XML y Web Semántica

Comparativa entre OWLIM y Virtuoso

Trabajo Fin de Carrera

Ingeniería Técnica en Informática de Gestión

Alumna: Cristina Jaén Ruiz-Rozas

Consultor: Joan Anton Perez Braña

11/06/2013

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	XML y Web Semántica. Comparativa entre Virtuoso DB y OWLIM
Nombre del autor:	Cristina Jaén Ruiz-Rozas
Nombre del consultor:	Joan Anton Perez Braña
Fecha de entrega (mm/aaaa):	06/2013
Área del Trabajo Final:	Web semántica
Titulación:	Ingeniería Técnica en Informática de Gestión

Resumen del Trabajo (máximo 250 palabras):

Este trabajo está enmarcado dentro del ámbito de la web semántica.

Por un lado se ha tratado de explicar de manera sencilla qué es la web semántica y cuáles son sus objetivos. Después se ha realizado una aproximación a los antecedentes de la web semántica y por último se ha hecho una descripción de sus componentes, de su arquitectura y del estado actual de la misma.

Relacionadas con la web semántica están las bases de datos con capacidad de almacenamiento de datos semánticos, y que están pensadas específicamente para dar respuesta a las nuevas necesidades de búsqueda de información que se plantean con las nuevas tendencias de la web orientadas a LOD (linked open data).

Se ha realizado un estudio de dos bases de datos: OWLIM y Virtuoso. Para realizar el estudio, se ha creado previamente una ontología relacionada con cursos de fomación, que ha servido de base para la realización de las pruebas necesarias para comparar las dos bases de datos objeto del estudio.

Abstract (in English, 250 words or less):

This work is included in semantic web sphere.

First, it has explained, in a simple way, what semantic web main goals are.

Next, it has dicussed, why semantic databases are so important in new internet field.

Bassicaly, semantic databases are the new manner of storing data wich will be used to give an answer to the new need of seeking information within linked open data on the internet. This data will be available to being used by seeking machines.

So, knowing that nowadays, data can be used, not only by people, but also by software and computers, it has been done a research about two diferent database engines: OWLIM and Virtuoso. Previously, it has been created a ontology about training courses wich has been the basis for testing and comparing the pair of databases subject of the research.

Palabras clave (entre 4 y 8):

Semantic web, xml, rdf, owl, OWLIM, Virtuoso, semantic database, LOD.

Índice

1 Introducción....1
1.1 Contexto y justificación del trabajo....2
1.2 Objetivos del trabajo....2
1.3 Enfoque y método seguido....3
1.4 Elementos del proyecto....4
1.5 Planificación del proyecto....4
1.5.1 Cronograma de tareas....5
1.6 Sumario de productos obtenidos....6
1.7 Descripción de capítulos....6
2 Introducción a la web semántica y ontologías....7
2.1 Antecedentes. Nacimiento de Internet....7
2.2 Accesibilidad web, usabilidad y arquitectura de la información....9
2.3 La web semántica....10
2.4 Arquitectura de la web semántica....12
2.5 Elementos de la web semántica....12
2.5.1 Metadatos....12
2.5.2 Ontologías....13
2.5.3 Estándares de representación semántica más utilizados....14
2.5.3.1 XML (eXtensible Markup Language)....14
2.5.3.2 XML Schema....15
2.5.3.3 RDF....15
2.5.3.4 RDF Schema....16
2.5.3.5 OWL....17
2.5.4 Lenguaje de consulta SPARQL....18
2.6 Situación actual de la web semántica....18
2.6.1 Linked Open data....20
2.6.1.1 OAI-PMH....21
2.6.1.2 SKOS....22
2.6.1.3 RDFa....22
3 Creación de ontología....23
3.1 Protégé....23
3.2 Ontología de cursos....25
3.2.1 Clases Principales de la ontología Cursos....25

3.2.2 Propiedades de la ontología....28
3.2.3 Instancias de la ontología....32
3.2.4 Visualización gráfica de la ontología....34
4 Almacenes de datos para albergar ontologías....35
5 OWLIM....37
5.1 Arquitectura y características de OWLIM-Lite....38
5.1.1 Modo de almacenamiento de OWLIM....39
5.1.2 Limitaciones de OWLIM....39
5.2 Carga de la ontología Cursos en OWLIM....39
5.3 Consultas con SPARQL....42
5.3.1 Consultas SELECT....43
5.3.2 consultas ASK....45
5.3.3 Consultas CONSTRUCT....47
5.3.4 Consultas DESCRIBE....48
5.3.5 Diferencias entre CONSTRUCT y DESCRIBE....49
6 VIRTUOSO DB....50
6.1 Características de Virtuoso....50
6.1.1 Arquitectura....50
6.1.2 Almacenamiento físico....51
6.2 Carga de ontología Cursos en Virtuoso....52
6.3 Consultas con SPARQL....54
6.3.1 Consultas SELECT....55
6.3.2 Consultas ASK....57
6.3.3 Consultas CONSTRUCT....58
6.3.4 Consultas DESCRIBE....59
6.3.5 Índices....59
7 Conclusiones....61
7.1 Acerca de OWLIM....61
7.2 Acerca de Virtuoso....61
7.3 OWLIM vs Virtuoso....62
8 Glosario....63
9 Referencias....66
9.1 Webgrafía....66

9.2 Bibliografía.....68
10 Anexo I.....1
10.1 Instalación de OWLIM.....1
10.1.1 Instalación de Java JDK.....1
10.1.2 Instalación de Tomcat.....1
10.1.3 Instalación de OWLIM propiamente dicha.....3
11 Anexo II.....6
11.1 Instalación de Virtuoso.....6
12 Anexo III.....9
12.1 Fichero RDF, resultado de consulta CONSTRUCT en OWLIM.....9
13 Anexo IV.....12
13.1 Consulta CONSTRUCT en Virtuoso y fichero resultado.....12
14 Anexo V.....13
14.1 Consulta DESCRIBE en Virtuoso y fichero resultado.....13

Índice de ilustraciones

Ilustración 1: Pila Web semántica....11
Ilustración 2: Estándares de representación para la web semántica....14
Ilustración 3: Grafo RDF....16
Ilustración 4: Web actual vs web semántica....19
Ilustración 5: Linked Open Data. Dbpedia....20
Ilustración 6: Metadatos de la Ontología Cursos....24
Ilustración 7: Clases de la ontología Cursos....25
Ilustración 8: Características de la clase AreaFormativa....26
Ilustración 9: Características de la clase Mixto....27
Ilustración 10: Características de la propiedad de tipo object es_creador_de....28
Ilustración 11: Características de la propiedad creado_por....29
Ilustración 12: Características de la propiedad nombre....30
Ilustración 13: Modificación de formulario de la clase Persona....33
Ilustración 14: Introducción de instancias....33
Ilustración 15: Propiedad a visualizar....33
Ilustración 16: Clases y subclases de la ontología Cursos....34
Ilustración 17: Relaciones entre clases....34
Ilustración 18: Visualización gráfica de las entidades....35
Ilustración 19: Arquitectura de OWLIM....38
Ilustración 20: Creación del repositorio....40
Ilustración 21: Características del repositorio....40
Ilustración 22: Información del repositorio creado....40
Ilustración 23: Lista de repositorios....41
Ilustración 24: Carga de datos de la ontología....41
Ilustración 25: Elección del fichero a cargar....41
Ilustración 26: Namespaces de la ontología....42
Ilustración 27: Ejemplo Consulta SPARQL....42
Ilustración 28: Resultado de la consulta de ejemplo....43
Ilustración 29: Resultado de la consulta 1....44
Ilustración 30: Resultado consulta 2....44
Ilustración 31: Resultado consulta 3....45
Ilustración 32: Resultado consulta 4....45
Ilustración 33: Resultado consulta 5 (ask)....46
Ilustración 34: Comprobación consulta 6 (ask)....47

Ilustración 35: Resultado consulta 7 (construct).....48
Ilustración 36: Resultado consulta 8 (describe).....49
Ilustración 37: Arquitectura de Virtuoso.....51
Ilustración 38: Resultado de la carga de la ontología.....52
Ilustración 39: Pantalla de carga de la ontología.....52
Ilustración 40: Resultado tripletas 1.....53
Ilustración 41: Resultado tripletas 2.....53
Ilustración 42: Resultado de consulta en SPARQL.....54
Ilustración 43: Resultado de la consulta con Interactive SQL.....55
Ilustración 44: Resultado consulta 1.....56
Ilustración 45: Consulta 2.....56
Ilustración 46: Resultado consulta 3 (select).....57
Ilustración 47: Resultado consulta 4 (select).....57
Ilustración 48: Comprobación consulta 5 (ask).....57
Ilustración 49: Comprobación consulta 6 (ask).....58
Ilustración 50: Resultado consulta 7.....58
Ilustración 51: Resultado consulta 8.....59
Ilustración 52: Resultado de creación de tabla para índice.....60
Ilustración 53: Procesos corriendo al instalar Tomcat.....2
Ilustración 54: Contenido del zip de instalación de OWLIM.....3
Ilustración 55: Contenido de la carpeta sesame_owlim.....3
Ilustración 56: Desde Tomcat, se observan los procesos de sesame corriendo.....4
Ilustración 57: Comprobación instalación Sesame workbench.....5
Ilustración 58: Comprobación openrdf-sesame.....6
Ilustración 59: Instalación de virtuoso.....7
Ilustración 60: Acceso a Virtuoso desde Conductor.....8
Ilustración 61: Paquetes DAV para Virtuoso.....8
Ilustración 62: Gestión de Virtuoso desde terminal.....9
Ilustración 63: Consulta construct con Virtuoso.....12
Ilustración 64: Consulta DESCRIBE con Virtuoso.....13
Ilustración 65: Fichero resultado consulta Describe.....13

Índice de tablas

Tabla 1: Hitos del proyecto.....4
Tabla 2: Web actual vs Web semántica.....12
Tabla 3: Propiedades de la ontología.....32

Tabla 4: Algunos SGBD semánticos.....36

1 Introducción

Internet es una fuente ingente de información. Pero la búsqueda de contenidos concretos, en muchas ocasiones, no resulta una tarea trivial. Mediante buscadores tratamos de obtener información sobre conocimientos que queremos ampliar, curiosidades, información concreta, como horarios de trenes, o teléfonos,... Pero los resultados que obtenemos no siempre son los esperados y además, no se diferencia cuál es su procedencia: blog, página de contenido html, foro... Y tampoco diferencia el tipo de contenido: si se trata de un audio, de un video o un documento de texto.

Por otra parte, a menudo encontramos información total o parcialmente repetida en múltiples páginas. Esta repetición de información viene justificada por las diferentes necesidades de los editores de páginas web a la hora de mostrar los contenidos en un formato determinado; por esa razón recurren a copiar el contenido de la página generadora de la información original, y a pegarlo en su propia página, dando el formato que desean a dicho contenido.

Esta forma de actuar hace que el contenido de la página copiada, en ocasiones no esté debidamente actualizado. En realidad, a la vez que se actualiza la página origen de los contenidos, todas las webs que han copiado algo de la misma, deberían actualizar también sus contenidos. Pero esa tarea no se suele realizar sistemáticamente.

De estas situaciones se deduce, que pese a la necesidad de presentar los mismos contenidos de diferentes maneras, la fuente de esos datos, debería ser única, y se evitaría así la necesidad de almacenar los mismos datos en múltiples servidores. De esta manera, los sitios que necesitaran disponer de esos datos los obtendrían siempre de la fuente y los presentarían dinámicamente en su propia página, de forma que los datos siempre estarían debidamente actualizados.

Ahora bien, es necesario que los datos de la fuente origen estén en un formato adecuado para que puedan ser debidamente aprovechados por las aplicaciones que quieran disponer total o parcialmente de ellos.

Es evidente que no cualquier tipo de dato es susceptible de ser compartido de una forma inteligente; los contenidos html de las páginas web tradicionales no pueden ser

mostrados de manera inteligente por diferentes servidores a no ser que se les dote de contenido semántico.

Hay que tener en cuenta, que los contenidos a los que dotaremos de semántica estarán definidos dentro de un dominio concreto. Este ámbito o dominio concreto es lo que se denomina ontología.

Por lo tanto, para que estas ontologías puedan ser explotadas, compartidas y utilizadas de manera inteligente desde múltiples puntos de acceso y aplicaciones, han de ser dotadas de contenido semántico, es decir, estos datos tienen que presentarse de una manera determinada para que puedan ser explotados.

Para la representación de los datos con contenidos semánticos nos ayudamos del lenguaje XML[1], que permitirá a las aplicaciones el acceso a datos determinados contenidos dentro de una ontología previamente definida.

El almacenamiento de estos datos con contenido semántico, lo haremos en bases de datos. Existen bases datos específicas pensadas para trabajar con este tipo de contenidos. Entre ellas Virtuoso DB y OWLIM.

1.1 Contexto y justificación del trabajo

El proyecto elegido está enmarcado dentro del área de XML y web semántica, concretamente en la comparación de dos tipos de bases de datos que dan soporte a los contenidos semánticos. La comparativa de estas dos bases de datos: VirtuosoDB y OWLIM, nos permitirá conocerlas más a fondo, y nos permitirá contraponer las ventajas e inconvenientes de cada una de ellas a la hora de ser utilizadas dentro de las web semánticas.

1.2 Objetivos del trabajo

El proyecto consistirá en el análisis de los SGBD[2] Virtuoso DB y OWLIM.

Como objetivos principales del proyecto tenemos:

- Análisis de las bases de datos Virtuoso DB y OWLIM.

1 **XML** según la wikipedia: *eXtended Mark Language,* lenguaje extendido de marcas. XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

2 **SGBD**: Sistema de gestión de base de datos.

Realización de un análisis del funcionamiento de ambas bases de datos en diferentes circunstancias mediante la utilización de una ontología que ayudará a realizar las comparaciones.

- Creación de una ontología para poder realizar la comparativa de las bases de datos mencionadas.

  La ontología que se creará estará basada en los cursos de formación que imparte el IVAP (Instituto Vasco de Administración Pública) y que están dirigidos a los empleados públicos de la CAPV. Está ontología se utilizará para la creación de las instancias y los metadatos que permitirán realizar el análisis de los dos SGBD.

- Ventajas e inconvenientes de cada una de las bases de datos

  Del análisis comparativo se obtendrán una serie de ventajas e inconvenientes de la utilización de cada una de las dos bases de datos analizadas.

- Conclusiones del estudio comparativo

  En base a los análisis efectuados, se tratará de llegar a una conclusión motivada de una elección de un sistema u otro a la hora de realizar un hipótetico trabajo relacionado con las bases de datos semánticas.

1.3 Enfoque y método seguido

Dentro de este punto se describen a grandes rasgos los pasos que se han seguido para realizar la comparativa entre las bases de datos semánticas OWLIM y Virtuoso.

Previa a la instalación de software y a la realización de las diferentes pruebas objeto de este trabajo, se ha realizado una lectura y búsqueda de información acerca de la web semántica en general y de su estado actual.

A continuación se han llevado a cabo las tareas técnicas:

- Instalación de Protégé y creación de la ontología.
- Instalación de los dos SGBD en sendas máquinas virtuales.
- Carga de los datos de la ontología en los dos SGBD.
- Creación de diferentes tipos de consultas SPARQL en ambos SGBD.

- Comparación de la facilidad de uso, funcionamiento, ventajas e inconvenientes de la utilización de cada una de las bases de datos.
- Conclusiones de la comparativa

1.4 Elementos del proyecto

A continuación se describe el material hardware y software utilizado para la realización de este trabajo.

- Ordenador Dell Vostro con procesador Intel Core I5 y 4Mb de RAM.
- Sistema Operativo Ubuntu 12.04
- Software de gestión de proyectos, OpenProj
- Software para la creación de ontologías Protégé versión 3.5_beta.
- Dos máquinas virtuales VirtualBox Ubuntu server 11.04.
- Software Virtuoso Open Source version.
- Software OWLIM Lite versión Open Source.
- Datos de los cursos del IVAP para crear la ontología de ejemplo.
- Software de oficina Libre Office para la redacción y presentación del proyecto.
- Software de audio y video Kazam para la creación de la presentación virtual.

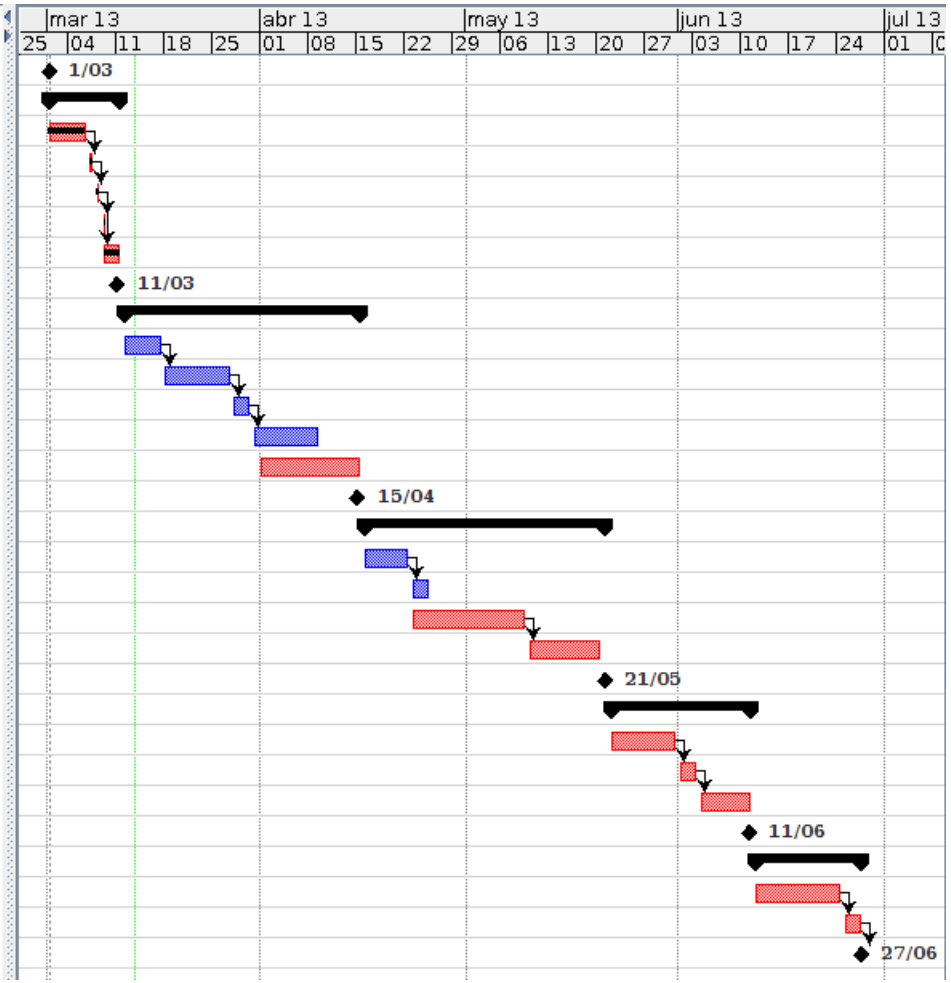
1.5 Planificación del proyecto

Hito	Descripción	Fecha de entrega
PEC1	Plan de trabajo	11/03/2013
PEC2	Estado del Arte	15/04/2013
PEC3	Borrador documento final	21/05/2013
Entrega final	Memoria del trabajo fin de carrera	11/06/2013
Debate virtual	Defensa del trabajo realizado	entre el 25 y el 27 de junio

Tabla 1: Hitos del proyecto

1.5.1 Cronograma de tareas

		Nombre	Duración	Inicio	Terminado
1	✔	Inicio del proyecto	0 days	1/03/13 8:00	1/03/13 8:00
2	✔	**⊟Plan de trabajo**	**11 days**	**1/03/13 8:00**	**11/03/13 17:00**
3	✔	Recopilación información preliminar	6 days	1/03/13 8:00	6/03/13 17:00
4	✔	Análisis opciones	1 day	7/03/13 8:00	7/03/13 17:00
5	✔	Elección opción de trabajo	1 day	8/03/13 8:00	8/03/13 17:00
6	✔	Confección calendario	1 day	9/03/13 8:00	9/03/13 17:00
7	✔	Redacción del documento	3 days	9/03/13 8:00	11/03/13 17:00
8		Entrega PEC1	0 days	11/03/13 8:00	11/03/13 8:00
9		**⊟Estado del arte**	**35 days**	**12/03/13 8:00**	**15/04/13 17:00**
10		Estudio del estado del arte	6 days	12/03/13 8:00	17/03/13 17:00
11		Instalación del software	10 days	18/03/13 8:00	27/03/13 17:00
12		Extracción de datos	3 days	28/03/13 8:00	30/03/13 17:00
13		Creación de metadatos e instancias	10 days	31/03/13 8:00	9/04/13 17:00
14		Redacción del documento	15 days	1/04/13 8:00	15/04/13 17:00
15		Entrega PEC2	0 days	15/04/13 7:00	15/04/13 8:00
16		**⊟Borrador documento final**	**35 days**	**16/04/13 8:00**	**21/05/13 8:00**
17		Revisión de la ontología	7 days	16/04/13 8:00	22/04/13 17:00
18		Creación de gráficos	3 days	23/04/13 8:00	25/04/13 17:00
19		Creación del borrador	17 days	23/04/13 8:00	9/05/13 17:00
20		Ejemplo de uso	11 days	10/05/13 8:00	20/05/13 17:00
21		Entrega PEC3	0 days	21/05/13 8:00	21/05/13 8:00
22		**⊟Entrega final**	**21 days?**	**22/05/13 8:00**	**11/06/13 17:00**
23		Redacción de la memoria final	10 days	22/05/13 8:00	31/05/13 17:00
24		Supervisión y corrección	3 days	1/06/13 8:00	3/06/13 17:00
25		Creación de la presentación virtual	8 days	4/06/13 8:00	11/06/13 17:00
26		Entrega final	0 days?	11/06/13 8:00	11/06/13 8:00
27		**⊟Debate Virtual**	**16 days**	**12/06/13 8:00**	**27/06/13 17:00**
28		Preparación de la defensa del TFC	13 days	12/06/13 8:00	24/06/13 17:00
29		Inicio del debate	3 days	25/06/13 8:00	27/06/13 17:00
30		Fin del debate	0 days	27/06/13 17:00	27/06/13 17:00



1.6 Sumario de productos obtenidos

- La ontología de cursos creada en Protégé
- Memoria con la descripción del desarrollo del proyecto y las conclusiones.
- Presentación del proyecto
- Video con la descripción del desarrollo del proyecto.

1.7 Descripción de capítulos

A continuación se describen los capítulos que conforman el presente trabajo.

Capítulo 1. Se realiza una descripición muy general del proyecto y se muestra la planificación seguida, para el desarrollo del mismo.

Capítulo 2. Estado del arte. Se analizan los antecedentes y la situación actual.

Capítulo 3. Se describe el proceso de creación de la ontología de cursos mediante la herramienta Protégé 3.5 Beta. Se especifican los elementos que forman la ontología.

Capítulo 4. Se enuncian las características de los almacenes de datos para albergar ontologías y se listan las bases de datos semánticas con soporte completo SPARQL.

Capítulo 5. Se explica la arquitectura, modo de almacenamiento y funcionamiento de la base de datos OWLIM. Se analizan los distintos tipos de consultas SPARQL que se pueden realizar.

Capítulo 6. Se expone la arquitectura, modo de almacenamiento y funcionamiento general de Virtuoso. Se exponen ejemplos de consultas SPARQL. Se describen los índices.

Capítulo 7. Conclusiones del análisis de las dos bases de datos. Ventajas e inconvenientes de cada una de ellas.

Capítulo 8. Referencias. Webgrafía y bibliografía.

Anexo I Instalación de OWLIM

Anexo II. Instación de Virtuoso DB

Anexo III Fichero RDF Resultado de consulta CONSTRUCT en OWLIM.

2 Introducción a la web semántica y ontologías

2.1 Antecedentes. Nacimiento de Internet

La idea de la interconexión de ordenadores mediante una red que los uniera era algo que ya existía desde finales de los años cincuenta, es decir poco después de la construcción de las primeras computadoras.

La primera red existente para interconectar ordenadores que estaban en distintos puntos, se realizó con fines militares en Estados Unidos en los años sesenta. Era la época de la guerra fría, y por lo tanto el objetivo de esta red era poder acceder a la información militar desde cualquier punto del país. Esta red se llamaba ARPANET[3], y en principio tuvo fines militares. Posteriormente, MILNET[4], una porción de esta red inicial, se desgajaría para dedicarse exclusivamente a asuntos militares y de espionaje, mientras que ARPANET continuó su andadura con fines académicos y de investigación.

Al mismo tiempo que se desarrollaba la red ARPANET, se comenzó a investigar en la forma de transmisión de los datos. Se hablaba ya, de la conmutación de paquetes. Los paquetes constan de dos partes: por un lado está la información que se quiere transmitir propiamente dicha y por otro lado, están los datos de control, mediante los cuales se indica a dónde se quiere transmitir esa información.

En 1969 había 4 hosts[5] conectados en ARPANET. Posteriormente se fueron añadiendo más hosts y paralelamente, se siguió investigando en el protocolo de red para la comunicación de los equipos.

ARPANET utilizó diferentes protocolos de transmisión de datos, entre ellos NCP[6]. En 1983, influenciados por el cambio que de protocolo de comunicaciones que ya se

3 ARPANET. (Advanced Research Projects Agency Network) Red de computadoras creada por encargo del Departamento de Defensa de los Estados Unidos como medio de comunicación para los diferentes organismos del país.

4 MILNET (MILitary NETwork) red de comunicación militar de las Fuerzas armadas de los Estados Unidos.

5 Host. Se refiere a las computadoras conectadas a una red.

6 NCP (Network Control Program) Conjunto original de protocolos de control de red de ARPANET

había realizado en MILNET, ARPANET adopta el protocolo TCP/IP[7], que hoy en día todavía se sigue utilizando.

ARPANET fue el germen inicial de INTERNET, con la filosofía de red de arquitectura abierta, que mantiene hoy en día. Es decir, es una red capaz de interconectar muchas redes de diferente naturaleza. En la red de arquitectura abierta, las redes individuales se diseñan por separado, pero disponen de una interfaz que ofrecen a usuarios y proveedores para que estas redes individuales puedan ser interconectadas.

Aunque, en 1985 ya se puede considerar que INTERNET había nacido, porque había una amplia comunidad de usuarios de universidades y de investigadores que utilizaban esta red compartida, hasta la aparición del protocolo http no se empezó a utilizar de una forma masiva por millones de usuarios de todo el mundo.

En 1990 Tim Berners-Lee y Robert Cailliau, presentan la primera propuesta del Word Wide Web[8]. Instalaron en el CERN[9] el primer servidor web y el primer navegador y lo abrieron al mundo. Tanto el software creado para el servidor, como el del navegador fueron liberados. Este hecho hizo que se difundiera rápidamente esta tecnología. Muestra de ello, es que en 1992 había 26 servidores Web y en 1995 pasaron a ser 200. A partir de ese momento su número creció de forma exponencial hasta llegar a la gran red de servidores que existe hoy en día.

