http://rdfweb.org/foaf/>  
<http://rhizomik.net/content/roberto/>  
<http://roda.ibit.org/herramientas.cfm>  
<http://sicrono.com/internet/10-aplicaciones-semanticas-el-futuro-de-internet/>

<http://swoogle.umbc.edu/2005>  
<http://texting.wordpress.com/2007/10/14/la-web-semantica-y-los-buzz-trends/>  
<http://trellis.semanticweb.org>  
<http://www.albertofortes.com/blog/?p=129>  
<http://www.celiagradin.net/kilkor/view.php?id=248>  
<http://www.cristofolrovira.com/>  
<http://www.cs.umd.edu/projects/plus/SHOE/>  
<http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html>  
<http://www.cs.us.es/~joaquin/blog/index.php/category/web-semantica/page/2/>  
<http://www.daml.org/tools>  
<http://www.daml.ri.cmu.edu/Cal/>  
[http://www.di.uniovi.es/~dani/?doctorado\\_y\\_m%E1ster:Curso\\_2007-08](http://www.di.uniovi.es/~dani/?doctorado_y_m%E1ster:Curso_2007-08)  
<http://www.dublincore.org>  
<http://www.elprofesionaldelainformacion.com/contenidos/2003/septiembre/2.pdf>  
<http://www.hipertexto.info/documentos/ontologias.htm>  
[http://www.hipertexto.info/documentos/web\\_semantica.htm](http://www.hipertexto.info/documentos/web_semantica.htm)  
[http://www.hipertexto.info/documentos/websemant\\_rec.htm](http://www.hipertexto.info/documentos/websemant_rec.htm)  
<http://www.ietf.org>  
<http://www.imresources.fit.qut.edu.au/vocab/>  
<http://www.infovis.net/printMag.php?num=26&lang=1>  
<http://www.ksl.stanford.edu/software/ontolingua/>  
<http://www.landcglobal.com>  
[http://www.lawebsemantica.com/index.html?\\*session\\*id\\*key\\*session\\*id\\*val\\*](http://www.lawebsemantica.com/index.html?*session*id*key*session*id*val*)  
[http://www.masternewmedia.org/es/web\\_social/aplicaciones\\_sociales/web\\_20\\_web\\_semantica\\_y\\_aPLICACIONES\\_socIALES\\_interconectadas\\_20060214.htm](http://www.masternewmedia.org/es/web_social/aplicaciones_sociales/web_20_web_semantica_y_aPLICACIONES_socIALES_interconectadas_20060214.htm)  
<http://www.mindswap.org/~aditkal/editor.shtml>  
[http://www.mkbergman.com/?page\\_id=325](http://www.mkbergman.com/?page_id=325)  
<http://www.ncb.ernet.in/groups/dake/annotate/details.shtml>  
<http://www.ontoknowledge.org>  
<http://www.ontoprise.com>  
<http://www.pecesama.net/weblog/2004/12/04/friend-of-a-friend-o-cmo-construir-redes-sociales-en-la-web-semantica/>  
<http://www.redwebsemantica.es/SemWeb/sewView/frames.jsp>  
<http://www.schemaweb.info/>  
<http://www.semanticweb.net/>  
<http://www.semanticplanet.com/>  
<http://www.semanticweb.com/>  
<http://www.semanticweb.org>  
<http://www.slideshare.net/Tana/web-semantica-tana-barcamp/>  
<http://www.swi.psy.uva.nl/projects/ibrow/home.html>  
<http://www.swi.psy.uva.nl/wondertools/index.html>  
<http://www.univ-savoie.fr/labos/syscom/Laurent.Denoue/>  
<http://www.w3.org/>  
<http://www.w3.org/1999/02/26-modules/Distribution.html>  
<http://www.w3.org/1999/02/26-modules/User/Annotations-HOWTO>  
<http://www.w3.org/1999/02/26-modules/User/RdfSQL-HOWTO.html>  
<http://www.w3.org/2001/03/identification-problem/>  
<http://www.w3.org/2001/Annotea/>



<http://www.w3.org/2001/sw/>  
<http://www.w3.org/2001/sw/Europe/>  
[http://www.w3.org/2001/sw/Europe/reports/open\\_demonstrators/hp-applications-survey](http://www.w3.org/2001/sw/Europe/reports/open_demonstrators/hp-applications-survey)  
<http://www.w3.org/2001/sw/WebOnt/>  
<http://www.w3.org/Amaya/>  
<http://www.w3.org/Signature/>  
<http://www.w3c.es/divulgacion/guiasbreves/WebSemantica>  
[http://www.wikilearning.com/articulo/la\\_web\\_semantica-implementacion/20829-3](http://www.wikilearning.com/articulo/la_web_semantica-implementacion/20829-3)  
[http://www.wikilearning.com/curso\\_gratis/biblioteca\\_digital\\_y\\_web\\_semantica/7891](http://www.wikilearning.com/curso_gratis/biblioteca_digital_y_web_semantica/7891)  
<http://www.wshoy.sidar.org/index.php>  
<http://www.wshoy.sidar.org/index.php?Herramientas>  
<http://www.wsindex.org/>  
<http://xml.mfd-consult.dk/foaf/explorer/>  
<http://xmlns.com/foaf/spec/>  
<https://listas.hispalinux.es/mailman/listinfo/web-semantica-ayuda>  
[www.isotopicmaps.org](http://www.isotopicmaps.org)  
[www.lluiscodina.com/webSemanticaOntologias2007.pdf](http://www.lluiscodina.com/webSemanticaOntologias2007.pdf)  
[www.slideshare.net/daniel.gayo/web-semantica](http://www.slideshare.net/daniel.gayo/web-semantica)