En 1991 se publicó la primera descripción de HTML[10] que se basa en la utilización de etiquetas y de hiperenlaces. Para poder identificar unívocamente los enlaces se utilizaron las URL[11] que nos permiten localizar e identificar distintos recursos en la web.

Al mismo tiempo que crecía el número de servidores web se fueron creando diferentes navegadores. El primero que tuvo interfaz gráfica y un relativo éxito fue *Mosaic* en

7 TCP/IP. Es una familia de protocolos de red en los que se basa Internet. Los protocolos más más importantes que la componen son: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

8 World Wide Web. Red informática mundial

9 CERN (Conseil Européen pour la Recherche Nucléaire) Organización Europea para la Investigación Nuclear

10 HTML. Hypertext markup lenguage. Lenguaje de marcado hipertextual

11 URL (Uniform Resource Locator) Localizador de recursos uniforme. Se utiliza para nombrar recursos en Internet.

1993, aunque al poco tiempo, en 1994, fue sustituido por el popular *Netscape Navigator*.

Los buscadores fueron elementos clave, ya que dotaron a la red de redes de su gran capacidad de búsqueda. Los primeros fueron *Archie* y *Gopher*. A continuación surgieron *Lycos*, *Altavista* y *Yahoo*. Estos dos últimos, fueron los más populares entre 1995 y 2001, hasta la aparición del todopoderoso *Google*.

Para velar por el desarrollo de Internet en general, en 1994 el propio Berners-Lee creó el consorcio W3C[12] que trabaja para que Internet pueda utilizar su máximo potencial mediante el desarrollo de estándares.

Uno de los aspectos de los que se ha ocupado el W3C en los últimos años ha sido, la accesibilidad web, la usabilidad y la arquitectura de la información.

2.2 Accesibilidad web, usabilidad y arquitectura de la información

La **accesibilidad we**b trata de la capacidad de las personas para acceder a los contenidos de las páginas web, independientemente de sus discapacidades (físicas, cognitivas, intelectuales).

Este aspecto se comenzó a tratar como algo dirigido a personas que tenían discapacidades reconocidas, pero posteriormente se constató que también se mejoraba la accesibilidad del resto de usuarios.

Algunos elementos que facilitan la accesibilidad son: disponer de textos escalables, publicar textos o archivos alternativos para imágenes y videos, añadir subtítulos a los videos, jerarquizar los elementos html, usar adecuadamente las tablas y las listas, ...

La accesibilidad está muy ligada con la **usabilidad**, o facilidad de uso, de los diferentes sitios. De esta forma no sólo se trata de facilitar la accesibilidad a los sitios para todos los usuarios, sino también de hacer más fácil la comprensión de los enlaces, la ubicación de los contenidos, el uso de los formularios, etc. A su vez, este último aspecto, de *facilitar el uso*, está muy ligado con el **diseño centrado en el usuario** (DCU[13]).

12 W3C (World Wide Web Consortium) consorcio internacional que realiza recomendaciones para la web.

13 DCU. Diseño centrado en el usuario

Hay que tener en cuenta que el DCU, no es sólo una corriente que se aplica a páginas web y aplicaciones, sino que es una disciplina que viene empleándose en todos los aspectos del diseño de cualquier producto que se consuma. El diseñador amplía el ámbito de su quehacer, que ya no se limitará a inventar un determinado producto, sino que además habrá de constatar que su producto es de fácil uso y para ello tendrá que posibilitar una cierta interacción con el usuario final, para poder comprobar este hecho.

En el ámbito del las páginas web y de las aplicaciones, el diseñador deberá conocer la forma de actuar del usuario, para ello necesitará algún tipo de *feed-back* o retroacción que le permita conocer los puntos débiles y fuertes de los productos que está diseñando.

Por otra parte, otro aspecto que a tener en cuenta en la construcción de los sitios, es la **arquitectura de la información**. La información se considera correctamente representada en función de los siguientes puntos.

- Organización y etiquetado de la información. Esquemas de navegación lógicos.
- Diseño estructural que facilite la finalización de las tareas. Se debe poder acceder a la tarea que se quiere realizar de una forma intuitiva.
- El diseño debe ser atractivo y debe facilitar la búsqueda de la información deseada.

Hasta ahora, en este trabajo, se ha descrito el desarrollo de la web, desde el punto de vista de las personas; pero, hay otro aspecto que se valora cada vez más, y que está relacionado con el acceso a los datos por parte de las máquinas. Además, está ligado con la inteligencia artificial y con la posibilidad de relacionar contenidos dispersos en diferentes servidores de la red y presentar dichos resultados agrupados, relacionados de una forma semántica. De este último aspecto se ocupa la **web semántica**.

2.3 La web semántica

La web semántica trata de relacionar contenidos existentes en la red y presentarlos de forma agrupada, y en ocasiones infiriendo nueva información. Un ejemplo de lo que obtendríamos mediante la web semántica sería lo siguiente: se publica una oferta de cursos dirigidos a empleados de la administración. En estos cursos se indica el título del curso y quiénes serán los ponentes de los mismos. Si contáramos con la web

semántica, además de mostrarnos esos datos, podría haber un razonador que ligara a cada uno de los ponentes con los libros que han publicado, ya que esa es una información que existe en la red. Para lograr enlazar estos datos, tienen que estar construidos de una manera especial de forma que puedan ser entendidos y procesados por un razonador. Esta máquina (razonador) deducirá qué libros o artículos han publicado los ponentes, y, si están relacionadas con el tema del curso en cuestión, nos los mostrará. El nuevo sistema, tendrá capacidad de inferir una información. Al ser capaz de "razonar o deducir" algo, se puede hablar de sistemas próximos a la inteligencia artificial.

Según Berners-Lee y Miller (2002): *La Web Semántica es una extensión de la actual Web en la que a la información disponible se le otorga un significado bien definido que permita a los ordenadores y las personas trabajar en cooperación*.

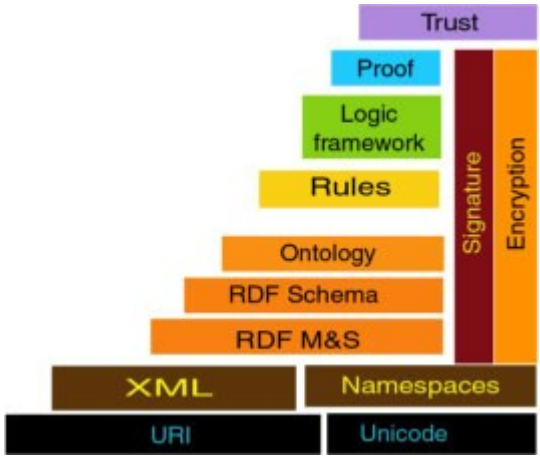


Ilustración 1: Pila Web semántica

La web semántica se basa en la idea de proporcionar datos definidos y enlazados permitiendo que aplicaciones heterogéneas localicen, integren, razonen y reutilicen la información presente en la web. El consorcio W3C, en colaboración con un gran número de investigadores y socios industriales, se encarga de la definición de estándares y tecnologías que permitan alcanzar los objetivos mencionados anteriormente.

Dentro de la web semántica la información es comprendida por las máquinas. Está dotada de etiquetas (metadatos[14]) que hacen que las búsquedas sean más eficientes.

No es un una web dirigida única y exclusivamente a las personas, como lo ha sido hasta ahora, sino que se basa en la interconectividad, en la interrelación y la estructuración de los contenidos de forma que puedan ser enlazados, mostrados y tratados de distintas maneras. Se facilitan procesos automáticos y se optimizan las búsquedas ya que se utilizan otros sistemas más allá de los puramente textuales.

14 Metadatos. Datos que describen a otros datos

2.4 Arquitectura de la web semántica

La arquitectura inicial de la web estaba basada en URL, HTTP y HTML. El consorcio W3C está transformando la arquitectura de la web para adaptarla a las necesidades de la web semántica que está basada en URI, HTTP y XML

Mediante la tabla representada a continuación se comparan los elementos de la web tradicional con los elementos básicos de la web semántica.

	Web original	Web semántica	
Localizador	URL	URI	**URI** es más amplio que **URL**. URI es un identificador más completo que URL y nos permite incluir dentro de una dirección, una subdirección para poder acceder a un fragmento concreto del recurso al que se hace referencia.
Protocolo	HTTP	HTTP	Es el protocolo de transferencia de hipertexto.
Lenguaje	HTML	XML	**HTML** es el lenguaje de marcado para las páginas web, en cambio **XML** es un lenguaje de marcado extendido que se utiliza también en bases de datos y para intercambio de datos.

Tabla 2: Web actual vs Web semántica

2.5 Elementos de la web semántica

Los elementos principales de la web semántica son: metadatos, ontologías, estándares de representación y lenguajes de consulta.

2.5.1 Metadatos

Los metadatos describen los recursos de información, siguiendo un determinado conjunto de reglas. Su objetivo es la descripción de recursos de Internet.

Los metadatos son descripciones estructuradas que describen las características y propiedades de los recursos para que sea fácil su recuperación, localización, valoración y administración. Los metadatos además de describir los recursos también son capaces de describir los procesos, componentes, restricciones y relaciones de dichos recursos.

DCMI[15], es la organización que se encarga de la definición de estándares para definir los metadatos. Los objetivos que persigue esta organización son los siguientes: simplicidad, interoperabilidad, consenso internacional, flexibilidad e independencia sintáctica.

2.5.2 Ontologías

Una ontología es la descripción detallada de un esquema conceptual dentro de un dominio dado. La ontología describe tipos de entidades y la forma en que éstas se relacionan entre sí. Las ontologías permiten desarrollar estructuras complejas para simplificar la comunicación entre personas y máquinas.

Los elementos de una ontología son:

- **Clases o conceptos**
  Son las ideas básicas de una ontología. Se organizan en taxonomías o dominios.
- **Atributos**
  Representan la estructura interna de los conceptos, se caracterizan por el dominio en el cual toman el valor.
- **Relaciones**
  Son las interacciones entre conceptos del mismo dominio. Forman la taxonomía del dominio.
- **Funciones**
  Son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.

15 DCMI. Dublin Core Metadata Iniciative. Organización cuyo objetivo es el fomento del uso de metadatos mediante el uso de estándares interoperables con vocabularios especializados, para describir recursos de la información. Ha creado el modelo de metadatos Dublin Core.

- **Instancias**
  Representan individuos concretos de un concepto o clase.
- **Axiomas**
  Son expresiones ciertas o reglas de la ontología. Estas expresiones se declaran sobre relaciones que deben cumplir los elementos de la ontología.

2.5.3 Estándares de representación semántica más utilizados

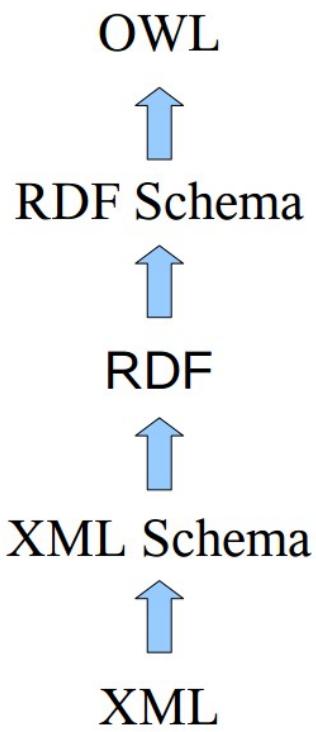


Ilustración 2: Estándares de representación para la web semántica

Con el fin de que los datos sean legibles por el software de las máquinas, se utilizan las URI[16] como identificadores únicos de recurso y el lenguaje XML (es un lenguaje de marcas extendido con más capacidades que HTML).

Por encima del estándar XML, apareció RDF[17] y posteriormente RDFS[18]. Por último apareció un lenguaje aún más avanzado aún, OWL, que aporta el vocabulario estándar para representar ontologías completas.

A continuación se describe cada uno de los estándares, desde el más sencillo al más completo.

2.5.3.1 XML (eXtensible Markup Language)

Es un metalenguaje para la definición de lenguajes que permiten el intercambio de datos estructurados correctamente, con independencia de la plataforma.

16 URI. Uniform Resource Identifier. Identificador único de recurso.

17 RDF. Resource Description Framework. Modelo de datos simple que permite identificar recusos y sus relaciones con otros recursos.

18 RDFS. RDF Schema. Vocabulario para describir las propiedades y las clases de los recursos RDF, permitiendo establecer jerarquías.

Permite estructurar datos en forma de árboles de etiquetas con atributos. Es un lenguaje de marcas derivado de SGML[19] que permite definir la gramática de lenguajes específicos.

Un documento XML contiene datos cuya estructura se define a partir de lo que se conoce como DTD[20].

2.5.3.2 XML Schema

XML Schema es la extensión del lenguaje XML que permite definir la estructura de los documentos XML.

XML Schema para solventar las limitaciones de las DTD, utiliza el propio lenguaje XML, para integrar las definiciones. Utiliza lo que se denominan Namespaces.

2.5.3.3 RDF

RDF es el lenguaje que nos permite representar los metadatos. Es un modelo para la descripción estructurada de recursos con un formato **sujeto-predicado-objeto**.

- El **sujeto** es el recurso a describir.
- El **predicado** es una propiedad o relación del recurso.
- El **objeto** es el valor asignado a esa propiedad o el recurso que estable la relación.

Una característica de RDF es que pueden ser representados como grafos.

19 SGML Standard Generalized Markup Language. Estándar de Lenguaje de Marcado Generalizado. Consiste en un sistema para la organización y etiquetado de documentos. Sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.

20 DTD. Document Type Definition. Es una definición de un documento SGML o XML que especifica restricciones en la estructura y en la sintaxis del mismo.

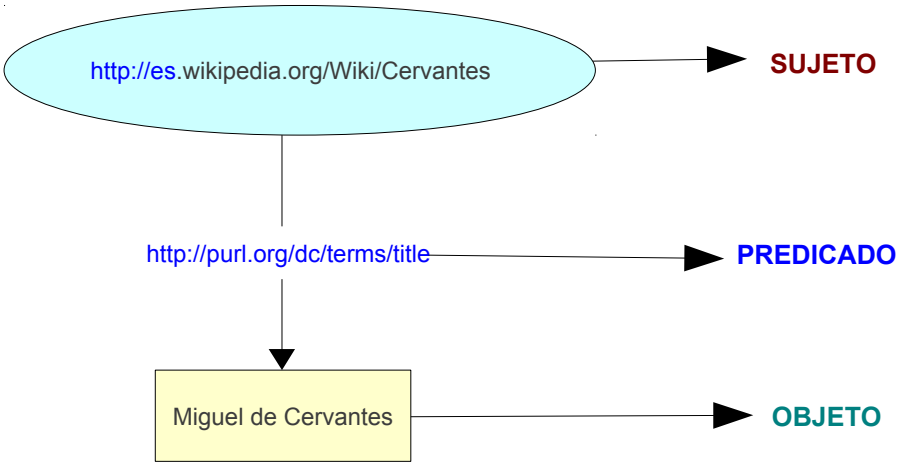


Ilustración 3: Grafo RDF

Los elementos principales de RDF son:

- **Recurso (sujeto)**: es lo que puede ser definido mediante una expresión RDF. Todos los recursos son identificados mediante su URI.
- **Propiedades (predicado)**: características para definir un recurso.
- **Sentencias (objeto)**: Es el valor asociado a la propiedad. Las sentencias son relaciones binarias entre dos elementos (relación entre sujeto y predicado). El valor asociado a la sentencia, puede ser otro recurso; en ese caso se identificará mediante su URI.

RDF utiliza los espacios de nombres XML. Estos se aplican tanto a los atributos como a los elementos de los recursos. Por ejemplo, para definir un espacio de nombres al que pertenece un elemento, hay que añadir a la definición del elemento, un atributo. El nombre del atributo será xmlns, es decir, *namespace xml*, y el valor puede ser una cadena de texto o una URI.

2.5.3.4 RDF Schema

RDF Schema es una ampliación de RDF que permite establecer jerarquías de generalización entre propiedades y recursos.

Con RDF Schema se pueden describir esquemas sencillos utilizando clases y subclases. También se pueden definir propiedades , su dominio de aplicación y su rango de valores posibles. RDFS nos permite el modelado de ontologías que no sean muy complicadas.

2.5.3.5 OWL

Es una extensión de RDF Schema.

Es un lenguaje desarrollado por el consorcio W3C. Su objetivo es la publicación y compartición en la web de ontologías.

Las ontologías son muy importantes porque bien definidas nos permiten diseñar y ejecutar procesos de inferencia.

OWL está por encima de RDF Schema, ya que además de las propiedades de RDFS añade otras propiedades como la relación inversa, y dispone de un conjunto adicional de axiomas que nos permite definir restricciones como la cardinalidad. También tiene capacidad para realizar combinaciones de clases (intersección, unión y complemento).

Cumple las siguientes características

- Capacidad de ser distribuido entre diferentes sistemas.
- Escalabilidad
- Compatibilidad con los estándares web para hacer posible la accesibilidad y la internacionalización.
- Es abierto y extensible

Los **elementos básicos** de OWL son los siguientes:

- **Clases e instancias**. Se definen mediante RDFS
- **Propiedades**. Se utiliza XMLS o RDFS.
- **Características de las propiedades**. Se pueden definir propiedades
    - transitivas
    - simétricas
    - funcionales
    - inversas
    - inversas simétricas
- **Restriciones** de propiedades
- **Mapeado** de ontologías.

  Operaciones sobre clases.

Hay 3 divisiones del lenguaje OWL:

- **OWL Lite**. Nos permite tener una clasificación jerárquica y restricciones simples.
- **OWL DL**. Nos proporciona las máxima expresividad de la ontología pero garantizándonos que podrá ser utilizada computacionalmente.
- **OWL Full**. Máxima expresividad y libertad a la hora de crear la ontología pero no garantiza la compatibilidad computacional.

  Para utilizar dentro del contexto de la web semántica se suele utilizar OWL DL, por proporcionar una gran flexibilidad en la representación sin perder capacidad computacional.

2.5.4 Lenguaje de consulta SPARQL

El lenguaje de consulta SPARQL[21] es el lenguaje estándar de consulta que se utilizar para los almacenes semánticos, tal y como SQL es el estándar para las bases de datos relacionales. Es el estándar oficial de W3C desde 2008 y permite expresar patrones de tripletas que se utilizan para encontrar correspondencias en el conjunto de sentencias RDF de un grafo.

Mediante SPARQL se dispone de cuatro formas de consulta:

– SELECT -> Utilizado para realizar selección de datos

– CONSTRUCT -> Sirve para generar nuevas tripletas RDF

– ASK -> Empleado para realizar preguntas. Devuelve TRUE o FALSE.

– DESCRIBE -> Devuelve un grafo RDF que describe los recursos encontrados.

2.6 Situación actual de la web semántica

Se empezó a hablar de las tecnologías de la web semántica hacia el año 1999.

La **accesibilidad web**, el **diseño centrado en el usuario** y la **arquitectura de la información**, se pueden considerar precursores de la aparición del concepto de web semántica. Estos tres aspectos ya están siendo utilizados cada vez de forma más amplia en la construcción de los nuevos sitios web. Los diseñadores web tratan de que sus sitios sean accesibles. En relación a la accesibilidad, gracias al impulso de leyes y decretos gubernamentales y autonómicos, que exigen el cumplimiento de la *pauta*

21 SPARQL. Protocol and RDF Query Language

WAI[22] *AA*, la mayoría de páginas web de administraciones públicas tienen un gran nivel de accesibilidad e interoperabilidad. En cuanto a los otros aspectos, las empresas dedicadas al desarrollo de páginas web, están introduciendo y aplicando el concepto de DCU, mejorando la interacción entre personas y máquinas, y prestando más atención a la arquitectura de la información, porque han constatado que las páginas así creadas son más utilizadas y visitadas.

En cambio, la aplicación de los conceptos propios de web semántica, es más compleja, y requiere de una adaptación más ardua, para lograr contenidos web que contengan significado.

Durante todos estos años, la web original ha ido creciendo exponencialmente, pero pese a las recomendaciones del W3C, la introducción de elementos de web semántica no se está adoptando de forma masiva. No obstante, cada vez más instituciones están explorando las posibilidades de la web semántica.

Con el advenimiento de las nuevas leyes de transparencia que se están implantando cada vez en más gobiernos, se plantea la posibilidad de que los datos que se ponen a disposición de los ciudadanos cumplan con los requisitos de la web semántica, fundamentados en la reutilización de los datos y el acceso a los mismos por máquinas o software con capacidades de búsqueda y de aprendizaje.

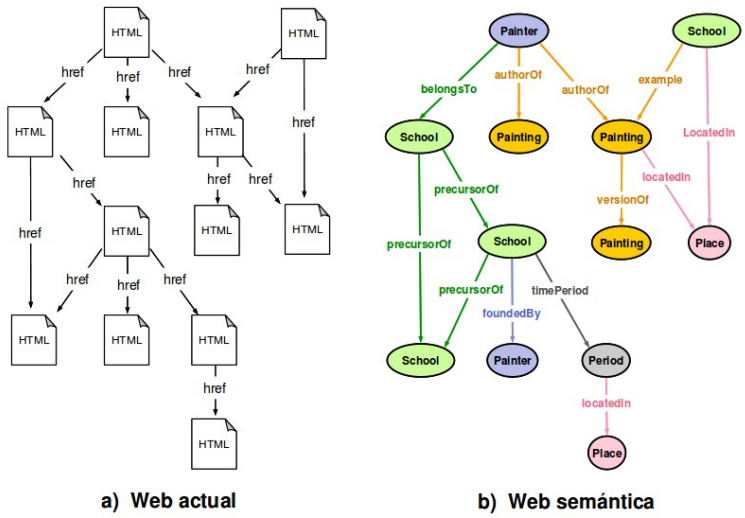


Ilustración 4: Web actual vs web semántica

22 WAI (Web Accessibility Initiative) Iniciativa para la Accesibilidad Web es una rama de W3C que trabaja la accesibilidad de la Web. Existen diferentes pautas de accesibilidad: A, AA y AAA. Actualmente en la administración se trata de alcanzar la “doble A”.

2.6.1 Linked Open data

LOD[23], o linked Open Data es una técnica para publicar información estructurada en la web. Se trata de poder utilizar datos, que son mantenidos por diferentes entidades, para poder referenciarlos y reutilizarlos libremente.

El consorcio W3C ya ha definido una serie de formatos y protocolos estándares abiertos.

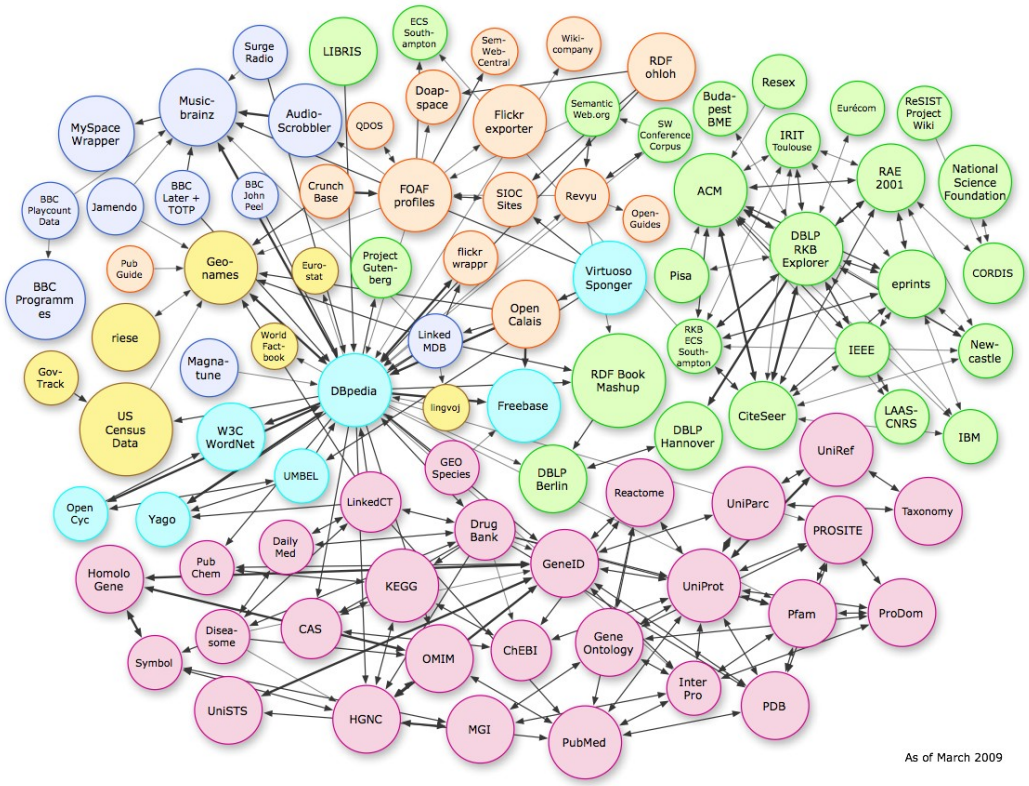


Ilustración 5: Linked Open Data. Dbpedia

La innovación fundamental de LOD, es que los datos son publicados en la web pero están pensados para que puedan ser accedidos para lectura, interpretación y procesado automático por parte de software informático.

Se pueden encontrar ejemplos de LOD consultando el registro de Open Knowldedge Foundation[24].

23 LOD. Linked Open Data. Datos abiertos enlazados. Técnica para publicar información estructurada en la web. Consulta de datos LOD: http://thedatahub.org/

24 http://thedatahub.org/

2.6.1.1 OAI-PMH

El protocolo **OAI-PMH**[25] (Open Archives Initiative-Protocol Metadata Harvesting), es una herramienta de interoperabilidad independiente de la aplicación que permite realizar el intercambio de información para que desde puntos (proveedores de servicio), se puedan hacer búsquedas que abarquen la información recopilada en distintos repositorios asociados (proveedores de datos).

Open Journal Systems[26], Greenstone[27] (software), EPrints[28] y DSpace[29], son compatibles de manera nativa con este protocolo.

25 OAI-PMH. Open Archives Initiative-Protocol Metadata Harvesting. Herramienta de interoperabilidad independiente de la aplicación. Permite realizar el intercambio de información. Desde proveedores de servicio se puedan hacer búsquedas en la información recopilada en distintos repositorios asociados.

26 Open Journal Systems (OJS) es un software de código abierto para la administración de revistas creado por el Public Knowledge Project, bajo licencia GNU General Public License. Fue diseñado para facilitar el desarrollo de publicaciones de acceso libre, proveyendo la infraestructura técnica para la presentación en línea de artículos de revista y el flujo editorial completo. http://pkp.sfu.ca/espanol, http://www.ojs.es/

27 Greenstone es un software de código abierto para el desarrollo de bibliotecas digitales y repositorios institucionales en varios idiomas y distribuido bajo la Licencia Pública General GNU. http://www.greenstone.org/index_es

28 EPrints. Es paquete de software open-source dedicado a construir repositorios de acceso directo compatibles con OAI-PMH. Muchas de sus características son comunes a los sistemas de gestión documental pero es utilizado fundamentalmente en repositorios institucionales y diarios científicos. http://www.eprints.org/

29 DSpace es un software de código abierto que provee herramientas para la administración de colecciones digitales. Se suele utilizar para repositorios institucionales. Soporta una gran variedad de datos: libros, tesis, fotografías, filmes, video, datos de investigación, etc. Los datos son organizados como ítems que pertenecen a una colección; cada colección pertenece a una comunidad. Está liberado bajo licencia BSD que permite a los usuarios personalizar o extender el software según se necesite. http://www.dspace.org/

Los metadatos a transmitir vía OAI-PMH deberán codificarse en Dublin Core[30] sin calificar con objeto de minimizar los problemas derivados de las conversiones entre múltiples formatos.

2.6.1.2 SKOS

SKOS[31], Simple Knowledge Organization System, tiene por objetivo proporcionar un modelo para representar distintos tipos de vocabularios controlados, como tesauros, clasificadores y taxonomías. Es una ontología OWL Full pero tiene un carácter muy simple y no contiene una semántica formal y completa. Los elementos de SKOS son clases y propiedades.

Con SKOS se incrementa la interoperabilidad del Sistema de Organización del Conocimiento, KOS[32], para generalizar los lenguajes documentales.

Se trata de un estandar sencillo para publicar KOS en la web semántica. Resulta mucho más sencillo su diseño que las ontologías OWL.

2.6.1.3 RDFa

RDFa, permite insertar declaraciones RDF en etiquetas XHTML en forma de atributos. De esta forma se pueden insertar declaraciones RDF en las etiquetas de documentos XHTML, de forma que puedan coexistir en la misma URL. Así existirá en la misma URL una versión adaptada para su consulta por parte de las personas y otra, adecuada para que sea procesada por las máquinas.

30 Dublin Core. DC. Modelo de metadatos elaborado y auspiciado por la DCMI (Dublin Core Metadata Initiative), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos. Las implementaciones de DC usan generalmente XML y se basan en RDF.

31 SKOS. Simple Knowledge Organization System, tiene por objetivo proporcionar un modelo para representar distintos tipos de vocabularios controlados como tesauros, clasificadores y taxonomías.

32 KOS. Knowldedge Organization System. Sistema de organización del conocimiento, ampliamente utilizado para generalizar lenguajes documentales.

3 Creación de ontología

Para poder llevar a término el objetivo de este proyecto, es necesario la creación de una ontología que nos servirá como base para el testeo de las dos bases de datos objeto de la comparativa.

Se ha instalado la herramienta de creación de ontologías Protégé.

3.1 Protégé

La versión de Protégé utilizada es la 3.5 Beta con plugin Jambalaya.

Protégé es un editor de ontologías de código abierto. Es uno de los editores más utilizados y permite la construcción de ontologías sobre RDFS, OWL y XML Schema.

Protégé permite la creación de ontologías, bien en base a frames y slots o en base al lenguaje OWL.

En este trabajo se utilizará Protégé-OWL porque es el que nos permite crear una ontología en OWL que es el estándar recomendado por W3C.

Protégé está basado en Java, por lo que es multiplataforma. Es open-source, extensible y tiene gran cantidad de plugins. Las ontologías que genera pueden ser exportadas a RDFS, OWL y XML Schema.

Se arranca Protégé desde el terminal tal y como se muestra a continuación.

```
$ cd Protege_3.5_beta/
$ ./run_protege.sh
```

Después podemos abrir la ontología de cursos

En la pestaña de metadatos podemos visualizar las siguientes características:

- La URI o dirección única de recurso.

- Los *namespaces*[33] con sus prefijos. Los prefijos son muy útiles ya que podemos referirnos a un namespace por su prefijo en lugar de por su nombre completo, lo que simplifica el manejo de los namespaces.

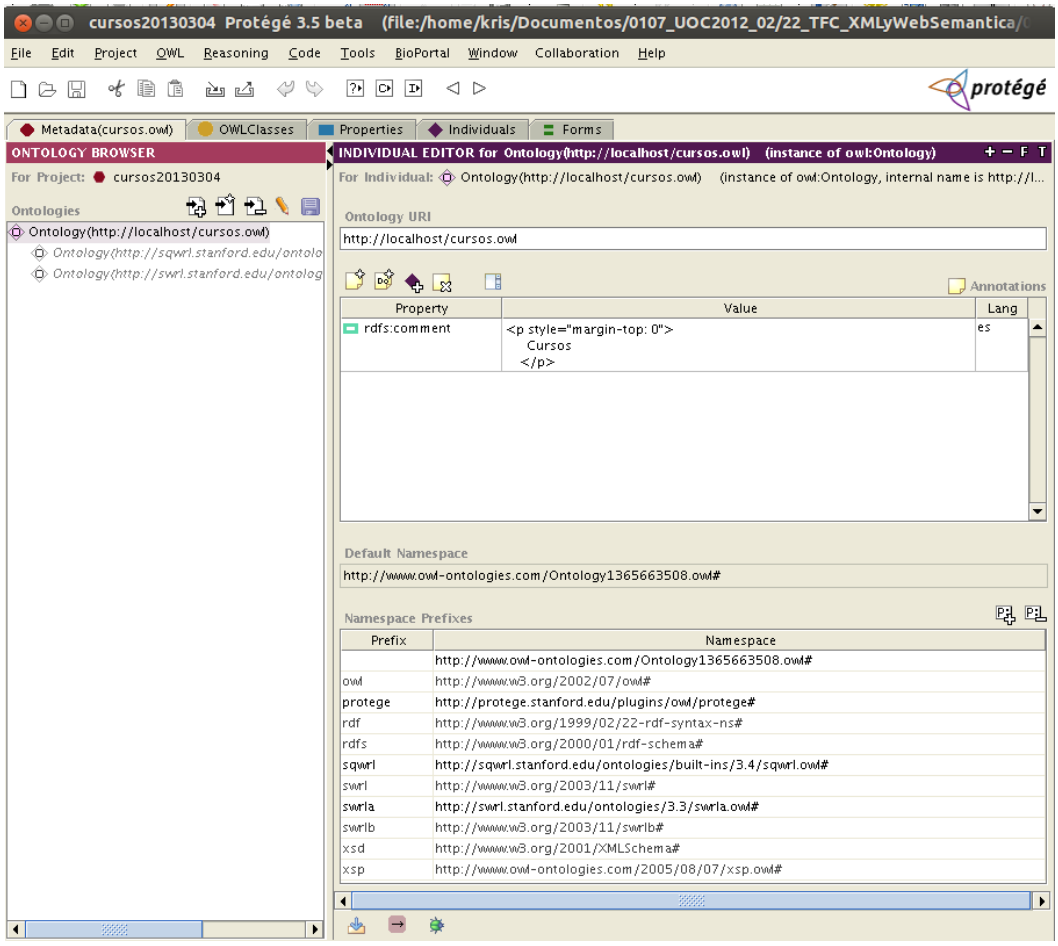


Ilustración 6: Metadatos de la Ontología Cursos

A continuación se muestran las características de la ontología cursos, y se explica como se ha ido definiendo la ontología Protégé.

33 Namespace o espacio de nombres XML es una recomendación W3C para proporcionar elementos y atributos con nombre único en una instancia XML. Una instancia XML puede contener nombres de elementos o atributos procedentes de más de un vocabulario XML. Si a cada uno de estos vocabularios se le da un espacio de nombres, un ámbito semántico propio, referenciado a una URI donde se listen los términos que incluye, se resuelve la ambigüedad existente entre elementos o atributos que se llamen igual, la homonimia. Los nombres de elementos dentro de un espacio de nombres deben ser únicos.

3.2 Ontología de cursos

La ontología que se ha creado define los cursos de formación, destinados a trabajadores de administraciones de la CAPV.

Esta ontología está basada principalmente en la clase **Curso**. En torno a ella se han creado el resto de clases.

Los cursos se pueden clasificar en **mixto**, **on-line** y **presencial**.

Además se han creado otra serie de clases y subclases que se han considerado necesarias para caracterizar los cursos.

3.2.1 Clases Principales de la ontología Cursos

La creación de clases dentro de una ontología se realiza desde la pestaña *OWLClasses.*

Se pueden definir las clases principales que serán subclases de la clase *Thing*, y diversas subclases.

Se puede indicar que las clases son disjuntas entre sí.

A continuación se muestran las clases y subclases principales de la ontología cursos tal y como se muestran en Protégé.

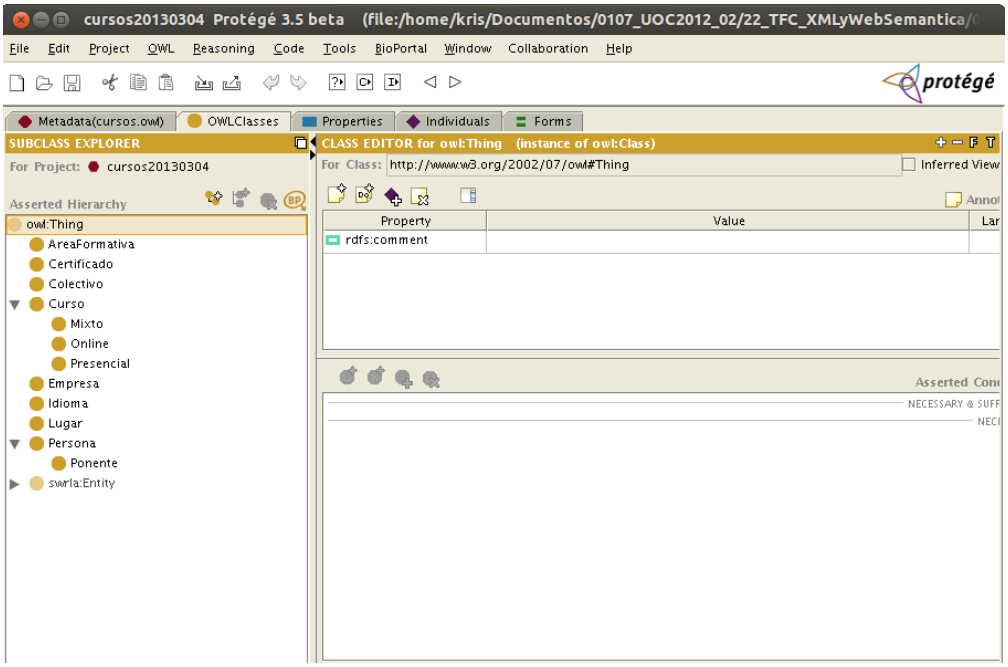


Ilustración 7: Clases de la ontología Cursos

Área Formativa. Se trata del área de conocimiento al que pertenece un curso. Un curso puede pertenecer a distintas áreas formativas.

Certificado. Se trata del tipo de certificado que se asigna a los cursos. Existen tipos de certificados específicos dependiendo del tipo de curso, por ejemplo, tenemos los certificados de Urbanismo, o los certificados de asistencia, ...

Colectivo. Define el colectivo de personas al que están dirigidos los cursos.

Curso. Define el concepto de curso como tal.

Subclases de Curso: Mixto, On-line y Presencial

Empresa Los cursos pueden ser impartidos por empresas, diseñados por empresas. A su vez, los ponentes de los cursos pueden pertenecer a alguna empresa.

Idioma Es el idioma de impartición del curso.

Persona Se refiere a una persona. Puede ser un ponente.

Subclases de Persona: Ponente

Si nos posicionamos sobre cada una de las clases se muestra las características de la misma.

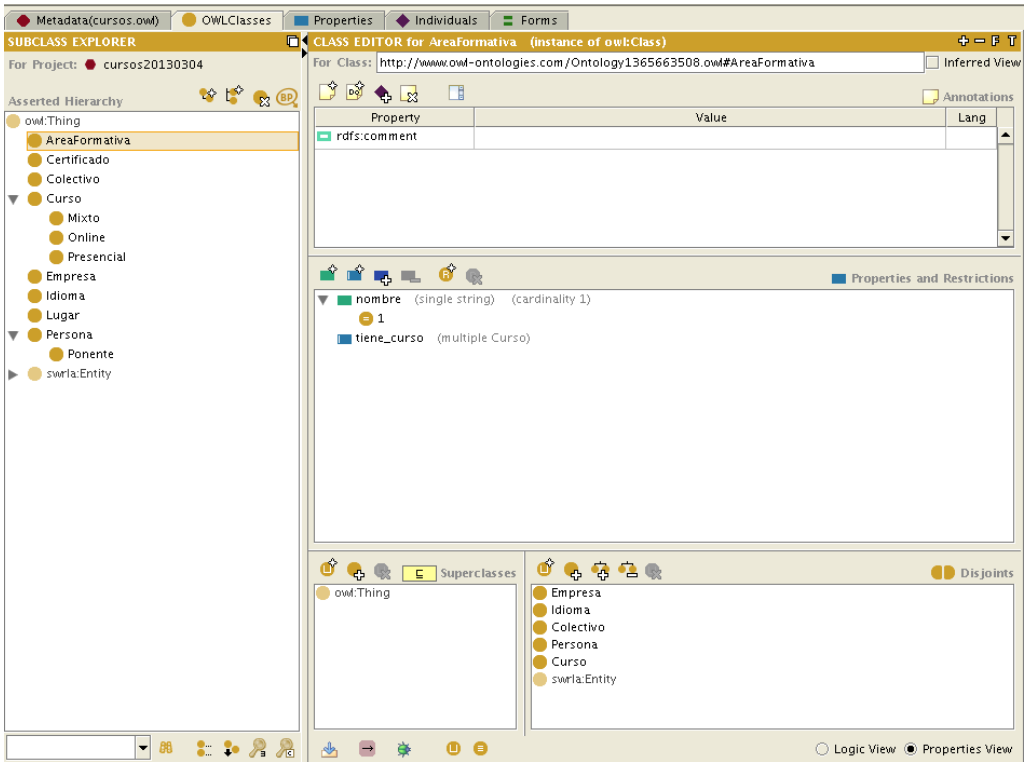


Ilustración 8: Características de la clase AreaFormativa

En el recuadro *Properties and restriccions* nos muestra dos propiedades:

- la propiedad *nombre* que es de tipo *datatype=string*, y tiene una cardinalidad obligada de 1.
- la propiedad *tiene_curso* que es de tipo *objeto=Curso* y tiene cardinalidad múltiple.

Además en el recuadro *superclasses* nos muestra si es subclase de otra. En este caso se ve que ese subclase de *Thing*.

Y en el recuadro *Disjoints*, nos muestra las clases disjuntas de *AreaFormativa*.

A continuación muestro otro ejemplo con las propiedades de la subclase *Mixto*.

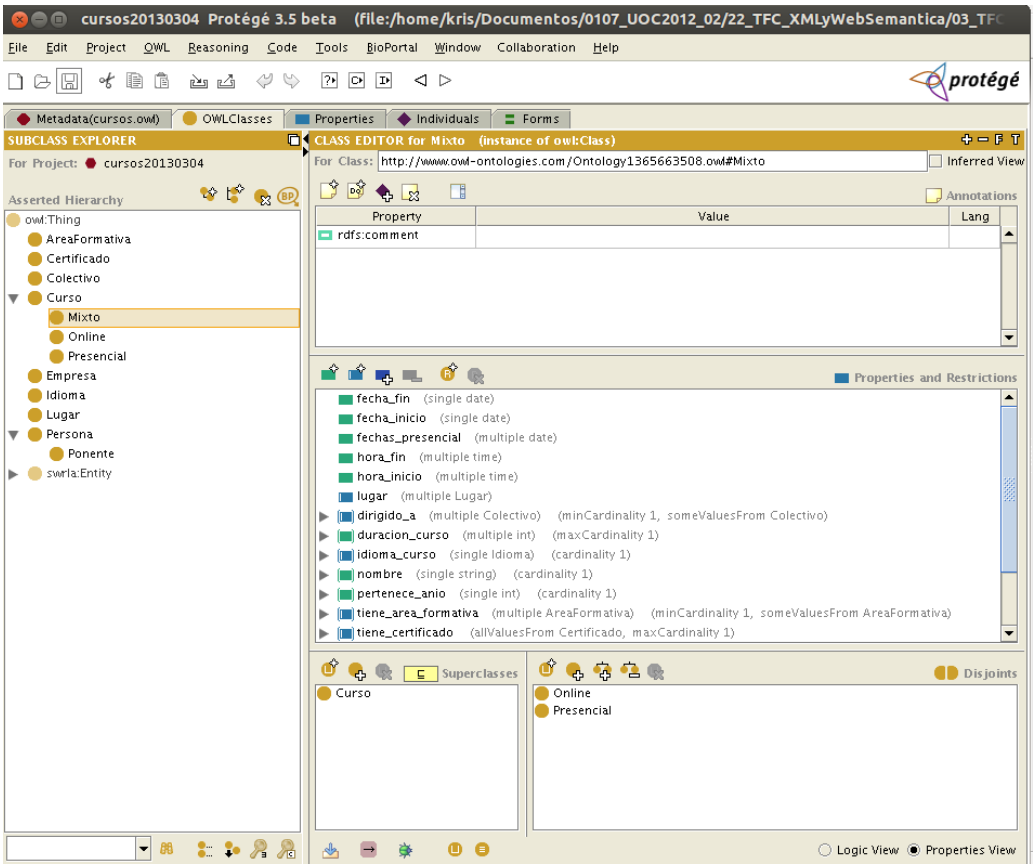


Ilustración 9: Características de la clase Mixto

Se puede apreciar que *Mixto* es una suclase de *Curso*, que es disjunta con las clases *Online* y *Presencial*, y se puede ver la lista de propiedades de la clase y sus restricciones. Hay una serie de propiedades que son heredadas de la clase curso

(aparecen marcadas entre paréntesis). El resto de propiedades, en cambio, son propias de la clase *Mixto*.

3.2.2 Propiedades de la ontología

Las propiedades que se asignan a las diferentes clases se crean desde la pestaña Properties.

Hay propiedades que son de tipo **object**. En el ejemplo vemos la propiedad *es_creador_de*.

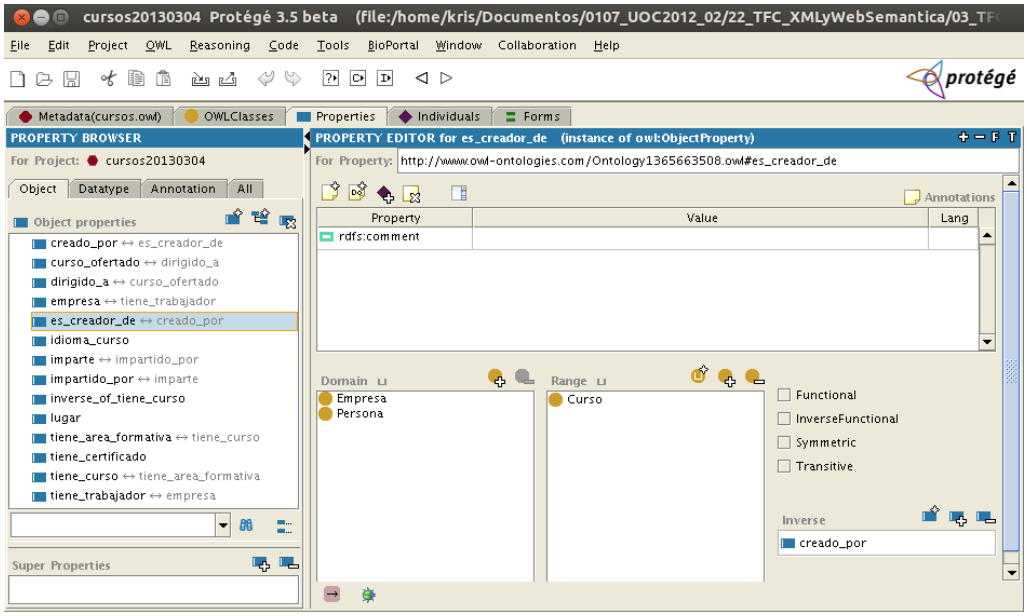


Ilustración 10: Características de la propiedad de tipo object es_creador_de

Tal y como vemos el *domain* de la propiedad corresponde a las clases *Empresa* y *Persona;* esto quiere decir que ambas clases tienen un atributo denominado *es_creador_de.*

Por otra parte, el *range* de la propiedad es la clase *Curso.* Es decir, que los valores que podrá tomar está propiedad están comprendidos en la clase *Curso.*

Esta propiedad la podríamos traducir como: "una *Empresa* y/o una *Persona* pueden crear algún *Curso"*

Asímismo vemos que está propiedad es *Inverse* con la propiedada *creado_por.*

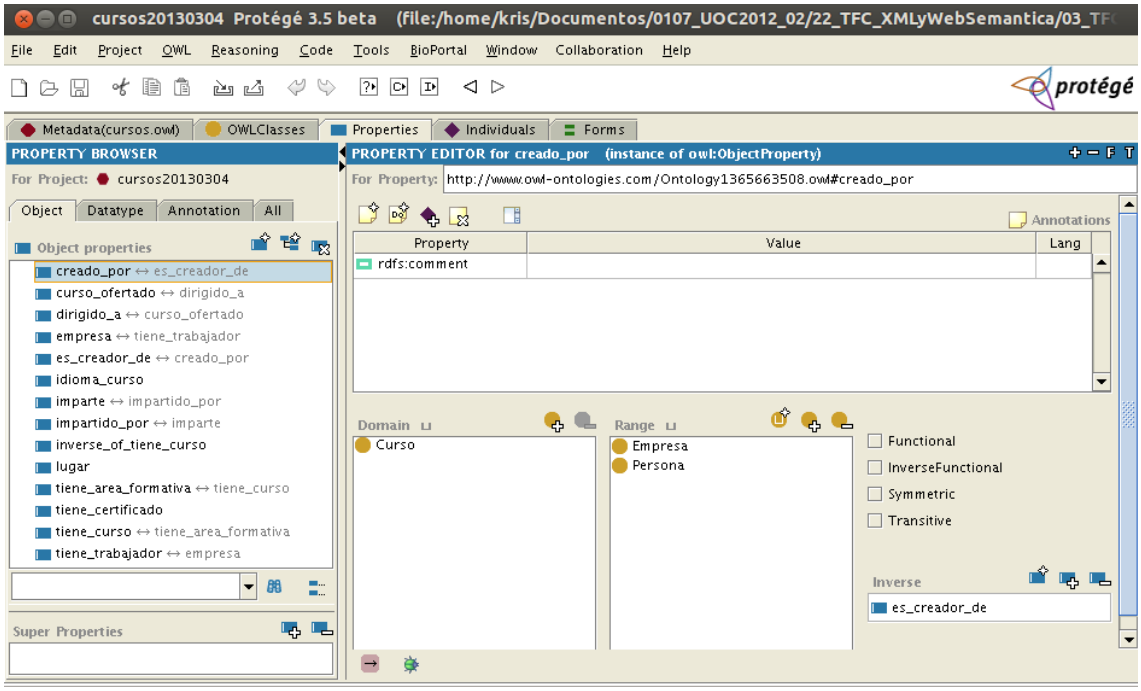


Ilustración 11: Características de la propiedad creado_por

La propiedad *creado_por* significa lo siguiente: "un *Curso* puede ser creado por una *Empresa* y/o por una *Persona.*

El hecho de que estas propiedades sean **inversas**, significa que si se crea una entidad de tipo *Curso,* y en el atributo *creado_por* se le añade una *Persona*, al consultar dicha *Persona* veremos que automáticamente se le ha rellenado el atributo *es_creador_de* con el *Curso.*

Otras propiedades que podemos crear son las de tipo **Datatype**.

En la imagen vemos la propiedad *nombre,* cuyo dominio está definido para las clases *Curso, Persona, Empresa, AreaFormativa, Colectivo* y *Lugar,* y que es de tipo *string. Además es functional.* Si una propiedad tiene la característica *functional* significa que deberá tener asignado obligatoriamente un valor. En este caso se refiere a la denominación de las instancias de las clases antes mencionadas.

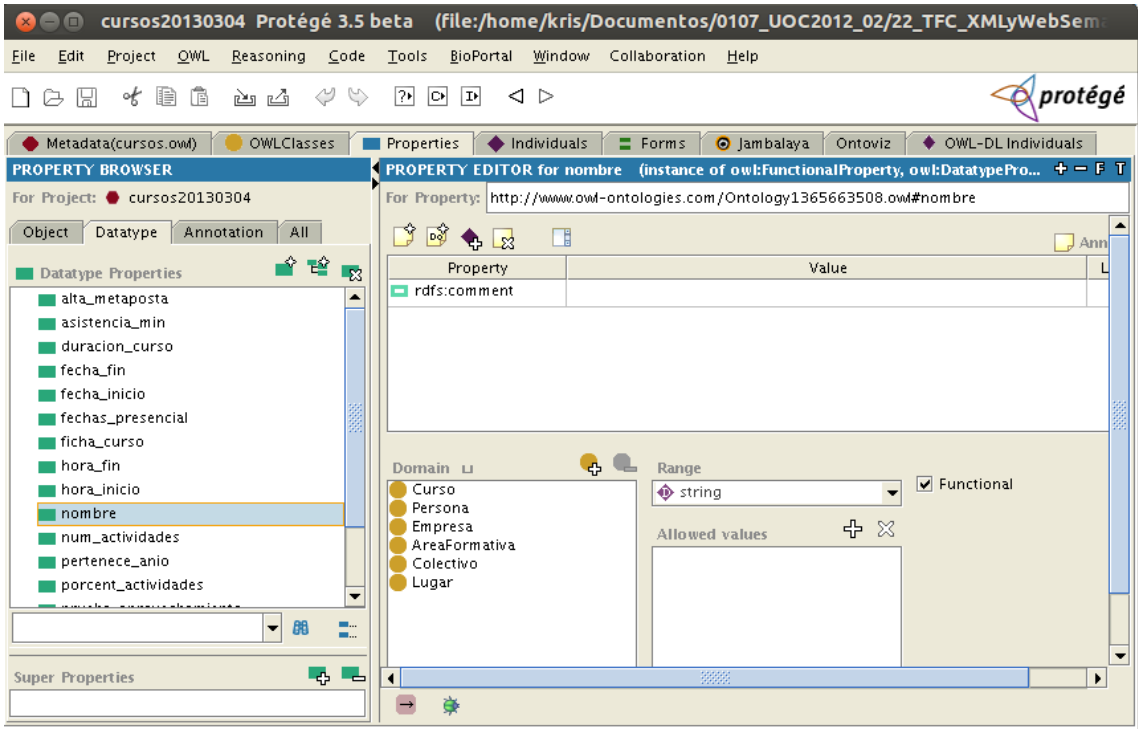


Ilustración 12: Características de la propiedad nombre

Si vemos las propiedades desde la pestaña *All* nos aparecerán juntas tanto las propiedades tipo *Object* como las propiedades *Datatype*.

En la tabla que se muestra a continuación se listan todas las propiedades que forman parte de la ontología, se muestra su descripción y su relación entre ellas.

Property	Domain	Range	Functional	Inverse	Descripción
alta_metaposta	Certificado	Boolean			Indica si requiere alta en metaposta para obtener el certificado
asistencia_min	Certificado	Int			Indica el porcentaje mínimo de asistencia para obtener el certificado
creado_por	Curso	Empresa Persona		es_creador_de	**Empresa** y/o **persona** creador de un curso.
curso_ofertado	Colectivo	Curso			**Cursos** ofertados para un colectivo.
dirigido_a	Curso	Colectivo		curso_ofertado	**Colectivo** al que está dirigido un curso.
duracion	Curso	Int			Indica la **duración** en horas de un curso

Property	Domain	Range	Functional	Inverse	Descripción
empresa	Persona	Empresa		tiene_trabajador	**Empresa** para la que trabaja una persona
es_creador_de	Empresa Persona	Curso		creado_por	**Curso** que ha sido creado por una empresa o persona
fecha_fin	Mixto On-line	Date	X		Indica la **fecha** de finalización de un curso on-line (o de la parte online de un curso mixto)
fecha_inicio	Mixto On-line	Date	X		Indica la **fecha** de inicio de un cursoon-line (o de la parte online de un curso mixto)
fechas_presencial	Mixto Presencial	Date			Indica las **fechas** de impartición de un curso. Un curso presencial se puede impartir en varios días no consecutivos.
ficha_curso	Curso	AnyURI			**URI** para la ficha del curso
hora_fin	Mixto Presencial	Time			Indica la **hora** de fin del curso presencial.
hora_inicio	Mixto Presencial	Time			Indica la **hora** de inicio del curso presencial.
idioma	Curso	Idioma	X		Indica el **idioma** de un Curso
imparte	Ponente Empresa	Curso		Impartido_por	**Curso** que imparte un Ponente y/o Empresa
impartido_por	Curso	Ponente Empresa		imparte	**Ponente** y/o **empresa** que harán de docentes para un curso
lugar	Mixto Presencial	Lugar			**Lugar** de impartición de cursos mixtos o presenciales.
nombre	Curso Persona Empresa AreaFormativa Colectivo Lugar	String	X		Nombre de curso, persona, empresa, colectivo, área formativa, lugar.
num_actividades	Certificado	Int			Indica el número mínimo de actividades para obtener el certificado
pertenece_anio	Curso	int	X		Indica el **año** al que pertenece el curso. El año

Property	Domain	Range	Functional	Inverse	Descripción
					es un int.
porcent_actividades	Certificado	Int			Indica el porcentaje mínimo de actividades para obtener el certificado
prueba_aprovechamiento	Certificado	Boolean			Indica si requiere la superación de la prueba de aprovechamiento para obtener el certificado
tiene_area_formativa	Curso	Area_formativa		tiene_curso	**Area formativa** a la que pertenece un curso
tiene_certificado	Curso	Certificado			**Tipo de Certificado** que está asignado a un curso
tiene_curso	AreaFormativa	Curso			**Curso** perteneciente a un área formativa
tiene_precio	Curso	Float	X		Indica el **precio** de un Curso
tiene_trabajador	Empresa	Persona		empresa	**Persona** que pertenece a una empresa
tipo	Certificado	String			Indica el tipo de certificado que se obtendrá tras la realización de un determinado curso.

Tabla 3: Propiedades de la ontología

3.2.3 Instancias de la ontología

Después de definir las clases, las propiedades y las restricciones de la ontología se pueden definir las distintas instancias que la forman. Para ello Protégé crea formularios por defecto, que están asociados a cada clase y que nos facilitan la misión de introducir nuevas instancias. Estos formularios pueden ser modificados desde la pestaña *Form.*

A continuación se muestra una pantalla en la que se podría modificar la ubicación de las propiedades en el formulario correspondiente a la clase Persona.

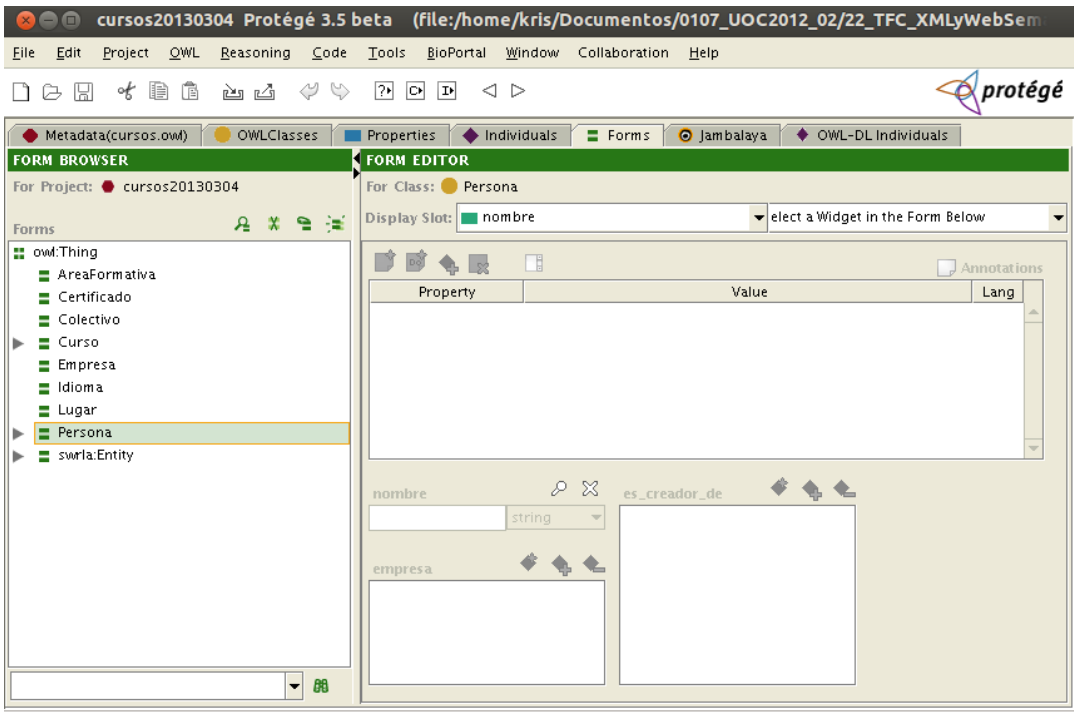


Ilustración 13: Modificación de formulario de la clase Persona

La creación de instancias se realiza desde la pestaña *Individuals*.

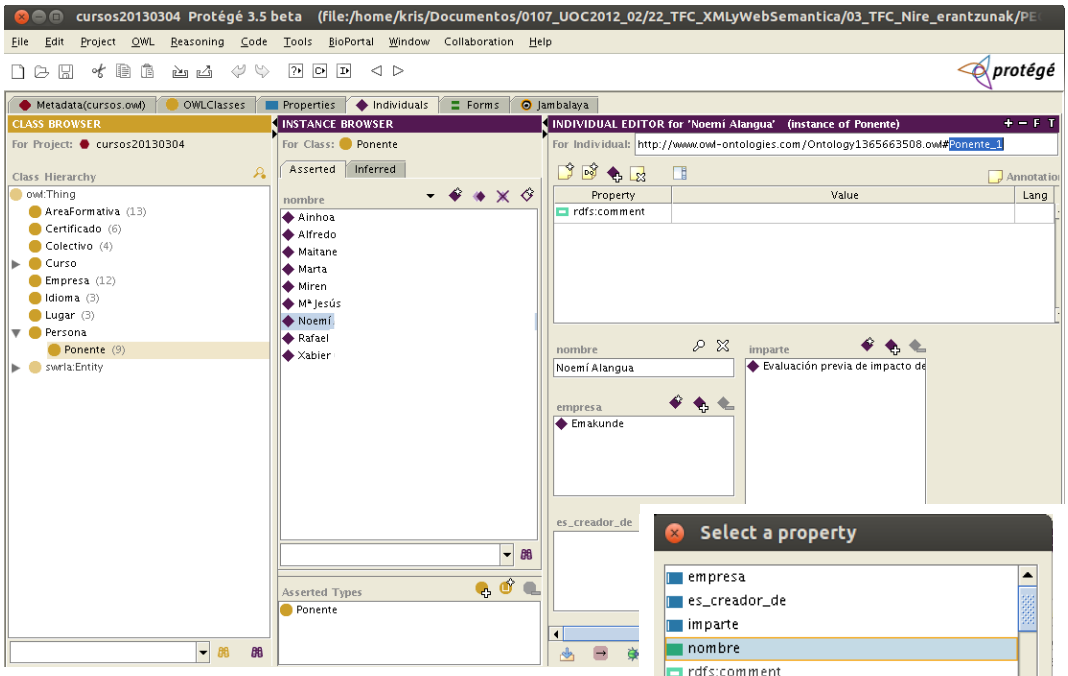


Ilustración 14: Introducción de instancias

Desde la ventana central, *Instance Browser,* podemos añadir o quitar individuos. Además nos permite elegir

Ilustración 15: Propiedad a visualizar

qué propiedad de la clase queremos que se visualice en el *Browser*, mediante la opción *set display property.*

En la ventana derecha, *Individual Editor,* podemos ir añadiendo las propiedades de las diferentes instancias.

3.2.4 Visualización gráfica de la ontología

La ontología se puede visualizar gráficamente. Para ello hay que instalar el pluggin Jambalaya y una vez instalado hay que activar la pestaña correspondiente desde *OWL -> Preferences -> Tabs -> JambalayaTab*.

A continuación se muestran los gráficos generados con Jambalaya, correspondientes a la ontología de cursos.

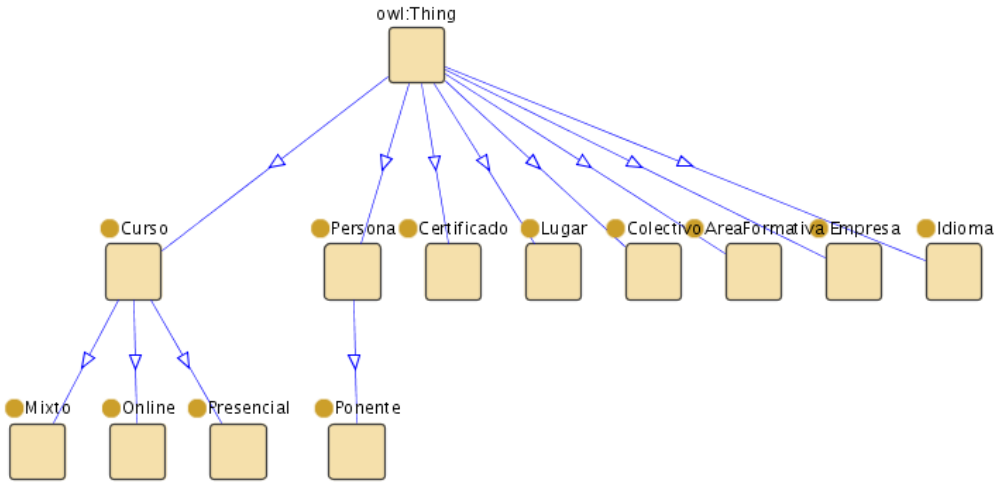


Ilustración 16: Clases y subclases de la ontología Cursos

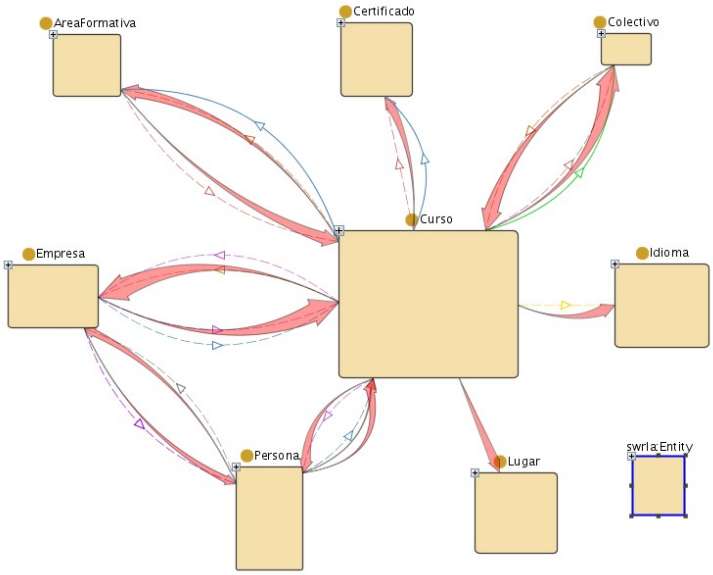


Ilustración 17: Relaciones entre clases

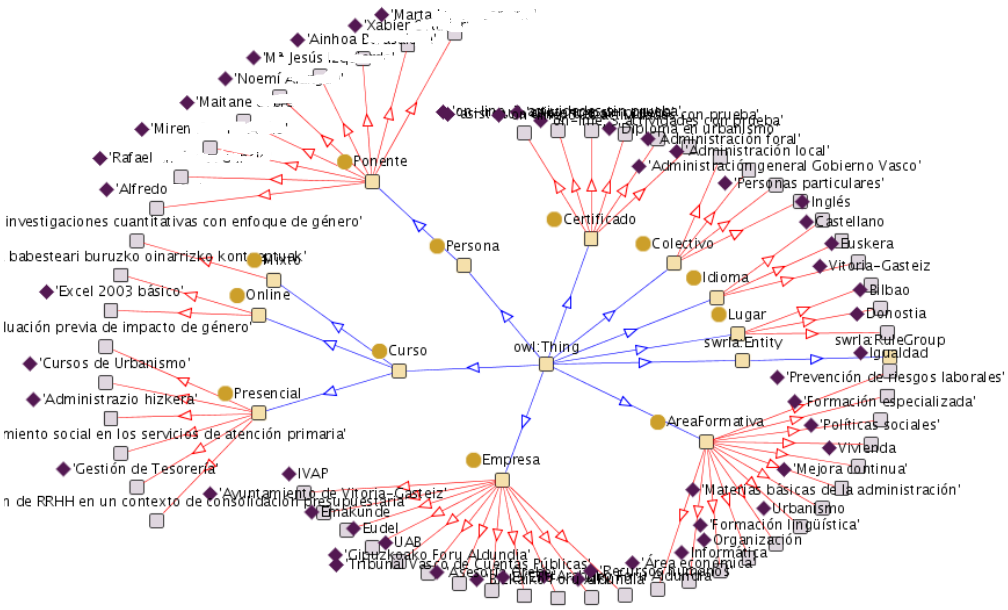


Ilustración 18: Visualización gráfica de las entidades

4 Almacenes de datos para albergar ontologías

Las poblaciones de ontologías deberían estar almacenadas en bases de datos destinadas a tal efecto. Hoy en día existen SGBD cuya orientación es semántica y por lo tanto están optimizados para el albergue de datos de este tipo. Su principal característica es la capacidad que tienen de almacenar tripletas del tipo sujeto-predicado-objeto, por esa razón se les llama bases de datos "triplestore"[34] (en ingleś). Están pensadas para el almacenamiento y la recuperación de tripletas mediante un lenguaje de consulta. Se pueden importar y exportar a formato RDF o a otros formatos basados en tripletas, y además poseen la capacidad de almacenar y manejar una cantidad enorme de información (billones de tripletas). En este tipo de bases de datos, al contrario de lo que ocurre en los SGBD relacionales, no es necesario mantener una estructura estricta de definición, y ante eventuales reformas de la configuración, o ante la definición de nuevas propiedades o relaciones, no es necesario hacer ningún cambio; basta con cargar las nuevas tripletas con las relaciones entre entidades, para que la estructura quede modificada, ya que en las tripletas se almacenan tanto la definición de los atributos y relaciones como los datos propiamente dichos. Por el contrario, en una base de datos relacional, cualquier cambio en la definición de la base

34 http://en.wikipedia.org/wiki/Triplestore

de datos suele ser muy costoso e incide negativamente en las posteriores consultas y adiciones de datos.

En los SGBD semánticos el lenguaje de consulta que ha propuesto el consorcio W3C como lenguaje estandar es SPARQL; es un lenguaje de alto rendimiento, parecido al SQL pero adaptado a la potencialidad de la web semántica y a la consulta en tripletas.

Algunas bases de datos semánticas han nacido desde el principio como tales, mientras que otras han surgido de la adaptación de bases de datos relacionales al modelo semántico basado en tripletas. Uno de los principales obstáculos radica en que la consulta de tripletas mediante SQL pierde eficicacia comparado con las consultas realizadas directamente mediante SPARQL en bases de datos nativas en las rendimiento es mucho mejor.

A continuación se adjunta una lista de bases de datos semánticas cuya característica es que ofrecen soporte completo para SPARQL (incluyendo UPDATE). Existen más bases de datos, pero aquí sólo se han contemplado las que tienen soporte SPARQL completo.

Nombre del producto	**Almacenaje nativo**	**APIs nativas**
4store	Triplestore	Sólo línes de comandos
AllegroGraph	Graph	Para los más modernos lenguajes de programación
ARC2	3rd party	PHP
BigDAta	Triplestore	Java
D2R Server	3rd party	Java
Dydra	Graph database in the cloud SaaS	REST API
Jena	Tuple store	Java
Meronymy SPARQL Database Server	Triplestore / Quadstore	Para los más modernos lenguajes de programación
OntoBroker	Triplestore	Java
Ontotext OWLIN	3rd party	ODBC, JDBC, ADO.NET, OLE DB, XMLA, HTTP, etc.
Parliament	3rd party	Java or C++
ERDF-Query	3rd party	Perl
Redland, Redstore	3rd party	C
StrixDB	Triplestore	Lua

Tabla 4: Algunos SGBD semánticos

Esta no es una lista de cerrada de bases de datos, prueba de ello es que a hace unos días, (el 15 de mayo) se publicaba el anuncio del lanzamiento de un nuevo producto destinado a almacén semántico, concretamente: MarkLogic[35].

No obstante, algunas bases de datos relacionales, pese a no estar expresamente concebidas para este fin, pueden albergar ontologías y se pueden utilizar en la web semántica.

Un ejemplo de esto es el SGBD Oracle, que está ampliamente extendido en multitud de sistemas. Las últimas versiones de Oracle proveen de utilidades para añadir administración de datos para la web semántica. Mediante la instalación de Jena[36], se puede realizar una conversión de datos a RDF y OWL ejecutando unas consultas y almacenando los datos obtenidos como tripletas RDF. Aunque como se ha comentado anteriormente, su rendimiento siempre será inferior al de los SGBD semánticos nativos.

En este trabajo se analiza el funcionamiento de dos de los sistemas de gestión de bases de datos semánticas mencionados anteriormente: **OWLIM** y **Virtuoso DB**.

5 OWLIM

OWLIM es una base de datos a escala industrial que usa estándares de la web semántica para lograr inferencia, integración y consolidación entre datos heterogéneos.

Sus características principales son:

- Tiene motor nativo RDF implementado en Java.
- Presenta un eficaz comportamiento tanto a través de Sesame como de Jena.
- Alta escalabilidad
- Soporte para interaccionar con RDFS, OWL 2 RL y OWL 2 QL.

Existen varias versiones de OWLIM. En este trabajo se utilizará la versión libre, OWLIM-Lite[37].

35 http://www.marklogic.com/

36 Jena es una infraestructura Java para la construcción de aplicaciones para web semántica.

37 http://owlim.ontotext.com/display/OWLIMv53/OWLIM-Lite

5.1 Arquitectura y características de OWLIM-Lite

OWLIM está desarrollado por la empresa ONTOTEXT que se dedica principalmente a la investigación y desarrollo de productos pensados para ser utilizados en la web semántica.

OWLIM está basado en el motor de vinculación de tripletas para razonamiento y reglas de Ontotext, TRREE., (motor nativo de vinculación de reglas RDF).

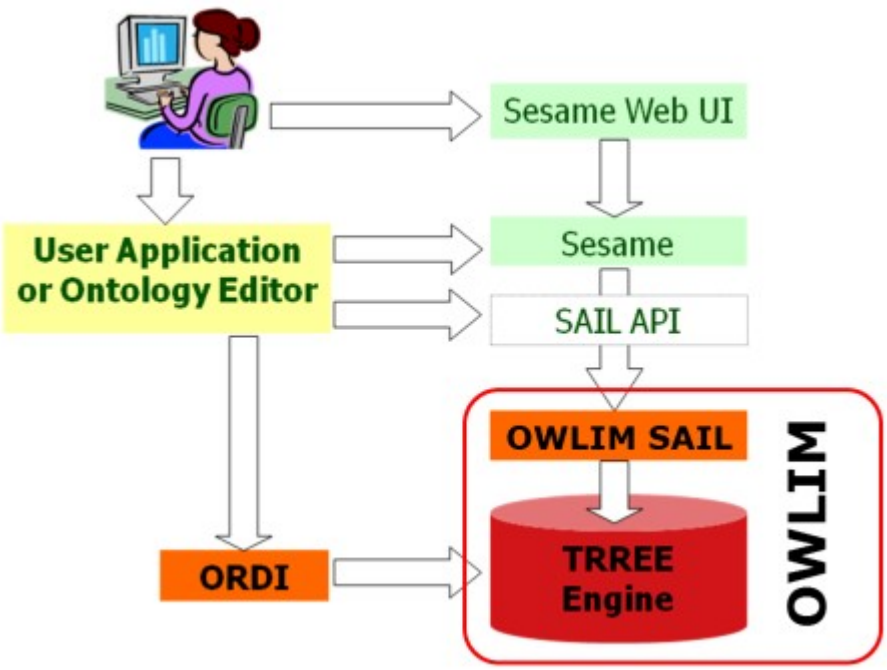


Ilustración 19: Arquitectura de OWLIM

La versión que se va utilizar es apropiada para un volumen medio de datos (por debajo de 100 millones de instancias) y para el prototipado. Esta versión es adecuada para ser utilizada como hardware de escritorio, mientras que las versiones superiores, permiten la instanciación en servidores y el manejo de mayor cantidad de información. El conjunto de reglas predefinido, combina de forma libre RDFS y OWL-Lite y permite obtener óptimo rendimiento y expresividad.

Las características principales son las siguientes:

- El razonador y el evaluador de consultas desempeñan su función completamente en la memoria principal. OWLIM es muy rápido debido a que almacena todo el contenido del repositorio en la memoria principal, lo que permite una búsqueda eficiente y una respuesta rápida a las consutas.

- Se empla una estrategia de persistencia que asegura la preservación de los datos y la consistencia.
- La carga de datos es muy rápida.
- La configuración es sencilla.

5.1.1 Modo de almacenamiento de OWLIM

Por otra parte la persistencia de OWLIM-Lite se implementa escribiendo en un fichero con formato N-Triple. El repositorio puede ser dividido en múltiples ficheros, de los cuales todos ellos excepto uno, son de sólo lectura. El fichero que se puede escribir tiene dos consideraciones; por una parte es el origen desde el cual son cargadas las tripletas, y también es el destino en el que se almacenan las nuevas instancias. Esta estrategia de backup asegura que no se produzcan pérdidas de las nuevas tripletas en caso de un fallo de corriente o un problema inesperado. Pese a ser relativamente sencilla se ha comprobado la fiabilidad de este modo de almacenamiento.

5.1.2 Limitaciones de OWLIM

Las limitaciones están relacionadas con dos aspectos: por un lado el límite del volumen de datos que se puede procesar está limitado al tamaño del memoria principal (RAM) que se esté utilizando. Por otro lado, OWLIM-Lite tiene limitaciones en cuanto a su estrategia de razonamiento cuando se producen muchos cambios en los datos y el número de instancias crece muy rápidamente, ya que se pierde eficiencia de consulta, y se degrada el rendimiento. Este problema está solventado o se da en menor medida en las versiones superiores de OWLIM.

5.2 Carga de la ontología Cursos en OWLIM[38]

Para poder cargar la ontología de Cursos, previamente tenemos que crear un repositorio en OWLIM. Para ello desde la consola workbench[39] *Repositories -> new repository*, creamos el repositorio que albergará la ontología de cursos.

38 La instalación de OWLIM se describe en el Anexo I.

39 http://localhost:8080/openrdf-workbench . Dependiendo de los casos habrá que sustituir localhost por el nombre de host en que se encuentre la instalación.

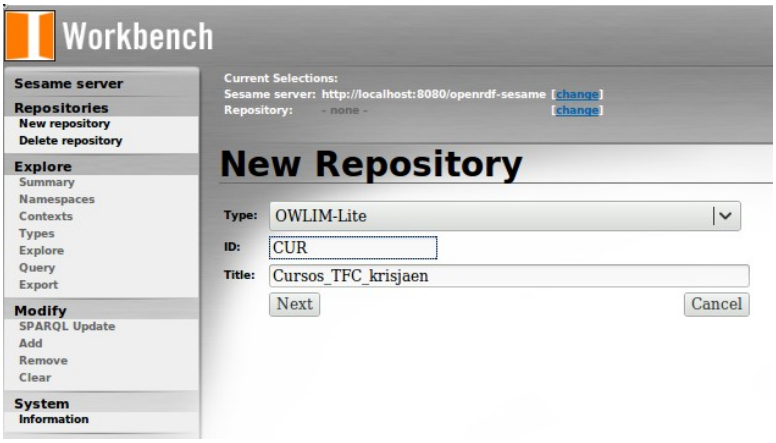


Ilustración 20: Creación del repositorio

Las características del repositorio las dejamos tal y como nos propone Sesame por defecto.

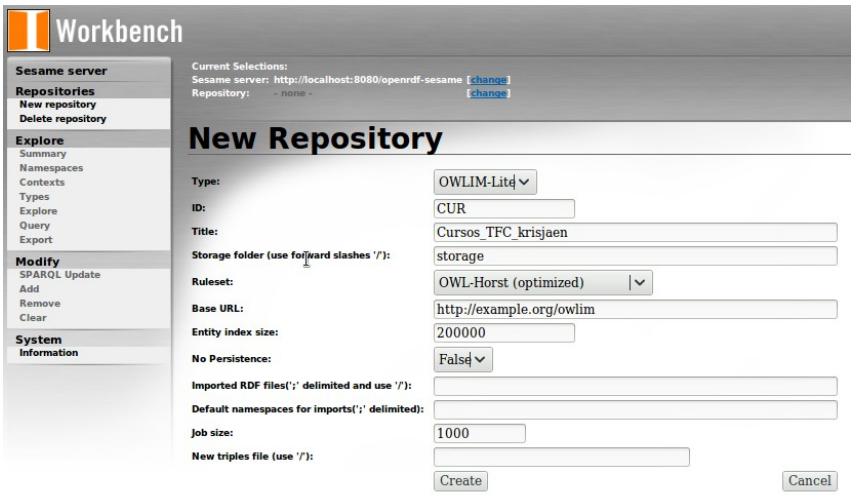


Ilustración 21: Características del repositorio

A continuación se visualiza el repositorio una vez creado.

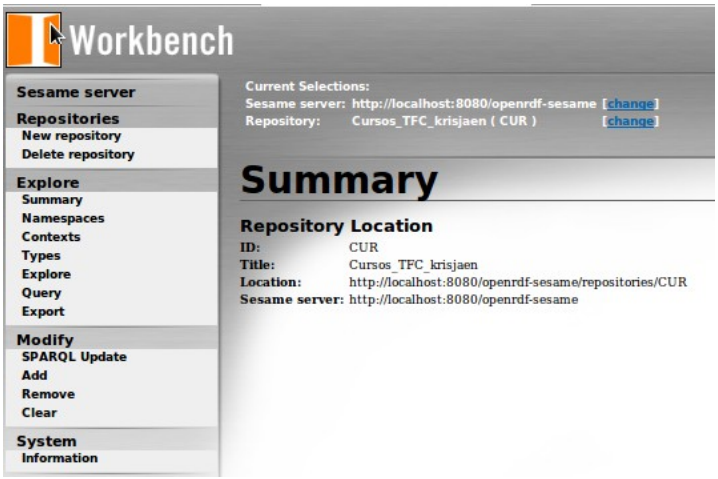


Ilustración 22: Información del repositorio creado

Dentro del workbench, desde el punto de menú *Repositories* nos muestra los dos respositorios que tenemos ahora: el repositorio inicial SYSTEM y el nuevo que hemos creado, CUR.

Ilustración 23: Lista de repositorios

A continuación vamos a realizar la carga del fichero que contiene la ontología.

Para ello, primero pincharemos en *Sesame server* para que se activen las opciones de menú. En segundo lugar, iremos al punto de menú *Modifiy -> Add.*

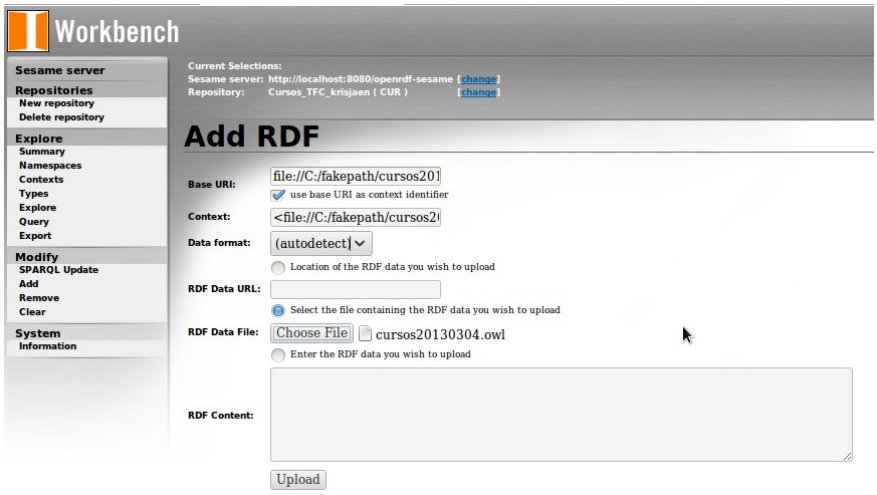


Ilustración 24: Carga de datos de la ontología

Pulsaremos en el botón Choose file y elegiremos el fichero a cargar.

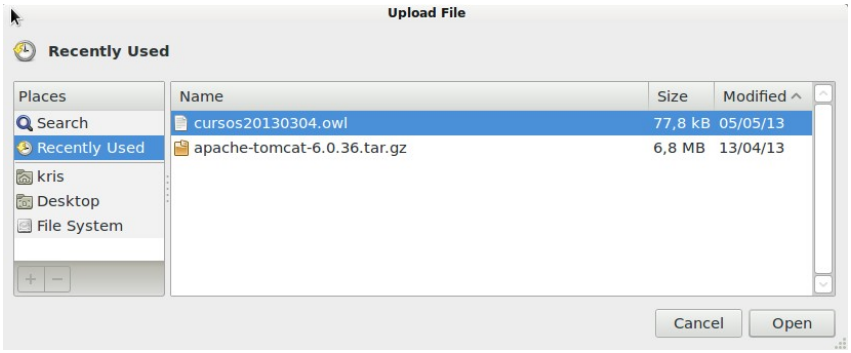


Ilustración 25: Elección del fichero a cargar

Despueś de realizada la carga, desde *Explore->Namespace* se pueden visualizar los namespaces asociados a la ontología cargada.

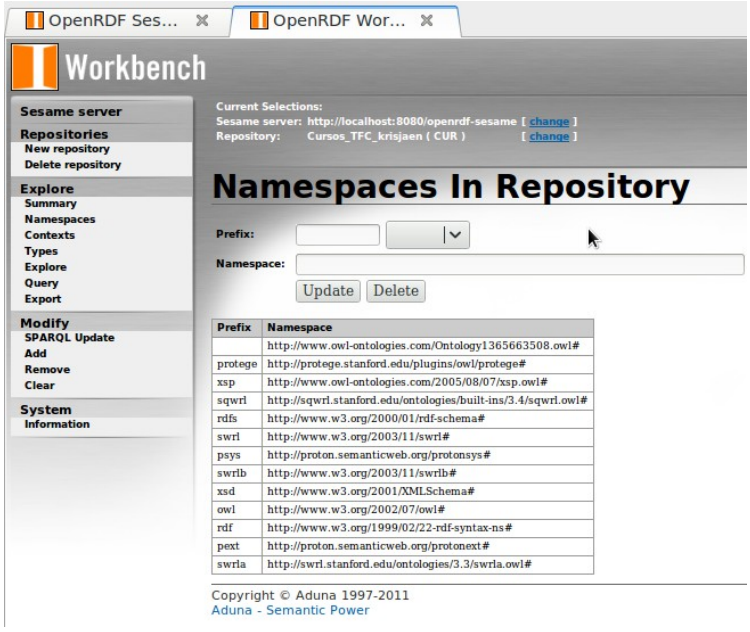


Ilustración 26: Namespaces de la ontología

A partir de este momento ya se puede pasar a realizar diferentes consultas a la ontología.

5.3 Consultas con SPARQL

Para realizar consultas vamos al punto de menú *Explore -> Query*

Tenemos seleccionado como lenguaje por defecto SPARQL y en la ventana *Query* escribimos la consulta que queremos realizar. Después pulsaremo *Execute.*

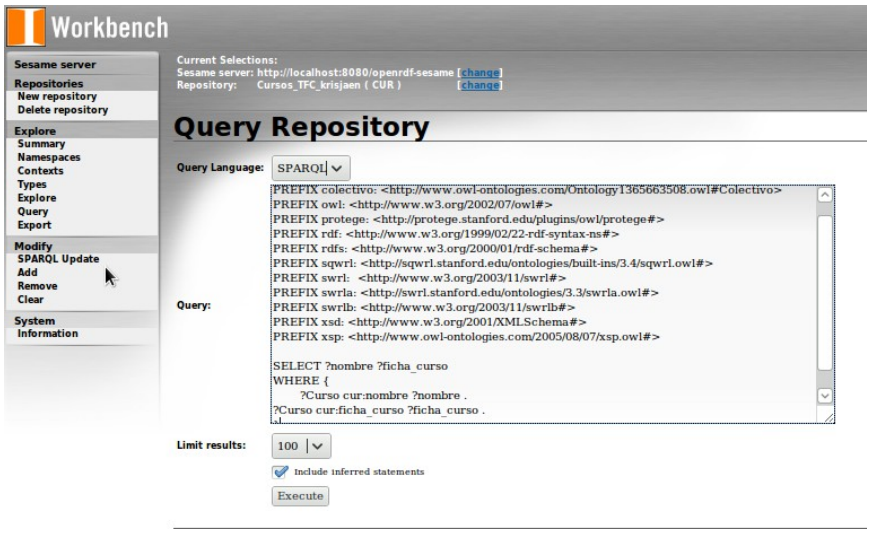


Ilustración 27: Ejemplo Consulta SPARQL

En el ejemplo se ha ejecutado una consulta muy sencilla en la que muestran los nombres de curso con sus fichas.

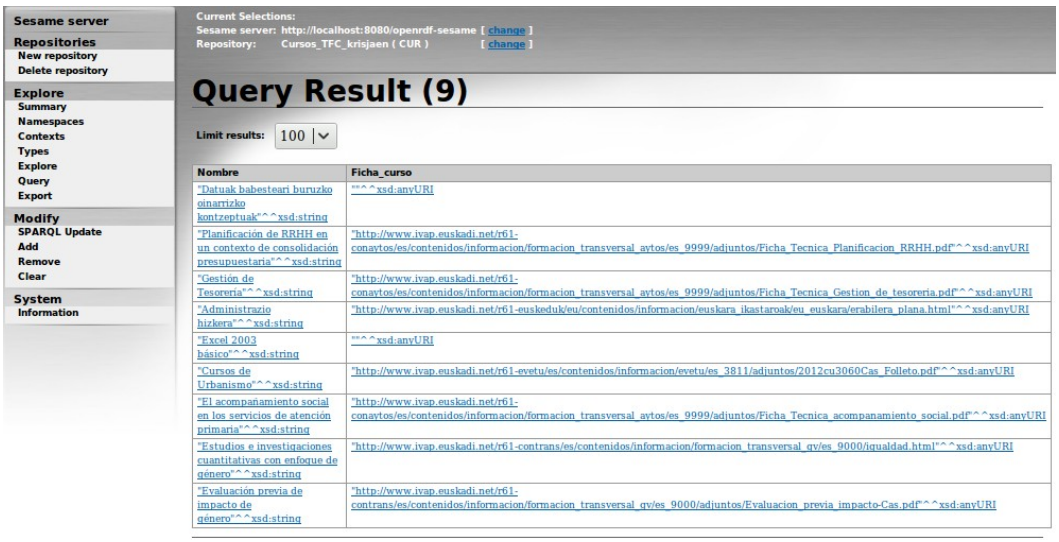


Ilustración 28: Resultado de la consulta de ejemplo

5.3.1 Consultas SELECT

Las consultas select sirven para realizar selección de datos.

A continuación se muestran una serie de consultas con sus resultados.

Consulta 1. Cursos presenciales y sus docentes, sean empresas o personas

```
PREFIX :<http://www.owl-ontologies.com/Ontology1365663508.owl#>
PREFIX protege:<http://protege.stanford.edu/plugins/owl/protege#>
PREFIX xsp:<http://www.owl-ontologies.com/2005/08/07/xsp.owl#>
PREFIX sqwrl:<http://sqwrl.stanford.edu/ontologies/built-
ins/3.4/sqwrl.owl#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX swrl:<http://www.w3.org/2003/11/swrl#>
PREFIX psys:<http://proton.semanticweb.org/protonsys#>
PREFIX swrlb:<http://www.w3.org/2003/11/swrlb#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pext:<http://proton.semanticweb.org/protonext#>
PREFIX swrla:<http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>
select ?nombre ?docente
where {?x :nombre ?nombre .
       ?x a :Presencial .
       ?x :impartido_por ?docente_aux .
       ?docente_aux :nombre ?docente .
}
```

En el siguiente pantallazo se muestra el resultado de la consulta 1.

Query Result (10)

Limit results: 200

Nombre	Docente
"Planificación de RRHH en un contexto de consolidación presupuestaria"^^xsd:string	"Rafael "^^xsd:string
"Gestión de Tesorería"^^xsd:string	"Alfredo ^^xsd:string
"Gestión de Tesorería"^^xsd:string	"Xabier "^^xsd:string
"Administrazio hizkera"^^xsd:string	"IVAP"^^xsd:string
"Cursos de Urbanismo"^^xsd:string	"EVETU"^^xsd:string
"El acompañamiento social en los servicios de atención primaria"^^xsd:string	"Miren o"^^xsd:string
"El acompañamiento social en los servicios de atención primaria"^^xsd:string	"Ainhoa "^^xsd:string
"Evaluación previa de impacto de género"^^xsd:string	"Mª Jesús "^^xsd:string
"Evaluación previa de impacto de género"^^xsd:string	"Noemí "^^xsd:string
"Evaluación previa de impacto de género"^^xsd:string	"Maitane "^^xsd:string

Ilustración 29: Resultado de la consulta 1

Para simplificar en las siguientes consultas omitiré los *prefix*. Estos serían iguales a los que aparecen en la primera consulta que se ha escrito.

Haciendo un pequeño cambio en la consulta anterior:

Consulta 2. Cursos presenciales cuyos docentes sean empresas

```
select ?nombre ?docente
where {?x :nombre ?nombre .
       ?x a :Presencial .
       ?x :impartido_por ?docent
       ?docente_aux a :Empresa .
       ?docente_aux :nombre ?docente .
}
```

El resultado de la consulta 2 en este caso es el siguient

Query Result (2)

Limit results: 200

Nombre	Docente
"Administrazio hizkera"^^xsd:string	"IVAP"^^xsd:string
"Cursos de Urbanismo"^^xsd:string	"EVETU"^^xsd:string

Ilustración 30: Resultado consulta 2

Consulta 3. Cursos dirigidos al colectivo "Administración local"

```
select ?nombre ?colectivo
where {?curx :nombre ?nombre .
       ?curx :dirigido_a ?colectivo_aux .
       ?colectivo_aux :nombre ?colectivo .
       filter regex(?colectivo, "Administración local") }
```

Y el resultado de la consulta es:

Query Result (6)

Limit results: 200

Nombre	Colectivo
"Datuak babesteari buruzko oinarrizko kontzeptuak"^^xsd:string	"Administración local"^^xsd:string
"Planificación de RRHH en un contexto de consolidación presupuestaria"^^xsd:string	"Administración local"^^xsd:string
"Gestión de Tesorería"^^xsd:string	"Administración local"^^xsd:string
"Excel 2003 básico"^^xsd:string	"Administración local"^^xsd:string
"Cursos de Urbanismo"^^xsd:string	"Administración local"^^xsd:string
"El acompañamiento social en los servicios de atención primaria"^^xsd:string	"Administración local"^^xsd:string

Ilustración 31: Resultado consulta 3

Consulta 4. Cursos presenciales creados por "Arabako Foru Aldundia" e impartidos en Bilbao.

```
select ?nombre ?creador ?lugar
where {?curx :nombre ?nombre .
       ?curx a :Presencial .
       ?curx :creado_por ?creador_aux .
       ?curx :lugar ?lugar_aux .
       ?lugar_aux :nombre ?lugar .
       ?creador_aux :nombre ?creador .
       filter regex(?creador, "Arabako Foru Aldundia")
       filter regex(?lugar, "Bilbao")
}
```

Los resultados de la consulta 4 se muestran a continuación.

Query Result (2)

Limit results: 200

Nombre	Creador	Lugar
"Planificación de RRHH en un contexto de consolidación presupuestaria"^^xsd:string	"Arabako Foru Aldundia"^^xsd:string	"Bilbao"^^xsd:string
"El acompañamiento social en los servicios de atención primaria"^^xsd:string	"Arabako Foru Aldundia"^^xsd:string	"Bilbao"^^xsd:string

Ilustración 32: Resultado consulta 4

5.3.2 consultas ASK

Las consultas ASK sirven para realizar preguntas. Devuelven TRUE o FALSE.

A continuación se muestran dos ejemplos:

Consulta 5. ¿Existe algún curso que se imparta en Euskera?

```
ask  {?curx :idioma_curso :Euskera .
}
```

El resultado a la consulta es:

Se puede realizar la comprobación con la siguiente consulta:

```
#Cursos con sus idiomas. Muestra la lista de cursos con sus idiomas
select ?nombre ?idioma_curso
where {?curx :nombre ?nombre .
       ?curx :idioma_curso ?idioma_curso .
}
```

Y el resultado es el siguiente:

Query Result (7)

Limit results: 200

Nombre	Idioma_curso	Lugar
"Planificación de RRHH en un contexto de consolidación presupuestaria"^^xsd:string	:Castellano	"Bilbao"^^xsd:string
"Gestión de Tesorería"^^xsd:string	:Castellano	"Donostia"^^xsd:string
"Administrazio hizkera"^^xsd:string	:Euskera	"Vitoria-Gasteiz"^^xsd:string
"Cursos de Urbanismo"^^xsd:string	:Castellano	"Donostia"^^xsd:string
"Cursos de Urbanismo"^^xsd:string	:Castellano	"Bilbao"^^xsd:string
"El acompañamiento social en los servicios de atención primaria"^^xsd:string	:Castellano	"Bilbao"^^xsd:string
"Evaluación previa de impacto de género"^^xsd:string	:Castellano	"Vitoria-Gasteiz"^^xsd:string

Ilustración 33: Resultado consulta 5 (ask)

Consulta 6. Pregunta si existen cursos Presenciales impartidos en Vitoria-Gasteiz en euskera

```
ask where {?curx :nombre ?nombre .
       ?curx a :Presencial .
       ?curx :idioma_curso :Euskera .
       ?curx :lugar ?lugar_aux .
       ?lugar_aux :nombre ?lugar
       filter regex(?lugar, "Vitoria-Gasteiz") }
```

La respuesta es **YES**, y la comprobación se puede realizar con la consulta siguiente:

```
#Cursos Presenciales impartidos en Vitoria-Gasteiz en euskera
select ?nombre ?lugar
where {?curx :nombre ?nombre .
       ?curx a :Presencial .
       ?curx :idioma_curso :Euskera .
       ?curx :lugar ?lugar_aux .
       ?lugar_aux :nombre ?lugar
       filter regex(?lugar, "Vitoria-Gasteiz")
}
```

Que nos muestra el siguiente resultado:

Ilustración 34: Comprobación consulta 6 (ask)

5.3.3 Consultas CONSTRUCT

Las consultas CONSTRUCT sirven para generar nuevas tripletas RDF. Este tipo de consulta retorna un grafo RDF en lugar de retornar una tabla.

En el ejemplo se ha construído un nuevo grafo que contiene los nombres de todos los cursos junto con el idioma en que están impartidos. Para construir el nuevo grafo se ha tenido que crear un nuevo PREFIX con la ruta del nuevo grafo de ejemplo.

Consulta 7. Creación de un nuevo grafo RDF mediante construct

```
PREFIX ej: <http://www.owl-ontologies.com/ejemplo.owl#>
construct {?x ej:nom ?nombre .
                    ?x ej:id ?idioma_curso .}
where {?x :nombre ?nombre .
              ?x :idioma_curso ?idioma_curso .}
```

Como resultado de la consulta nos muestra en pantalla las tripletas que ha formado y nos da opción de bajarnos el fichero con las nuevas tripletas RDF con el botón *Download*.

El fichero resultado de la consulta CONSTRUCT se puede consultar en el Anexo III.

A continuación se muestra la pantalla con las nuevas tripletas. Se observa que aparecen dos tripletas por cada curso, una para definir el idioma y otra en la que se define el nombre del curso.

Current Selections:
Sesame server: http://localhost:8080/openrdf-sesame [change]
Repository: Cursos_TFC_krisjaen (CUR) [change]

Query Result (18)

Download format: RDF/XML Download

Limit results: 200

Subject	Predicate	Object
:Online_37	<http://www.owl-ontologies.com/ejemplo.owl#nom>	"Datuak babesteari buruzko oinarrizko kontzeptuak"^^xsd:string
:Online_37	<http://www.owl-ontologies.com/ejemplo.owl#id>	:Euskera
:Presencial_7	<http://www.owl-ontologies.com/ejemplo.owl#nom>	"Planificación de RRHH en un contexto de consolidación presupuestaria"^^xsd:string
:Presencial_7	<http://www.owl-ontologies.com/ejemplo.owl#id>	:Castellano
:Presencial_22	<http://www.owl-ontologies.com/ejemplo.owl#nom>	"Gestión de Tesorería"^^xsd:string
:Presencial_22	<http://www.owl-ontologies.com/ejemplo.owl#id>	:Castellano
:10	<http://www.owl-ontologies.com/ejemplo.owl#nom>	"Administrazio hizkera"^^xsd:string
:10	<http://www.owl-ontologies.com/ejemplo.owl#id>	:Euskera
:Online_26	<http://www.owl-ontologies.com/ejemplo.owl#nom>	"Excel 2003 básico"^^xsd:string
:Online_26	<http://www.owl-ontologies.com/ejemplo.owl#id>	:Castellano
:Presencial_34	<http://www.owl-ontologies.com/ejemplo.owl#nom>	"Cursos de Urbanismo"^^xsd:string
:Presencial_34	<http://www.owl-ontologies.com/ejemplo.owl#id>	:Castellano

Ilustración 35: Resultado consulta 7 (construct)

5.3.4 Consultas DESCRIBE

Las consultas DESCRIBE devuelven un grafo RDF en el que describen los recursos encontrados. Estas consultas se construyen básicamente como las consultas SELECT. En el caso del ejemplo le pedimos que nos describa todos los recursos que tengan asociado el atributo nombre o el atributo idioma.

Consulta 8. Grafo RDF con la descripción de los recursos de la consulta

```
#Describe Cursos
describe ?nombre ?idioma_curso
where {?curx :nombre ?nombre .
       ?curx :idioma_curso ?idioma_curso .
}
```

Query Result (66)

Download format: RDF/XML | Download

Limit results: 200

Subject	Predicate	Object
:Online_37	:nombre	"Datuak babesteari buruzko oinarrizko kontzeptuak"^^xsd:string
:Presencial_7	:nombre	"Planificación de RRHH en un contexto de consolidación presupuestaria"^^xsd:string
:Presencial_22	:nombre	"Gestión de Tesorería"^^xsd:string
:10	:nombre	"Administrazio hizkera"^^xsd:string
:Online_26	:nombre	"Excel 2003 básico"^^xsd:string
:Presencial_34	:nombre	"Cursos de Urbanismo"^^xsd:string
:Presencial_31	:nombre	"El acompañamiento social en los servicios de atención primaria"^^xsd:string
:Mixto_25	:nombre	"Estudios e investigaciones cuantitativas con enfoque de género"^^xsd:string
:Presencial_52	:nombre	"Evaluación previa de impacto de género"^^xsd:string
:Euskera	rdf:type	:Idioma
:Castellano	rdf:type	:Idioma
:Euskera	rdf:type	:Idioma
:Castellano	rdf:type	:Idioma
:10	:idioma_curso	:Euskera
:Online_37	:idioma_curso	:Euskera
:Presencial_52	:idioma_curso	:Castellano
:Mixto_25	:idioma_curso	:Castellano
:Presencial_31	:idioma_curso	:Castellano
:Presencial_34	:idioma_curso	:Castellano
:Online_26	:idioma_curso	:Castellano
:Presencial_22	:idioma_curso	:Castellano

Ilustración 36: Resultado consulta 8 (describe)

5.3.5 Diferencias entre CONSTRUCT y DESCRIBE

Como se ha visto en el apartado CONSTRUCT, esta consulta nos devuelve las tripletas que le digamos. Nos permite, o extraer las tripletas sin cambiarlas, o extraer y añadir algunas tripletas transformadas. En el ejemplo anterior se han extraído las tripletas sin realizar ninguna transformación.

En cambio la consulta DESCRIBE pregunta por las tripletas que describen un recurso en particular y es el propio proceso de consulta el que decide qué tripletas y qué descripciones del recurso va a mostrar. En nuestro caso le hemos preguntado por el nombre y el idioma de los cursos y nos ha devuelto 66 filas, que son todas las veces en que aparece el atributo nombre (aparece en muchas clases) o el atributo idioma.

6 VIRTUOSO DB

Características generales de VIRTUOSO DB

- Es una base de datos objeto-relacional SQL de alto rendimiento.
- Tiene un compilador, un lenguaje de almacenamiento. Soporta SQL-99 y tiene interfaces con ODBC, JDBC, ADO, .NET y OLE/DB.
- Está capacitado para servir dinámicamente páginas que han sido escritas en el lenguaje web de virtuoso, en PHP, ASP, .net y otros.
- Soporta SPARQL embebido en SQL para poder hacer consultas sobre datos RDF.

La versión que se ha utilizado para el presente trabajo es Virtuoso Open-Source Edition.

6.1 Características de Virtuoso

6.1.1 Arquitectura

Virtuoso Universal Server es un motor de base de datos híbrido que combina la funcionalidad de las bases de datos tradicionales RDBMS, ORDBMS, las bases de datos virtuales, RDF, XML, texto libre, servidor de aplicaciones web y funcionalidades de servidor de archivos en un único sistema. Más que tener servidores dedicados para cada una de las funcionalidades mencionadas, Virtuoso es en realidad un “servidor universal” que habilita un sólo proceso multihilo, que implementa a su vez múltiples protocolos.

Virtuoso está diseñado para sacar provecho del soporte de hilos y de múltiples CPU's del sistema operativo. Consiste en un único proceso con un conjunto de hilos compartidos entre los clientes. Los múltiples hilos pueden trabajar en un único índice tipo árbol, y apenas interfieren entre ellos. Existe una caché que comparte páginas de la base de datos entre los diferentes hilos. Mientras tanto, las páginas usadas de la caché, se van escribiendo a disco mediante un proceso en segundo plano.

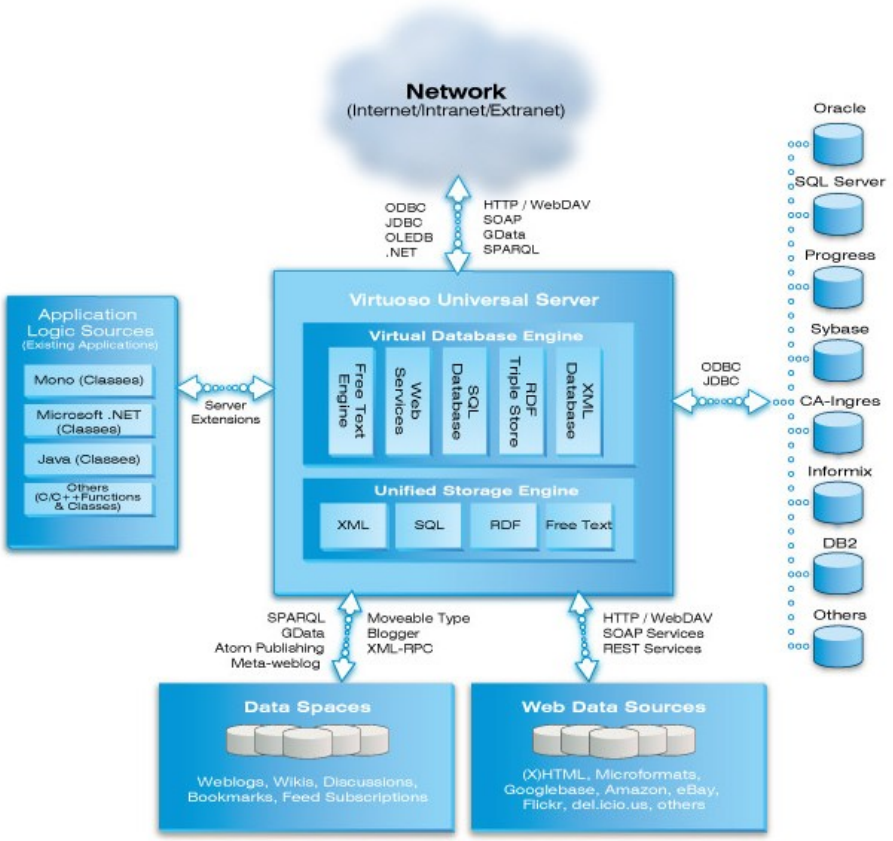


Ilustración 37: Arquitectura de Virtuoso

6.1.2 Almacenamiento físico

Virtuoso proporciona un modelo objeto-relacional extendido, que combina la flexibilidad del acceso relacional con el poder de la herencia, introducción de datos en tiempo de ejecución y acceso basado en la identidad.

El servidor de base de datos incluye un archivo físico. Con éste interacciona la memoria y los procesos de sistema operativo. Existe un proceso principal que tiene otros subprocesos a la escucha en puertos específicos para HTTP, SOAP y otros protocolos.

Para organizar la información, Virtuoso utiliza el almacenamiento genérico. Se basa en una tabla con cuatro columnas: sujeto, predicado, objeto y grafo.

Las tablas son almacenadas en un conjunto específico de archivos. Otro conjunto de archivos, separado del anterior se utiliza para almacenar los datos temporales. El máximo tamaño de cada conjunto de archivos es de 32 terabytes, para 4G x 8K páginas.

El sistema de backup que utiliza es parecido al utilizado en las bases de datos relacionales. Existe un punto de checkpoint, en el que se puede realizar un backup limpio de la base de datos, y a partir de ahí se puede guardar un fichero log con las transacciones que se han realizado.

Soporta los cuatro niveles de aislamiento definidos para las base de datos relacionales. Éstos pueden ser especificados operación por operación en cada transacción.

Virtuoso puede actuar también como gestor de recursos o coordinador de transacciones.

6.2 Carga de ontología Cursos en Virtuoso[40]

Primero accedemos a la versión web de Virtuoso[41].

Desde la pestaña *Database->Interactive SQL* le indicamos el nombre de la ontología que queremos cargar.

```
sparql load <http://www.owl-ontologies.com/Ontology1365663508.owl#>
```

A continuación nos devuelve el resultado de la carga de la ontología.

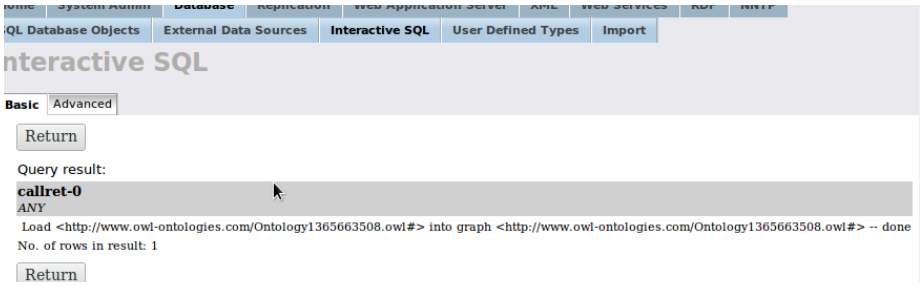


Ilustración 38: Resultado de la carga de la ontología

Cargar ontología desde RDF -> RDF Store Upload, subimos la ontología de cursos.

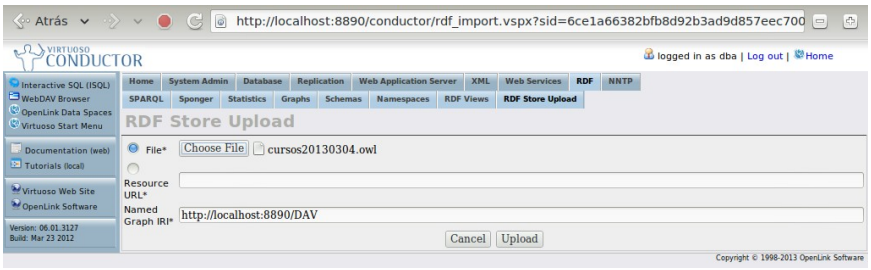


Ilustración 39: Pantalla de carga de la ontología

40 La instalación de Virtuoso se puede consultar en el Anexo II

41 http://localhost:8890/conductor Host y puerto puede cambiar dependiendo de la instalación.

Se comprueba que se visualizan las tripletas de la ontología para ello ejecutamos

```
SPARQL
DEFINE get:soft "replace"
select distinct
from <http://www.owl-ontologies.com/Ontology1365663508.owl#>
where { ?s ?p ?o}
```

Y nos muestra las siguientes pantallas de resultados

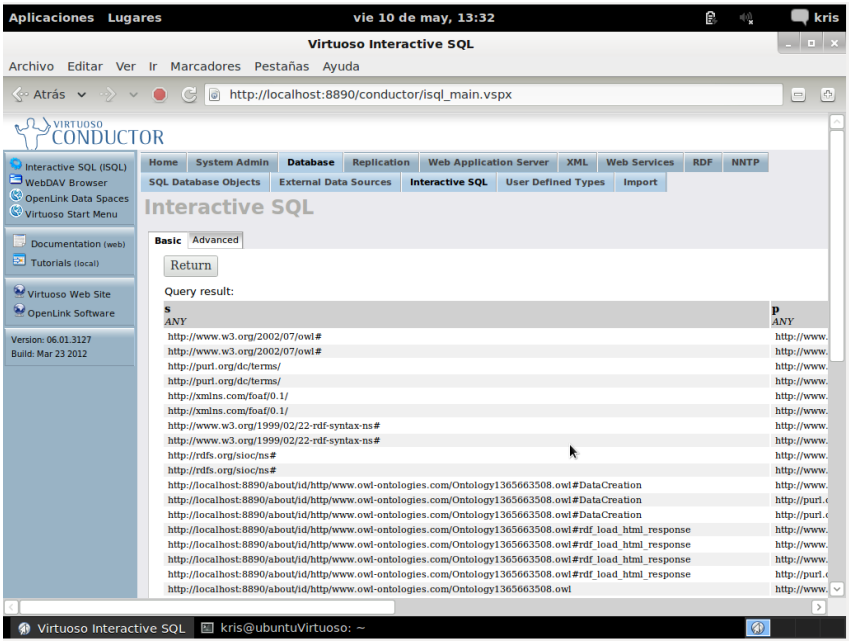


Ilustración 40: Resultado tripletas 1

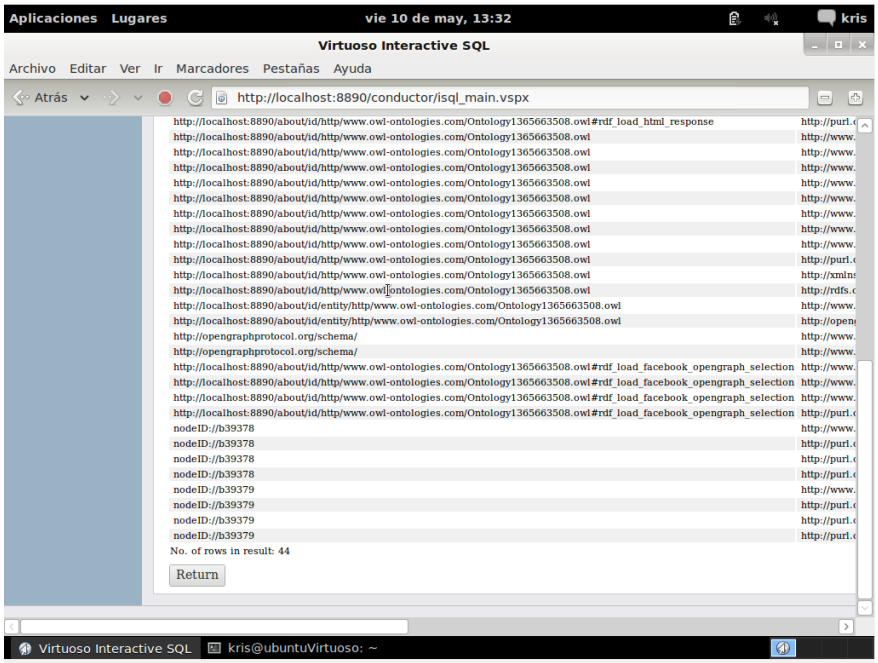


Ilustración 41: Resultado tripletas 2

6.3 Consultas con SPARQL

Las consultas SPARQL usando Conductor se pueden realizar de dos maneras:

- Desde RDF -SPARQL. Desde este punto se ejecutan como consultas SPARQL puras y los resultados se muestran en la misma pantalla a continuación de la consulta.

La consulta realizada nos devuelve los cursos con los docentes que los imparten.

SPARQL Execution

Query | Saved Queries

Default Graph IRI

Query

```
PREFIX swrl:<http://www.w3.org/2003/11/swrl#>
PREFIX psys:<http://proton.semanticweb.org/protonsys#>
PREFIX swrlb:<http://www.w3.org/2003/11/swrlb#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pext:<http://proton.semanticweb.org/protonext#>
PREFIX swrla:<http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>
select ?nombre ?docente
where {?x :nombre ?nombre .
      ?x a :Presencial .
      ?x :impartido_por ?docente_aux .
      ?docente_aux :nombre ?docente .

}
```

Execute | Save | Load | Clear

nombre	docente
"El acompañamiento social en los servicios de atención primaria"^^<http://www.w3.org/2001/XMLSchema#string>	"Ainhoa [redacted]"^^<http://www.w3.org/2001/XMLSchema#string>
"El acompañamiento social en los servicios de atención primaria"^^<http://www.w3.org/2001/XMLSchema#string>	"Miren Edurne [redacted]"^^<http://www.w3.org/2001/XMLSchema#string>
"Cursos de Urbanismo"^^<http://www.w3.org/2001/XMLSchema#string>	"EVETU"^^<http://www.w3.org/2001/XMLSchema#string>
"Administrazio hizkera"^^<http://www.w3.org/2001/XMLSchema#string>	"IVAP"^^<http://www.w3.org/2001/XMLSchema#string>
"Gestión de Tesorería"^^<http://www.w3.org/2001/XMLSchema#string>	"Xabier [redacted]"^^<http://www.w3.org/2001/XMLSchema#string>
"Gestión de Tesorería"^^<http://www.w3.org/2001/XMLSchema#string>	"Alfredo [redacted]"^^<http://www.w3.org/2001/XMLSchema#string>
"Planificación de RRHH en un contexto de consolidación presupuestaria"^^<http://www.w3.org/2001/XMLSchema#string>	"Rafael [redacted]"^^<http://www.w3.org/2001/XMLSchema#string>
"Evaluación previa de impacto de género"^^<http://www.w3.org/2001/XMLSchema#string>	"Maitane [redacted]"^^<http://www.w3.org/2001/XMLSchema#string>
"Evaluación previa de impacto de género"^^<http://www.w3.org/2001/XMLSchema#string>	"Noemí [redacted]"^^<http://www.w3.org/2001/XMLSchema#string>
"Evaluación previa de impacto de género"^^<http://www.w3.org/2001/XMLSchema#string>	"Mª Jesús [redacted]"^^<http://www.w3.org/2001/XMLSchema#string>

Ilustración 42: Resultado de consulta en SPARQL

- Desde Database - Interative SQL. Este punto es en realidad para la ejecución de consultas SQL, por lo que habrá que añadir a la consulta una línea al inicio con la claúsula SPARQL y, un punto y coma al final. Al ejecutar la consulta desde este punto, nos da el mismo número de filas pero el resultado es mucho más compacto por que se omiten las URIs de cada elemento.

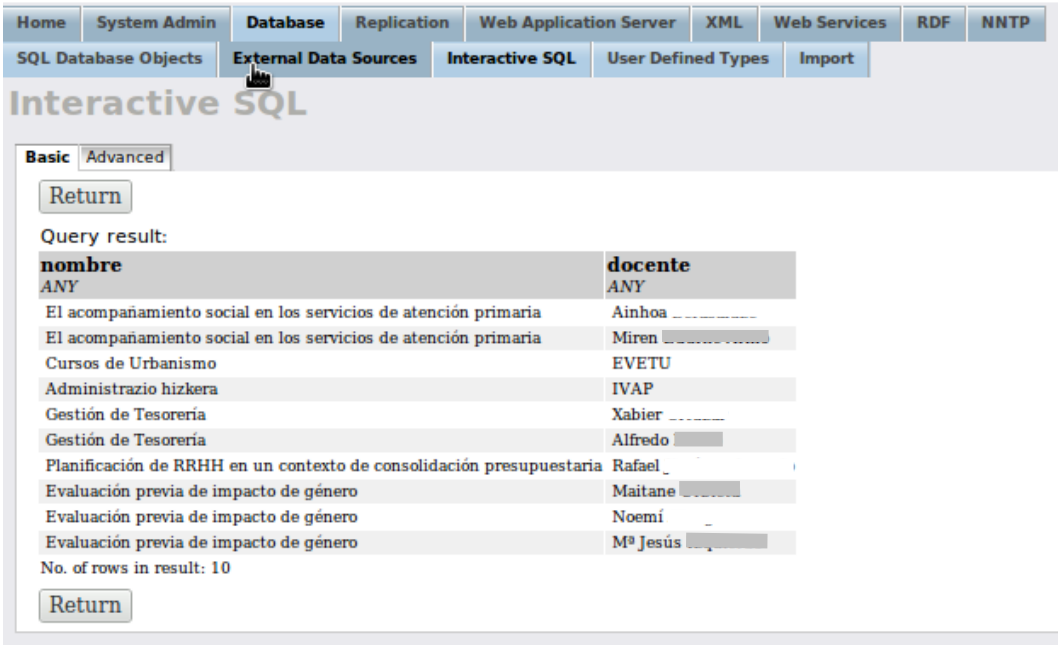


Ilustración 43: Resultado de la consulta con Interactive SQL

6.3.1 Consultas SELECT

A continuación se van a ejecutar las mismas consultas que se ejecutaron en OWLIM.

Consulta 1. Cursos presenciales y sus docentes, sean empresas o personas

```
PREFIX :<http://www.owl-ontologies.com/Ontology1365663508.owl#>
PREFIX protege:<http://protege.stanford.edu/plugins/owl/protege#>
PREFIX xsp:<http://www.owl-ontologies.com/2005/08/07/xsp.owl#>
PREFIX            sqwrl:<http://sqwrl.stanford.edu/ontologies/built-
ins/3.4/sqwrl.owl#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX swrl:<http://www.w3.org/2003/11/swrl#>
PREFIX psys:<http://proton.semanticweb.org/protonsys#>
PREFIX swrlb:<http://www.w3.org/2003/11/swrlb#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pext:<http://proton.semanticweb.org/protonext#>
PREFIX swrla:<http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>
select ?nombre ?docente
where {?x :nombre ?nombre .
       ?x a :Presencial .
       ?x :impartido_por ?docente_aux .
       ?docente_aux :nombre ?docente . }
```

En la siguiente pantalla se muestra el resultado de la consulta 1

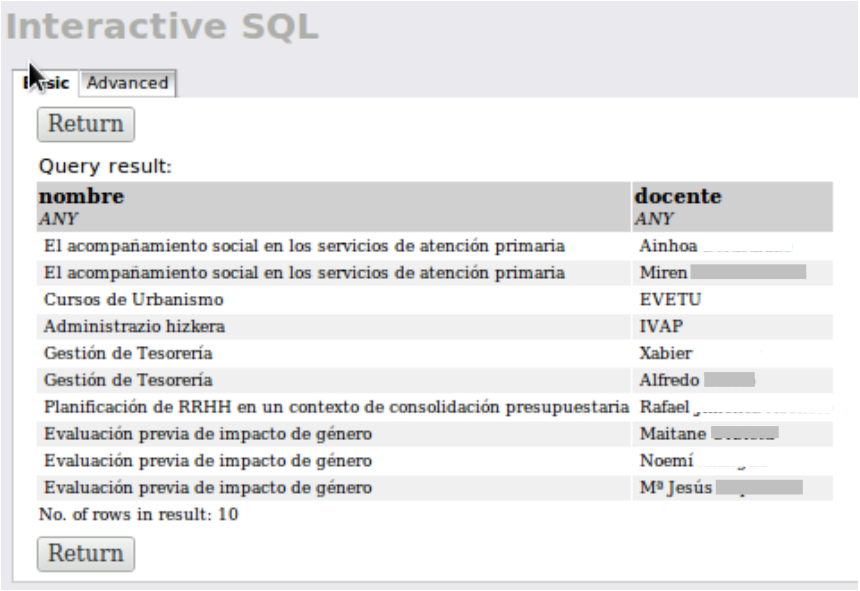


Ilustración 44: Resultado consulta 1

Consulta 2. Cursos presenciales cuyos docentes sean empresas

```
select ?nombre ?docente
where {?x :nombre ?nombre .
       ?x a :Presencial .
       ?x :impartido_por ?docente_aux .
       ?docente_aux a :Empresa .
       ?docente_aux :nombre ?docente . }
```

El resultado de la consulta 2 en este caso es el siguiente:

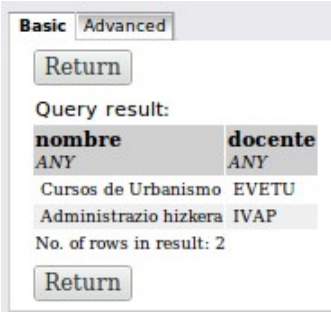


Ilustración 45: Consulta 2

Consulta 3. Cursos dirigidos al colectivo "Personas particulares"

```
#PREFIX (se omiten)
select ?nombre ?colectivo
where {?curx :nombre ?nombre .
       ?curx :dirigido_a ?colectivo_aux .
       ?colectivo_aux :nombre ?colectivo .
       filter regex(?colectivo, "Personas particulares")}
```

Y el resultado de la consulta es:

nombre	colectivo
"Cursos de Urbanismo"^^<http://www.w3.org/2001/XMLSchema#string>	"Personas particulares"^^<http://www.w3.org/2001/XMLSchema#string>

Ilustración 46: Resultado consulta 3 (select)

Consulta 4. Cursos presenciales, creados por "Arabako Foru Aldundia" e impartidos en Bilbao.

```
select ?nombre ?creador ?lugar
where {?curx :nombre ?nombre .
       ?curx a :Presencial .
       ?curx :creado_por ?creador_aux .
       ?curx :lugar ?lugar_aux .
       ?lugar_aux :nombre ?lugar .
       ?creador_aux :nombre ?creador .
       filter regex(?creador, "Arabako Foru Aldundia")
       filter regex(?lugar, "Bilbao") }
```

Los resultados de la consulta 4 se muestran a continuación.

nombre	creador	lugar
"El acompañamiento social en los servicios de atención primaria"^^<http://www.w3.org/2001/XMLSchema#string>	"Arabako Foru Aldundia"^^<http://www.w3.org/2001/XMLSchema#string>	"Bilbao"^^<http://www.w3.org/2001/XMLSchema
"Planificación de RRHH en un contexto de consolidación presupuestaria"^^<http://www.w3.org/2001/XMLSchema#string>	"Arabako Foru Aldundia"^^<http://www.w3.org/2001/XMLSchema#string>	"Bilbao"^^<http://www.w3.org/2001/XMLSchema

Ilustración 47: Resultado consulta 4 (select)

6.3.2 Consultas ASK

Consulta 5. ¿Existe algún curso que se imparta en Euskera?

```
ask  {?curx :idioma_curso :Euskera . }
```

El resultado a la consulta es **true**.

Podemos realizar la comprobación con la siguiente consulta:

```
#Cursos con sus idiomas. Muestra la lista de cursos con sus idiomas
select ?nombre ?idioma_curso
where {?curx :nombre ?nombre .
       ?curx :idioma_curso ?idioma_curso .   }
```

Y el resultado es el siguiente:

nombre	idioma_curso	lugar
"El acompañamiento social en los servicios de atención primaria"^^<http://www.w3.org/2001/XMLSchema#string>	http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano	"Bilbao"^^<http://www.w3.org/2001/XMLSchema#st
"Cursos de Urbanismo"^^<http://www.w3.org/2001/XMLSchema#string>	http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano	"Bilbao"^^<http://www.w3.org/2001/XMLSchema#st
"Cursos de Urbanismo"^^<http://www.w3.org/2001/XMLSchema#string>	http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano	"Donostia"^^<http://www.w3.org/2001/XMLSchema#
"Administrazio hizkera"^^<http://www.w3.org/2001/XMLSchema#string>	http://www.owl-ontologies.com/Ontology1365663508.owl#Euskera	"Vitoria-Gasteiz"^^<http://www.w3.org/2001/XMLSchema#st
"Gestión de Tesorería"^^<http://www.w3.org/2001/XMLSchema#string>	http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano	"Donostia"^^<http://www.w3.org/2001/XMLSchema#
"Planificación de RRHH en un contexto de consolidación presupuestaria"^^<http://www.w3.org/2001/XMLSchema#string>	http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano	"Bilbao"^^<http://www.w3.org/2001/XMLSchema#st
"Evaluación previa de impacto de género"^^<http://www.w3.org/2001/XMLSchema#string>	http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano	"Vitoria-Gasteiz"^^<http://www.w3.org/2001/XMLSchema#st

Ilustración 48: Comprobación consulta 5 (ask)

Consulta 6. Pregunta si existen cursos Presenciales impartidos en Vitoria-Gasteiz en euskera

```
ask where {?curx :nombre ?nombre .
       ?curx a :Presencial .
       ?curx :idioma_curso :Euskera .
       ?curx :lugar ?lugar_aux .
       ?lugar_aux :nombre ?lugar
       filter regex(?lugar, "Vitoria-Gasteiz")
}
```

La respuesta es **true**, y la comprobación se puede realizar con la consulta siguiente:

```
#Cursos Presenciales impartidos en Vitoria-Gasteiz en euskera
select ?nombre ?lugar
where {?curx :nombre ?nombre .
       ?curx a :Presencial .
       ?curx :idioma_curso :Euskera .
       ?curx :lugar ?lugar_aux .
       ?lugar_aux :nombre ?lugar
       filter regex(?lugar, "Vitoria-Gasteiz")
}
```

Que nos muestra como resultado:

nombre	lugar
"Administrazio hizkera"^^<http://www.w3.org/2001/XMLSchema#string>	"Vitoria-Gasteiz"^^<http://www.w3.org/2001/XMLSchema#string>

6.3.3 Consultas CONSTRUCT

Consulta 7. Creación de un nuevo grafo RDF mediante construct

Desde RDF -> Namespaces he creado el Namespace para el ejemplo.

A continuación he ejecutado la consulta:

```
PREFIX ej: <http://www.owl-ontologies.com/ejemplo.owl#>
construct {?x ej:nom ?nombre .
           ?x ej:id ?idioma_curso .}
where {?x :nombre ?nombre .
       ?x :idioma_curso ?idioma_curso . }
```

Resultado de la consulta:

406 Not Acceptable

An appropriate representation of the requested resource index.html could not be found on this server.

Available variant(s):

- index.html , type application/rdf+xml, charset UTF-8

Ilustración 50: Resultado consulta 7

Después de realizar algunas investigaciones se constata que se pueden ejecutar consultas SPARQL desde la dirección http://localhost:8890/sparql.

Al ejecutarlo desde este punto, directamente nos genera un RDF con las tripletas correspondientes a la consulta. Se puede consultar en el anexo IV.

6.3.4 Consultas DESCRIBE

Consulta 8. Grafo RDF con la descripción de los recursos de la consulta

```
#Describe Cursos con sus idiomas. Muestra la lista de cursos con
sus idiomas
describe ?nombre ?idioma_curso
where {?curx :nombre ?nombre .
       ?curx :idioma_curso ?idioma_curso .
}
```

406 Not Acceptable

An appropriate representation of the requested resource index.html could not be found on this server.

Available variant(s):

- index.html , type application/rdf+xml, charset UTF-8

Al igual que en el caso anterior, la consulta DESCRIBE también ha habido que ejecutarla desde http://localhost:8890/sparql para obtener resultado. Se puede visualizar la consulta y el fichero resultado de la misma, en el Anexo V

6.3.5 Índices

El esquema de índices por defecto consiste en 2 índices completos sobre las cuádruplas RDF (recordemos que Virtuoso crea tablas con 4 columnas: sujeto, predicado, objeto y grafo) y 3 índices parciales. Este esquema de índices se adapta a todo tipo de espacios de trabajo, sin importar las consultas específicas. Se pueden crear índices alternativos aunque generalmente no son necesarios.

El esquema de índices consiste en los siguientes índices

PSOG – Clave primaria

POGS – Índice de mapa de bits para búsquedas en el valor de un objeto.

SP – Índice parcial para casos donde sólo se especifica el sujeto.

OP – Índice parcial para casos donde sólo se especifica el objeto.

GS – Índice parcial para casos en que sólo se especifica el grafo.

Para crear los índices se hace desde *Interactive SQL* y se utilizan las siguientes instrucciones:

```
CREATE TABLE DB.DBA.RDF_QUAD (
  G IRI_ID_8,
  S IRI_ID_8,
  P IRI_ID_8,
  O ANY,
  PRIMARY KEY (P, S, O, G) )
```

```
ALTER INDEX RDF_QUAD ON DB.DBA.RDF_QUAD
  PARTITION (S INT (0hexffff00));
```

```
CREATE DISTINCT NO PRIMARY KEY REF BITMAP INDEX RDF_QUAD_SP
  ON RDF_QUAD (S, P)
  PARTITION (S INT (0hexffff00));
```

```
CREATE BITMAP INDEX RDF_QUAD_POGS
  ON RDF_QUAD (P, O, G, S)
  PARTITION (O VARCHAR (-1, 0hexffff));
```

```
CREATE DISTINCT NO PRIMARY KEY REF BITMAP INDEX RDF_QUAD_GS
  ON RDF_QUAD (G, S)
  PARTITION (S INT (0hexffff00));
```

```
CREATE DISTINCT NO PRIMARY KEY REF INDEX RDF_QUAD_OP
  ON RDF_QUAD (O, P)
  PARTITION (O VARCHAR (-1, 0hexffff));
```

Tanto la tabla inicial como cada uno de los índices ya existían por lo que Virtuoso nos ha devuelto en todos los casos un mensaje similar al siguiente:

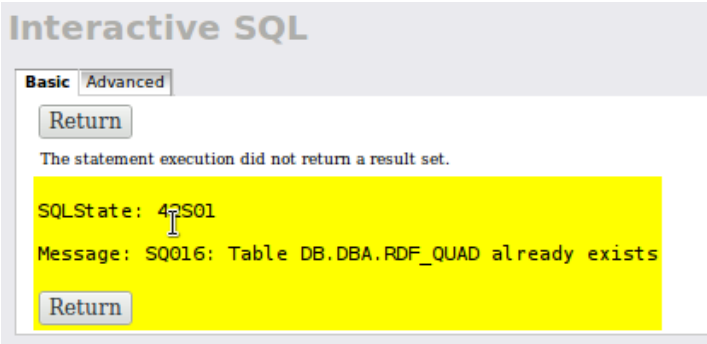


Ilustración 52: Resultado de creación de tabla para índice

7 Conclusiones

7.1 Acerca de OWLIM

La instalación aunque no es sencilla se consigue razonablemente. Los tutoriales y ayudas encontradas están bastante claros.

La interfaz de trabajo Sesame Workbench hace que sea sencilla la interacción con la ontología. Los menús de interacción son pocos pero suficientes para realizar consultas y navegar en los Namespaces y los atributos de los datos.

La carga de la ontología es sencilla. No hay ningún problema de ejecución de las consultas SPARQL. Los filtros se ejecutan correctamente.

Es una herramienta sencilla de utilizar y eficiente

7.2 Acerca de Virtuoso

La instalación inicial de Virtuoso no es muy complicada.

El acceso a la herramienta Conductor es sencillo. Pero luego nos damos cuenta de que hay muchísimas posibilidades para hacer las cosas. Comenzar a utilizar la herramienta no es tarea fácil, ya que cuesta encontrar la manera de realizar la carga de los datos de la ontología. Además el proceso de carga no es en absoluto intuitivo. Las ayudas son un poco caóticas y cuesta encontrar lo que se necesita.

A la hora de realizar consultas SPARQL se nos dan muchas posibilidades. Se pueden ejectuar desde el entorno SPARQL puro o utilizando la entrada de datos para SQL. Desde Interactive SQL la visualización resulta mucho más compacta y atractiva, ya que no muestra las URI's asociadas a los resultados.

En cuanto a las consultas SPARQL se ejecutan correctamente salvo si en el filtro de la consulta introducimos un literal que contenga tildes. En ese caso la consulta no funciona desde el CONDUCTOR.

Las consultas de tipo CONSTRUCT y DESCRIBE no pueden ser ejecutadas desde el Interface Conductor. Si se pueden ejecutar desde SPARQL.

7.3 OWLIM vs Virtuoso

OWLIM es una herramienta sencilla pero que resulta muy eficiente y rápida en las consultas. En general puede ser una herramienta adecuada para una gran número de instalaciones. Además tiene la ventaja de ser un base de datos nativa RDF. No se han detectado errores ni problemas de ejecución.

Virtuoso resulta una herramienta muy potente y con grandes posibilidades de representación de los datos gracias a la utilización de gráficos además de los datos tabulados. No todos los tipos de consultas pueden ser ejecutadas desde el interfaz Conductor. En ocasiones, cuando se ejecutan muchas consultas seguidas tiende a colgarse. Esto puede ser debido a que al no ser una base de datos nativa RDF y estar SPARQL embebido en SQL las consultas no llegan a ser tan eficientes.

Las consultas se ejecutan en memoria y esto hace que sea a la vez su ventaja y su defecto: su ventaja porque ante la mayor parte de las consultas, se ejecutarán muy rápido. Defecto, porque las consultas que llenen la memoria, harán que el servidor no pueda responder.

En ambos sistemas tanto la carga de datos como las consultas, se ejecutan muy rápido. Haría falta unos juegos de datos muy grandes para intentar establecer diferencias en el rendimiento. No obstante se observan ciertos problemas en Virtuoso, al utilizar algunos datos con tildes o al realizar consultas repetidamente.

A favor de Virtuoso se puede decir que además de ser una base de datos semántica también se puede utilizar como base de datos relacional, lo que la hacer muy versátil.

En contra de Virtuoso se puede decir que precisamente debido a su gran versatilidad, hace que algunas tareas que podrían ser sencillas, se compliquen debido a lo intrincado de las soluciones. Además al no ser una base de datos nativa RDF, la gestión de datos semánticos da algunos problemas en las consultas, por lo que se requerirá más esfuerzo de configuración.

En resumen parece que OWLIM es más adecuada para la gestión de almacenes de datos semánticos. La carga de la ontología es sencilla, las consultas se ejecutan eficientemente y no se han detectado problemas ni de funcionamiento, ni de configuración. Virtuoso abarca otras posibilidades, además de la semántica al ser una

base de datos relacional, pero es más difícil de utilizar. La carga de datos no es nada intuitiva, y da problemas de en algunos tipos de consultas.

La versión estudiada para ambos productos ha sido la versión Open-source. En ambos casos existe la posibilidad de adquirir versiones que nos permitirán gestionar mayor cantidad de datos y tener más funcionalidades.

8 Glosario

ARPANET. (Advanced Research Projects Agency Network) Red de computadoras creada por encargo del Departamento de Defensa de los Estados Unidos como medio de comunicación para los diferentes organismos del país.

CERN (Conseil Européen pour la Recherche Nucléaire) Organización Europea para la Investigación Nuclear

DCMI. Dublin Core Metadata Iniciative. Organización cuyo objetivo es el fomento del uso de metadatos mediante el uso de estándares interoperables con vocabularios especializados, para describir recursos de la información. Ha creado el modelo de metadatos Dublin Core.

DCU. Diseño centrado en el usuario

DSpace. Software de código abierto que provee herramientas para la administración de colecciones digitales, y comúnmente es usada como solución de repositorio institucional. Soporta una gran variedad de datos, incluyendo libros, tesis, fotografías, filmes, video, datos de investigación y otras formas de contenido. Los datos son organizados como ítems que pertenecen a una colección; cada colección pertenece a una comunidad. Está liberado bajo licencia BSD que permite a los usuarios personalizar o extender el software según se necesite. http://www.dspace.org/

DTD. Document Type Definition. Es una definición de un documento SGML o XML que especifica restricciones en la estructura y en la sintaxis del mismo.

Dublin Core. **DC**. Modelo de metadatos elaborado y auspiciado por la DCMI (Dublin Core Metadata Initiative), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos. Las implementaciones de DC usan generalmente XML y se basan en RDF.

EPrints. Es paquete de software open-source dedicado a construir repositorios de acceso directo compatibles con OAI-PMH. Muchas de sus características son comunes a los sistemas de gestión documental pero es utilizado fundamentalmente en repositorios institucionales y diarios científicos. http://www.eprints.org/

Greenstone. Software de código abierto para el desarrollo de bibliotecas digitales y repositorios institucionales en varios idiomas y distribuido bajo la Licencia Pública General GNU. http://www.greenstone.org/index_es

Host. Se refiere a las computadoras conectadas a una red.

HTML. Hypertext markup lenguage. Lenguaje de marcado hipertextual

Jena. Infraestructura Java para la construcción de aplicaciones para web semántica.

KOS. Knowldedge Organization System. Sistema de organización del conocimiento, ampliamente utilizado para generalizar lenguajes documentales.

LOD. Linked Open Data. Datos abiertos enlazados. Técnica para publicar información estructurada en la web. Consulta de datos LOD: http://thedatahub.org/

Metadatos. Datos que describen a otros datos

MILNET (MILitary NETwork) red de comunicación militar de las Fuerzas armadas de los Estados Unidos**.**

Namespace o espacio de nombres XML. Es una recomendación W3C para proporcionar elementos y atributos con nombre único en una instancia XML. Una instancia XML puede contener nombres de elementos o atributos procedentes de más de un vocabulario XML. Si a cada uno de estos vocabularios se le da un espacio de nombres, un ámbito semántico propio, referenciado a una URI donde se listen los términos que incluye, se resuelve la ambigüedad existente entre elementos o atributos que se llamen igual, la homonimia. Los nombres de elementos dentro de un espacio de nombres deben ser únicos.

NCP (Network Control Program) Conjunto original de protocolos de control de red de ARPANET

OAI-PMH. Open Archives Initiative-Protocol Metadata Harvesting. Herramienta de interoperabilidad independiente de la aplicación. Permite realizar el intercambio de información. Desde proveedores de servicio se puedan hacer búsquedas en la información recopilada en distintos repositorios asociados.

Ontología. Describen tipos de entidades y la forma en que se relacionan entre sí. Especificación formal y explícita de una conceptualización compartida.

Open Journal Systems (OJS) es un software de código abierto para la administración de revistas creado por el Public Knowledge Project, bajo licencia GNU General Public License. Fue diseñado para facilitar el desarrollo de publicaciones de acceso libre, proveyendo la infraestructura técnica para la presentación en línea de artículos de revista y el flujo editorial completo. http://pkp.sfu.ca/espanol , http://www.ojs.es/

OWL Es una extensión de RDF Schema. Es un lenguaje desarrollado por el consorcio W3C cuyo objetivo es la publicación y compartición en la web de ontologías.

OWLIM. Base de datos semántica.

Protégé. Aplicación para creación de ontologías.

RDF. Resource Description Framework. Modelo de datos simple que permite identificar recusos y sus relaciones con otros recursos.

RDF. Resource Description Framework. Modelo para la descripción estructurada de recursos.

RDFS. RDF Schema. Vocabulario para describir las propiedades y las clases de los recursos RDF, permitiendo establecer jerarquías.

SGBD: Sistema de gestión de base de datos.

SGML Standard Generalized Markup Language. Estándar de Lenguaje de Marcado Generalizado. Consiste en un sistema para la organización y etiquetado de documentos. Sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.

SKOS. Simple Knowledge Organization System, tiene por objetivo proporcionar un modelo para representar distintos tipos de vocabularios controlados como tesauros, clasificadores y taxonomías.

SPARQL Protocol and RDF Query Language. Lenguaje de consulta parecido al SQL que sirve para hacer consultas en bases de datos semánticas. Es el estándar del W3C.

TCP/IP. Es una familia de protocolos de red en los que se basa Internet. Los protocolos más más importantes que la componen son: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

URI. Uniform Resource Identifier. Identificador único de recurso.

URL (Uniform Resource Locator) Localizador de recursos uniforme. Se utiliza para nombrar recursos en Internet.

Virtuoso DB. Base de datos híbrida pero con capacidad para gestionar bases de datos semánticas.

W3C (World Wide Web Consortium) consorcio internacional que realiza recomendaciones para la web.

WAI (Web Accessibility Initiative) Iniciativa para la Accesibilidad Web es una rama de W3C que trabaja la accesibilidad de la Web. Existen diferentes pautas de accesibilidad: A, AA y AAA. Actualmente en la administración se trata de alcanzar la “doble A”.

World Wide Web. Red informática mundial

XML según la wikipedia: *eXtended Mark Language,* lenguaje extendido de marcas. XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

9 Referencias

9.1 Webgrafía

- **Wikipedia**. Historia de Internet

  http://es.wikipedia.org/wiki/Historia_de_Internet

- **Internet Society**. Breve historia de Internet

  http://www.internetsociety.org/es/breve-historia-de-internet

- **Wikipedia**. Tim Berners-Lee

  http://es.wikipedia.org/wiki/Tim_Berners-Lee

- **Jose Luis Fuertes Castro et Loic Martinez Normand**. Accesibilidad Web. TRANS. Num II-2007. Disponible en línea 12/04/2013

  http://www.trans.uma.es/pdf/Trans_11/T.135-154CastroyNormand.pdf

- **Christian Bizer, Richard Cyganiak, and Tobias Gauß**. The RDF Book Mashup: From Web APIs to a Web of Data . Freie Universit at Berli. Disponible en línea a 12/04/2013

  http://www.dfki.uni-kl.de/~grimnes/2007/06/SFSW07Papers/6.pdf

- **Pablo Navarro**. Computando la vida social: el nuevo papel de los métodos de análisis sociológico en Internet. Universidad de Oviedo. Disponible en línea a 12/04/2013

  http://www.reis.cis.es/REIS/PDF/REIS_108_061168422555245.pdf

- **Wikipedia**. Web semántica

  http://es.wikipedia.org/wiki/Web_sem%C3%A1ntica

- María Jesús Lamarca Lapuente. Hipertexto: El nuevo concepto de documento en la cultura de la imagen. **RDF**

  http://www.hipertexto.info/documentos/rdf.htm

- **Luis Criado Fernández**. TESIS DOCTORAL. PROCEDIMIENTO SEMI-AUTOMÁTICO PARA TRANSFORMAR LA WEB EN WEB SEMÁNTICA. UNED Madrid 2009. Disponible en línea a 12/04/2013

  http://e-spacio.uned.es/fez/eserv.php?pid=tesisuned:IngInf-Lcriado&dsID=Documento.pdf

- **Pablo Castells**. La web semántica. Escuela Politécnica Superior Universidad Autónoma de Madrid. Disponible en línea a 12/04/2013
  http://www.ii.uam.es/~castells
  http://arantxa.ii.uam.es/~castells/publications/castells-uclm03.pdf

- **Virtuoso** Open Source edition

  http://www.openlinksw.com/wiki/main/#Download,%20Build,%20and%20Use%20--%20Get%20Started!

- **OWLIM**

  http://www.ontotext.com/owlim

- **Swoogle**. Buscador de ontologías

  http://swoogle.umbc.edu/

- **Juan Antonio Pastor Sanchez**. Tesis Doctoral. Diseño de un sistema colaborativo para la creación y gestión de tesauros en Internet basado en SKOS. Universidad de Murcia. Disponible en línea a 12/04/2013

  http://www.tesisenred.net/bitstream/handle/10803/10914/PastorSanchez.pdf?sequence=1

- **How to import ontology into virtuoso?**

  http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtTipsAndTricksGuideImportOntology

- Imagen **Semantic Web**

http://oxygen.lcs.mit.edu/images/SemanticWeb.jpg

- Imagen **Linked Open data**

  https://nablux.net/tgp/media/rdftalk2012/img/lod-datasets_2009-03-05_colored.png

- Instalación de Virtuoso

  http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSDebianNotes

9.2 Bibliografía

- **Juan Antonio Pastor Sánchez**. Tecnologías de la Web Semántica. Barcelona: Editorial UOC, 2011. ISBN:8497884744.
- **Lluís Codina, Mari-Carmen Marcos, Rafael Pedraza (coords.)** Web semántica y sistemas de información documental. Editorial Trea S.L.
- **Juan Antonio Pastor Sánchez (coordinador), Pedro Manuel Díaz Ortuño, Francisco Javier Martínez Méndez, Tomás Saorín Pérez.** Construcción de servicios de información digital. Editorial DM, 2010. ISBN: 978-84-8425-784-4
- **Segaran, Toby; Evans, Colin; Taylor, Jamie**. Programming the semantic web. O'Reilly Media, Inc. 2009. 302 p. Ebook ISBN:978-0-596-80518-0

  http://www.gise.cse.iitb.ac.in/wiki/images/3/3a/PTSW.pdf

10 Anexo I

10.1 Instalación de OWLIM

Para la instalación de OWLIM es necesario tener instalado TOMCAT. A su vez, para instalar Tomcat, hace falta que esté instalado JAVA. Por lo tanto primero mostraré la instalación de los requisitos y por último la instalación de OWLIM propiamente dicha.

10.1.1 Instalación de Java JDK

Desde la terminal se ejecuta

```
$ sudo apt-get install openjdk-6-jdk
```

De esta forma java se instala en el siguiente ubicación: /usr/lib/jvm/java-6-openjdk

Después hay que editar las variables de entorno de java. Para ello se edita /etc/bash.bashrc

```
$ sudo vim /etc/bash.bashrc
```

Al final del fichero se añaden las variables de entorno de java, para ello se añaden las siguientes líneas:

```
JAVA_HOME=/usr/lib/jvm/jav-6-openjdk
export JAVA_HOME
PATH=$PATH: PATH:$JAVA_HOME/bin
export PATH
```

Después de rearrancar el equipo comprobamos que las variables de entorno están correctas con:

```
$ echo $JAVA_HOME
$ echo $PATH
```

Por último comprobamos la versión de java que se ha instalado con

```
$ java -version
$ javac -version
```

10.1.2 Instalación de Tomcat

Desde una terminal se ejecuta las siguientes órdenes:

```
$ sudo apt-get update
$ sudo apt-get install tomcat6
$ sudo apt-get install tomcat6-admin
```

El programa se instala en /var/lib/tomcat6.

Ahora hay que configurar los usuarios que accederán a la administración de Tomcat.

Para ello editamos tomcat-users.xml

```
$ sudo vim /etc/tomcat6/tomcat-users.xml
```

Entre las etiquetas <tomcat-users> y </tomcat-users> insertamos las líneas para definir el usuario administrador y su contraseña.

```
<tomcat-users>
<role rolename="admin"/>
<role rolename="manager"/>
<user username="tomcat" password="tomcat" roles "admin, manager"/>
</tomcat-users>
```

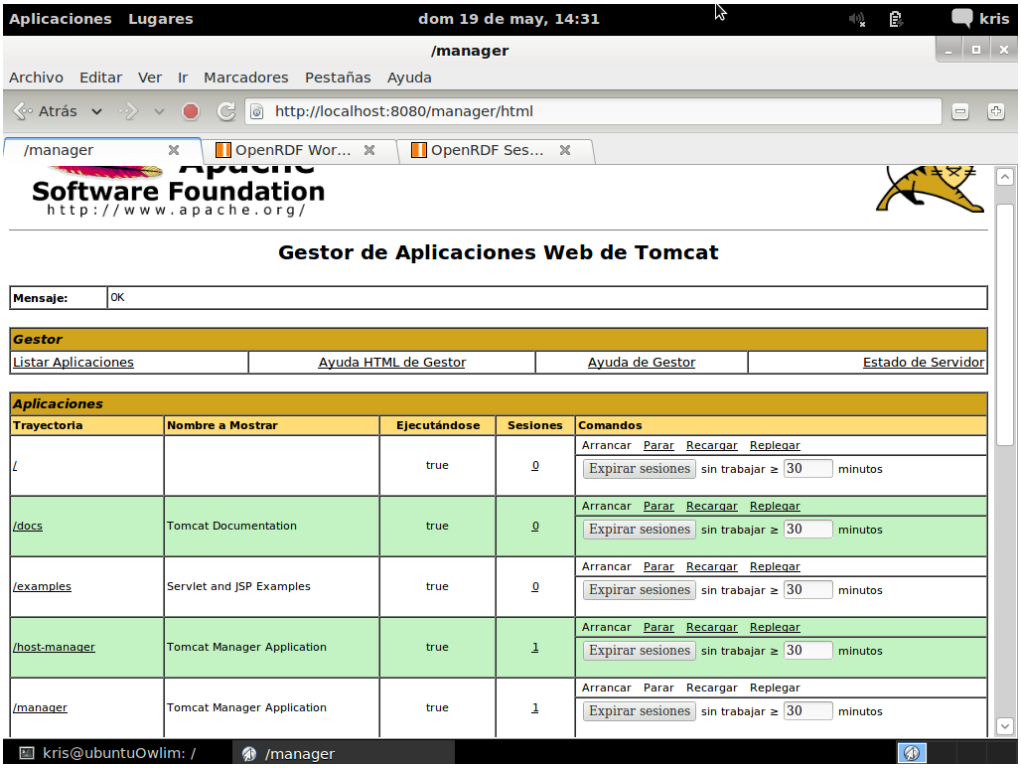
Para arrancar, rearrancar o parar tomcat se utiliza:

```
$ sudo .//etc/init.d/tomcat6 start
$ sudo .//etc/init.d/tomcat6 restart
$ sudo .//etc/init.d/tomcat6 stop
```

Para comprobar que Tomcat funciona escribimos la siguiente dirección en el navegador:

http://localhost:8080/manager/html

Y nos muestra una pantalla como la siguiente:

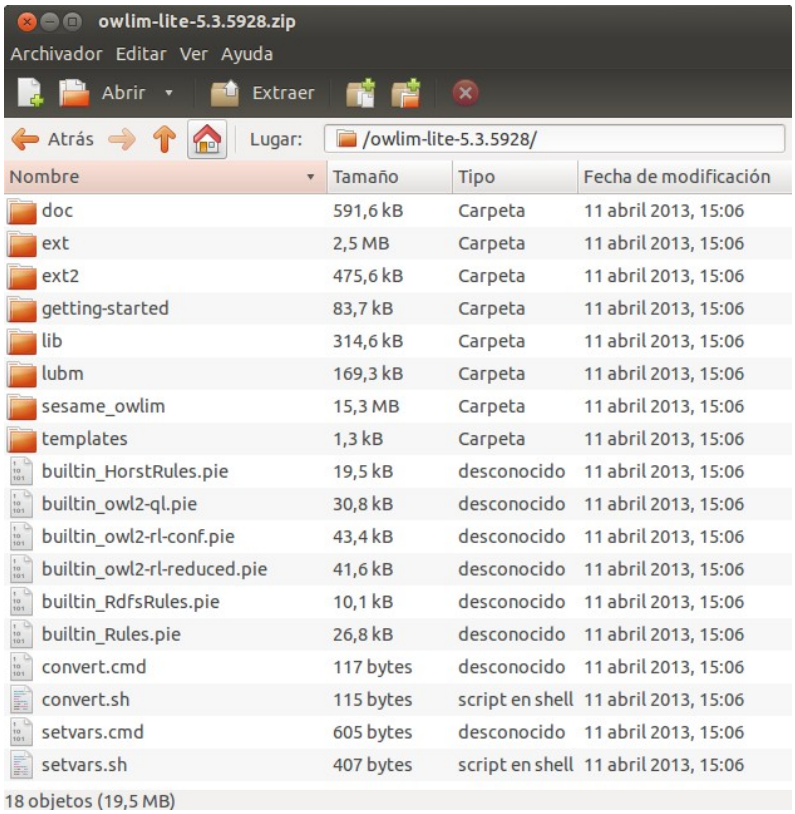


10.1.3 Instalación de OWLIM propiamente dicha

Una vez instalados los prerequisitos se puede proseguir con la instalación de OWLIM.

Me he bajado `owlim-lite-5.3.5928.zip`

El contenido del zip es el que se muestra a continuación.



El directorio sesame_owlim contiene lo siguiente:

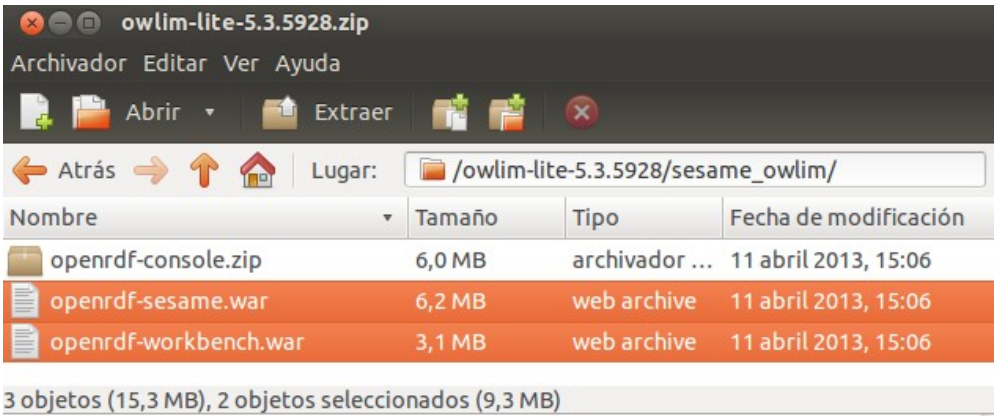
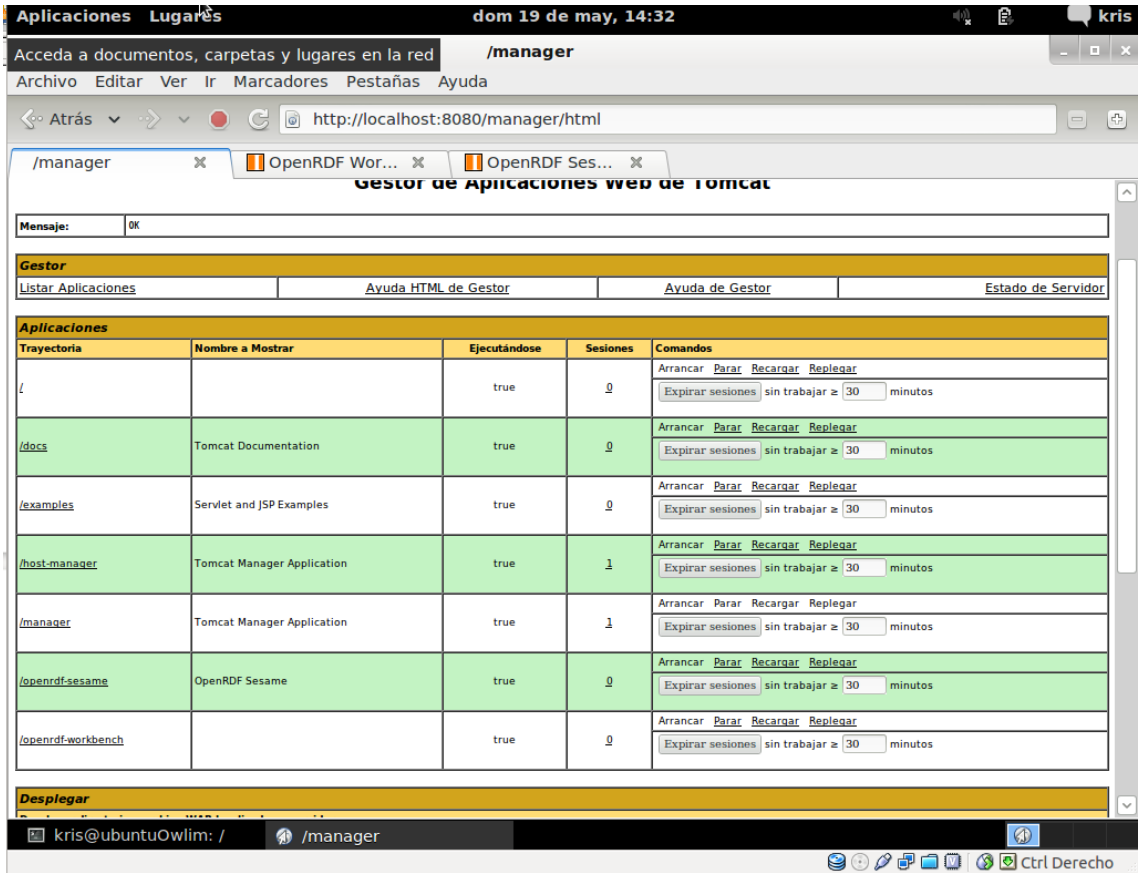


Ilustración 55: Contenido de la carpeta sesame_owlim

Los ficheros `openrdf-sesame.war` y `openrdf-workbench.war` se copian a la ubicación del servidor `tomcat /var/lib/tomcat6/webapps`

En el navegador accedemos a la siguiente dirección http://localhost:8080/managaer/html para comprobar que se han cargado correctamente.

A partir de ese momento se pueden ver en el servidor tomcat las nuevas aplicaciones.



Para que probar que funcionan cada una de las aplicaciones, en el navegador escribo http://localhost:8080/openrdf-workbench y nos muestra lo siguiente:

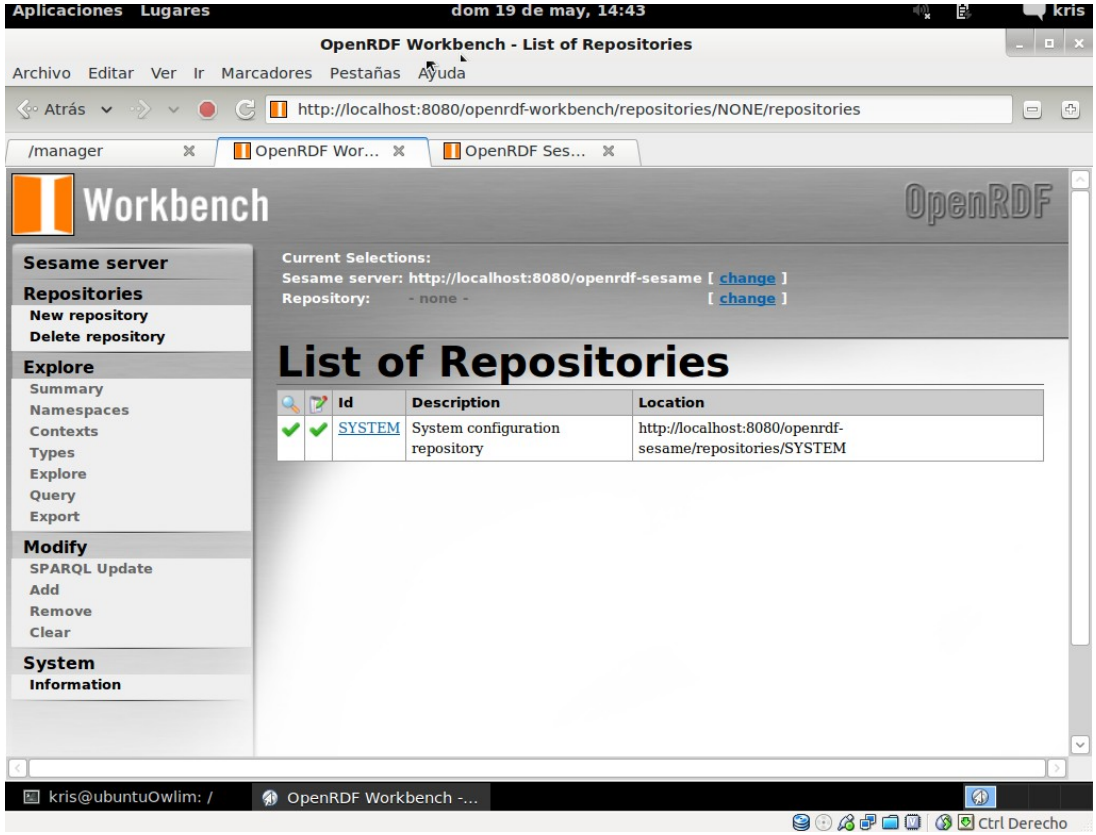


Ilustración 57: Comprobación instalación Sesame workbench

Después probamos la dirección http://localhost:8080/openrdf-sesame. En este caso nos da un error que dice lo siguiente: "unable to create logging directory /usr/share/tomcat6/.aduna/openrdf-sesame/logs"

Para solucionar el error se crea el directorio que falta y se le asigna como usuario y grupo, tomcat6:

```
$ cd /usr/share/tomcat6
$ sudo mkdir .aduna
$ sudo chown tomcat6:tomcat6 .aduna
```

Despueś de corregido el error al entrar en la consola de sesame me muestra lo siguiente:

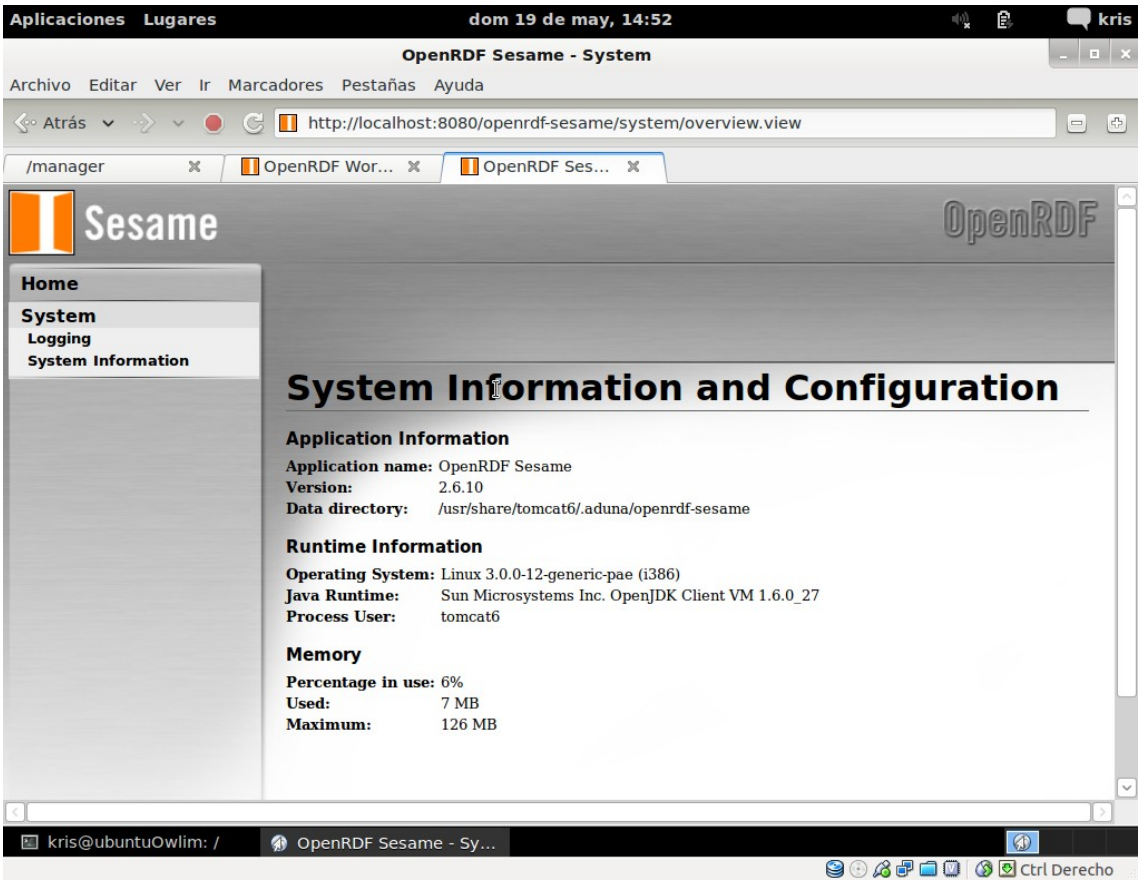


Ilustración 58: Comprobación openrdf-sesame

11 Anexo II

11.1 Instalación de Virtuoso

Se puede instalar directamente desde los repositorios de Ubuntu.

Desde el terminar del server de Ubuntu primero actualizamos el índice de paquetes disponibles.

```
$ sudo apt-get update
```

A continuación realizamos la instalación de virtuoso.

```
$ sudo aptitude install virtuoso-opensource
```

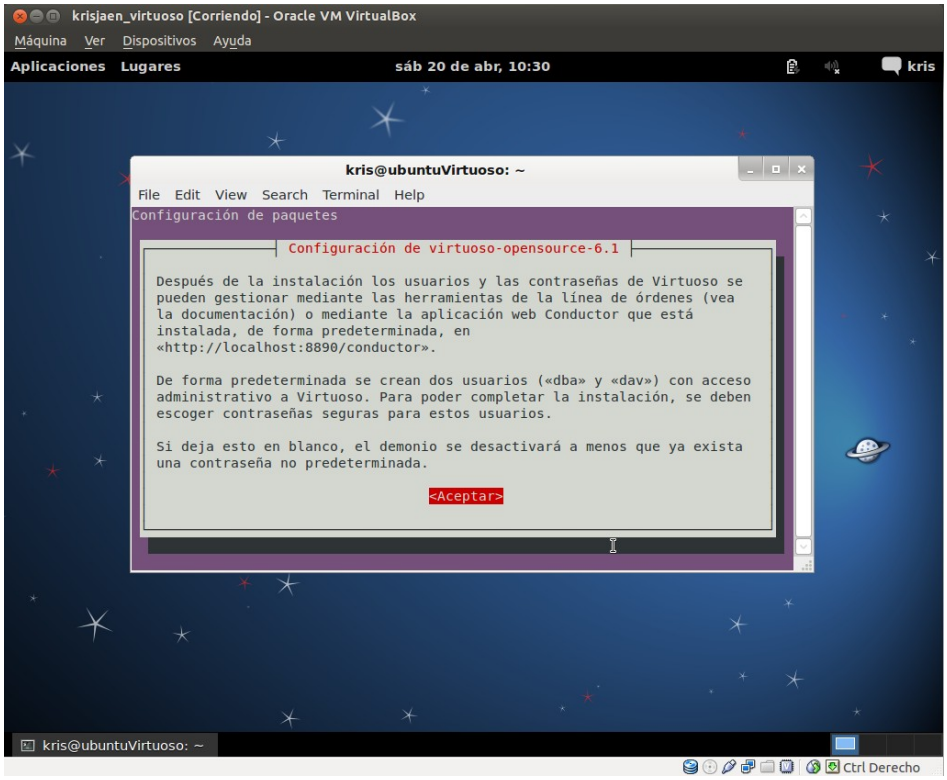


Ilustración 59: Instalación de virtuoso

Las pantallas de instalación piden una contraseña para los usuarios dba y dav.

Posteriormente cuando nos conectemos a *Virtuoso Conductor* necesitaremos dichas contraseñas. *Virtuoso Conductor* se instala por defecto junto con el paquete de instalación de *Virtuoso Open-Source.*

Comprobamos los paquetes de virtuoso susceptibles de ser instalados.

```
kris@ubuntuVirtuoso:~$ apt-cache search '^virtuoso'
```

Después instalamos los paquetes vad. En realidad lo que hace es bajarse los paquetes que le hemos indicado. Para instalarlos realmente lo haremos desde *Virtuoso Conductor.*

```
sudo apt-get install virtuoso-vad-{isparql,ods,tutorial}
```

Después de instalado podemos acceder al administrador de la base de datos desde el navegador poniendo http://localhost:8890/conductor y se nos muestra la siguiente pantalla de administración.

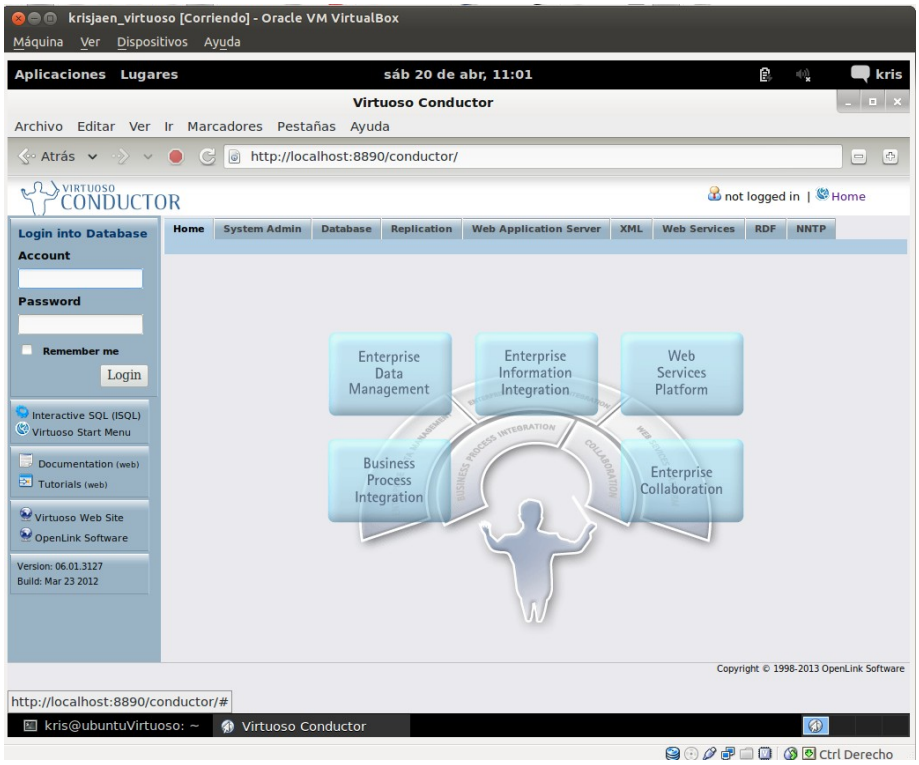
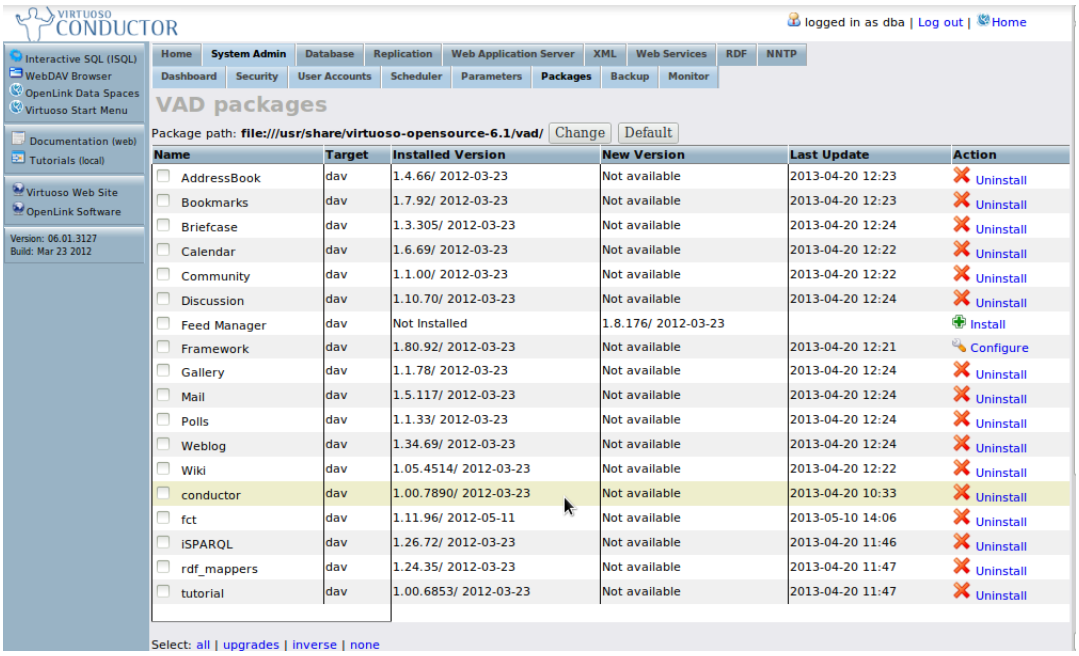


Ilustración 60: Acceso a Virtuoso desde Conductor

Nos podemos conectar con el usuario dba, con la contraseña que le hayamos puesto en la instalación. Si la hemos dejado en blanco, la contraseña por defecto será dba.

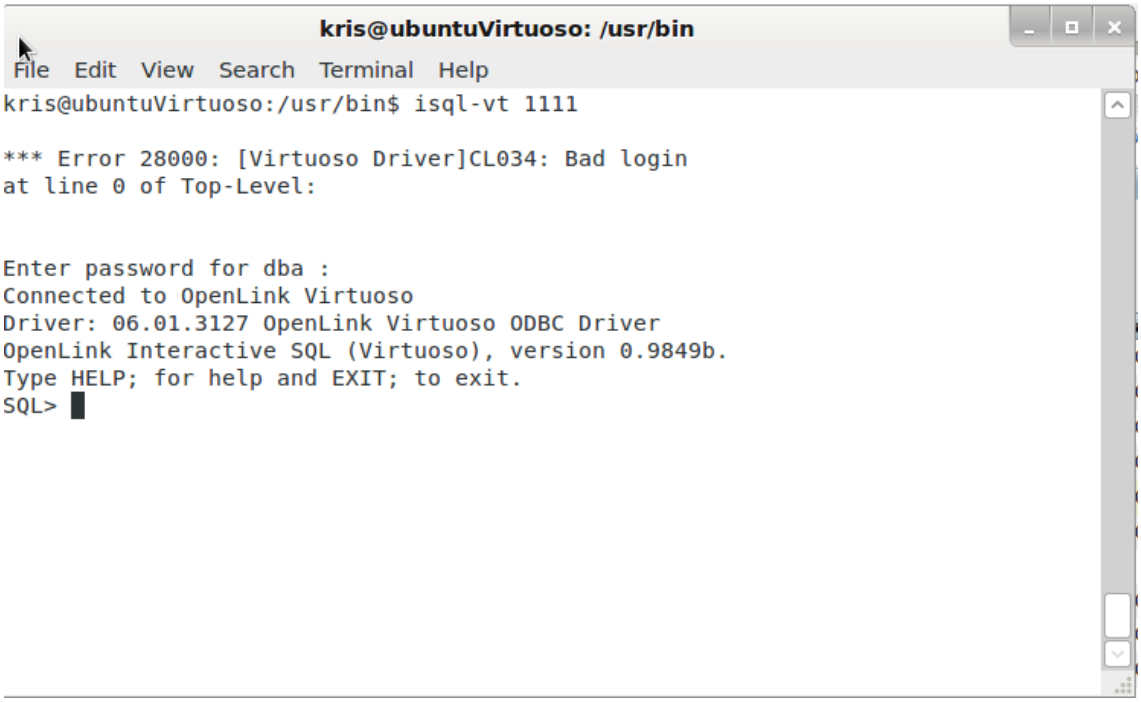
Para instalar los paquetes VAD que queramos, una vez conectados iremos a *System Admin -> Packages.* Nos mostrará la pantalla que podemos ver a continuación desde la cual podemos instalar, desinstalar o actualizar los paquetes que deseemos.



Name	Target	Installed Version	New Version	Last Update	Action
AddressBook	dav	1.4.66/ 2012-03-23	Not available	2013-04-20 12:23	Uninstall
Bookmarks	dav	1.7.92/ 2012-03-23	Not available	2013-04-20 12:23	Uninstall
Briefcase	dav	1.3.305/ 2012-03-23	Not available	2013-04-20 12:24	Uninstall
Calendar	dav	1.6.69/ 2012-03-23	Not available	2013-04-20 12:22	Uninstall
Community	dav	1.1.00/ 2012-03-23	Not available	2013-04-20 12:22	Uninstall
Discussion	dav	1.10.70/ 2012-03-23	Not available	2013-04-20 12:24	Uninstall
Feed Manager	dav	Not Installed	1.8.176/ 2012-03-23		Install
Framework	dav	1.80.92/ 2012-03-23	Not available	2013-04-20 12:21	Configure
Gallery	dav	1.1.78/ 2012-03-23	Not available	2013-04-20 12:24	Uninstall
Mail	dav	1.5.117/ 2012-03-23	Not available	2013-04-20 12:24	Uninstall
Polls	dav	1.1.33/ 2012-03-23	Not available	2013-04-20 12:24	Uninstall
Weblog	dav	1.34.69/ 2012-03-23	Not available	2013-04-20 12:24	Uninstall
Wiki	dav	1.05.4514/ 2012-03-23	Not available	2013-04-20 12:22	Uninstall
conductor	dav	1.00.7890/ 2012-03-23	Not available	2013-04-20 10:33	Uninstall
fct	dav	1.11.96/ 2012-05-11	Not available	2013-05-10 14:06	Uninstall
iSPARQL	dav	1.26.72/ 2012-03-23	Not available	2013-04-20 11:46	Uninstall
rdf_mappers	dav	1.24.35/ 2012-03-23	Not available	2013-04-20 11:47	Uninstall
tutorial	dav	1.00.6853/ 2012-03-23	Not available	2013-04-20 11:47	Uninstall

Ilustración 61: Paquetes DAV para Virtuoso

Se pueden realizar una gran cantidad de operaciones de gestión desde el *Conductor.* También existe la posibilidad de conectarse desde línea de comandos.



Desde la terminal también podremos realizar tareas de gestión y mantenimiento de la base de datos.

12 Anexo III

12.1 Fichero RDF, resultado de consulta CONSTRUCT en OWLIM

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
      xmlns="http://www.owl-ontologies.com/Ontology1365663508.owl#"

xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
      xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
      xmlns:sqwrl="http://sqwrl.stanford.edu/ontologies/built-
ins/3.4/sqwrl.owl#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:swrl="http://www.w3.org/2003/11/swrl#"
      xmlns:psys="http://proton.semanticweb.org/protonsys#"
      xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:pext="http://proton.semanticweb.org/protonext#"

xmlns:swrla="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#"
      xmlns:ej="http://www.owl-ontologies.com/ejemplo.owl#"
      xmlns:sesame="http://www.openrdf.org/schema/sesame#"
```

```
      xmlns:fn="http://www.w3.org/2005/xpath-functions#">

<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#Online_37">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Datuak
babesteari buruzko oinarrizko kontzeptuak</ej:nom>
      <ej:id                            rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Euskera"/>
</rdf:Description>

<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#Presencial_7">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Planificació
n de RRHH en un contexto de consolidación presupuestaria</ej:nom>
      <ej:id                            rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Castellano"/>
</rdf:Description>

<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#Presencial_22">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Gestión   de
Tesorería</ej:nom>
      <ej:id                            rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Castellano"/>
</rdf:Description>

<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#10">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Administrazi
o hizkera</ej:nom>
      <ej:id                            rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Euskera"/>
</rdf:Description>

<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#Online_26">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Excel   2003
básico</ej:nom>
      <ej:id                            rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Castellano"/>
</rdf:Description>

<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#Presencial_34">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Cursos    de
Urbanismo</ej:nom>
      <ej:id                            rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Castellano"/>
</rdf:Description>
```

```
<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#Presencial_31">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">El
acompañamiento    social    en    los    servicios    de    atención
primaria</ej:nom>
      <ej:id                          rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Castellano"/>
</rdf:Description>

<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#Mixto_25">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Estudios   e
investigaciones cuantitativas con enfoque de género</ej:nom>
      <ej:id                          rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Castellano"/>
</rdf:Description>

<rdf:Description                          rdf:about="http://www.owl-
ontologies.com/Ontology1365663508.owl#Presencial_52">
      <ej:nom
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Evaluación
previa de impacto de género</ej:nom>
      <ej:id                          rdf:resource="http://www.owl-
ontologies.com/Ontology1365663508.owl#Castellano"/>
</rdf:Description>
</rdf:RDF>
```

13 Anexo IV

13.1 Consulta CONSTRUCT en Virtuoso y fichero resultado

La consulta se ha ejecutado desde http://localhost:8890/sparql

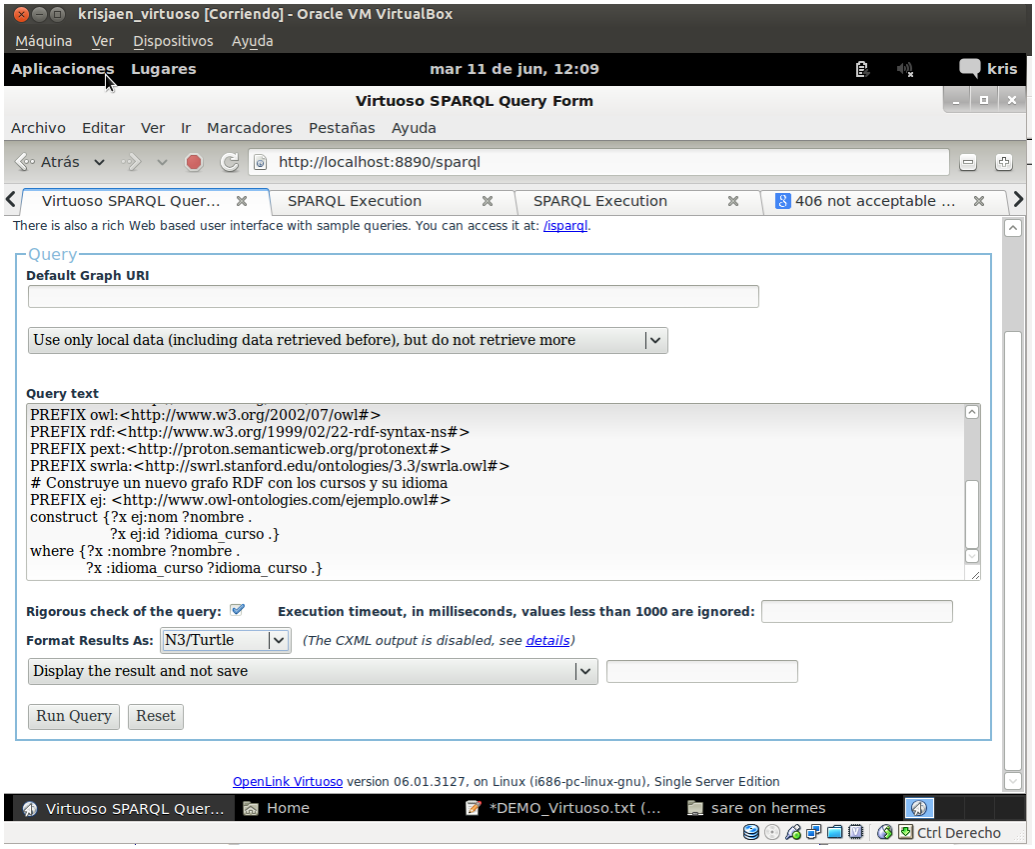


Ilustración 63: Consulta construct con Virtuoso

Tal y como se ve en la consulta la salida del fichero se ha dicho que sea en formato N3/Turtle. Se muestra a continuación el fichero resultado

```
@prefix ej:<http://www.owl-ontologies.com/ejemplo.owl#> .
@prefix ns1:     <http://www.owl-
ontologies.com/Ontology1365663508.owl#> .
ns1:Presencial_31      ej:id ns1:Castellano .
ns1:Online_37    ej:id ns1:Euskera .
ns1:Presencial_34      ej:id ns1:Castellano .
<http://www.owl-ontologies.com/Ontology1365663508.owl#10>     ej:id
      ns1:Euskera .
ns1:Presencial_22      ej:id ns1:Castellano .
ns1:Presencial_7 ej:id ns1:Castellano .
ns1:Online_26    ej:id ns1:Castellano .
ns1:Presencial_52      ej:id ns1:Castellano .
ns1:Mixto_25     ej:id ns1:Castellano .
```

14 Anexo V

14.1 Consulta DESCRIBE en Virtuoso y fichero resultado

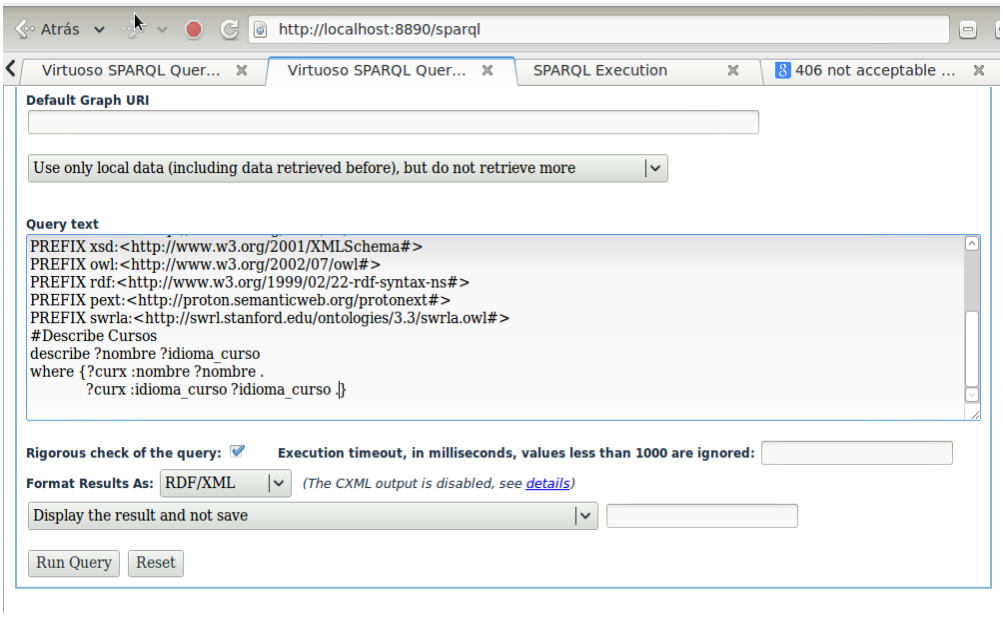


Ilustración 64: Consulta DESCRIBE con Virtuoso

En este caso se ha dicho que el fichero resultado de la consulta sea en formato RDF/XML. Se muestra a continuación el fichero resultado.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<rdf:RDF>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Online_37">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Euskera"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Euskera">
    <rdf:type rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Idioma"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Presencial_7">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Mixto_25">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Presencial_52">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Online_26">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Presencial_22">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Presencial_34">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano">
    <rdf:type rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Idioma"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#10">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Euskera"/>
  </rdf:Description>
 ▼<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1365663508.owl#Presencial_31">
    <n0pred:idioma_curso rdf:resource="http://www.owl-ontologies.com/Ontology1365663508.owl#Castellano"/>
  </rdf:Description>
</rdf:RDF>
```

Ilustración 65: Fichero resultado consulta Describe