UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

MODALIDAD PRESENCIAL

Publicación de datos universitarios observando los principios de Linked Data

Trabajo de fin de carrera previo a la obtención del Título de Ingeniero en Sistemas Informáticos y Computación.

Autor:

Darwin Leonardo Sarango Romero

Director:

Ing. Nelson Oswaldo Piedra Pullaguari

Codirector:

Ing. Janet Alexandra Chicaiza Espinosa

Loja – Ecuador

2011

CERTIFICACIÓN

Ing.

Nelson Oswaldo Piedra

DIRECTOR DEL PROYECTO DE FIN DE CARRERA

CERTIFICA:

Que una vez concluido el trabajo de investigación con el tema **"PUBLICACIÓN DE DATOS ENLAZADOS OBSERVANDO LOS PRINCIPIOS DE LINKED DATA"** previa la obtención del título de Ingeniero en Sistemas Informáticos y Computación, realizado por el profesional en formación **DARWIN LEONARDO SARANGO ROMERO** ha sido dirigido, supervisado y revisado en todas sus partes, por lo mismo, cumple con los requisitos legales exigidos por la Universidad Técnica Particular de Loja, quedando autorizada su presentación.

Loja, Diciembre del 2011

- -

Ing. Nelson Oswaldo Piedra

CESIÓN DE DERECHOS

Yo, **Darwin Leonardo Sarango Romero**, declaro ser autor del presente trabajo y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que su parte pertinente textualmente dice: "Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la universidad".

- -

Darwin Leonardo Sarango Romero

AUTORÍA

Todos los criterios, opciones, afirmaciones, análisis, interpretaciones, conclusiones, recomendaciones y todos los demás aspectos vertidos en el presente trabajo son de absoluta responsabilidad del autor.

Loja, Diciembre de 2011

- -

Darwin Leonardo Sarango Romero

DEDICATORIA

El presente trabajo está dedicado de manera muy especial a mi madres Rosa Elvira Romero Ortiz y María del Carmen Sarango Romero, a mis hermanos David Ismael Delgado Sarango y Jonathan Oscar Jaramillo Sarango, a mi tía Martha Sarango Romero, a mi prima Guiselle Álvarez Sarango, a mi tío Eduardo Sarango Romero, y a Ruth Hidalgo Tene quienes son mi fortaleza y motor de mi vida.

A todos mis profesores por brindarme sus conocimientos a lo largo de toda la carrera, también a mis amigos y compañeros con quienes he tenido el gusto de compartir su valioso tiempo.

Darwin Leonardo Sarango Romero

AGRADECIMIENTO

Agradezco a mis madres y a mi familia quienes me inculcaron valores para mi formación como persona, a mis profesores quienes contribuyeron a mis conocimientos académicos, en especial al Ing. Nelson Piedra, director de tesis que con su conocimiento, empeño y motivación, fue el guía durante el desarrollo del presente trabajo de investigación.

A la Ing. Janeth Chicaiza, codirectora de tesis, quien con sus conocimientos contribuyó con el presente trabajo, además a los ingenieras María del Carmen Cabrera, María Belén Mora, a los ingenieros Jorge López, Eduardo Encalada miembros del grupo de Investigación de la Escuela de Ciencias de la Computación, y a todas las personas que de una u otra manera me brindaron su apoyo en esta ardua tarea de investigación.

Darwin Leonardo Sarango Romero

ÍNDICE DE CONTENIDO

Certificación……..... 2
Cesión de derechos……... 3
Autoría……..... 4
Dedicatoria…….....5
Agradecimiento……...6
Índice de contenidos……7
Índice de Figuras……..... 11
Índice de Tablas…….....14
Resumen……. 15
Introducción…….....17

TABLA DE CONTENIDO

1.0. Estado Del Arte 19
1.1. Introducción 20
1.2. Evolución de la web 21
1.2.1. Fundamentos de la Web 21
1.2.2. Web Social 22
1.2.3. Web Semántica 24
1.2.4. Anotaciones Semánticas 27
1.2.5. Web de Datos 28
1.2.6. Linked Data 28
1.2.7. Principios de Linked Data 28
1.2.8. Publicación de Datos con RDF 29
1.3. Tecnologías 31

1.3.1. Identificadores Uniformes de Recursos (URI´s) 32
1.3.2. Representación de Datos RDF 32
1.3.3. XML vs RDF 35
1.3.4. RDF Schema 35
1.3.5. Propiedades para la Definición de Esquemas. 36
1.4. Motores de Representación (RDF Data Store) 36
1.4.1. 4Store 37
1.4.2. OpenLinkVirtuoso 37
1.4.3. Sesame 38
1.4.4. Oracle Spatial 11g 38
1.4.5. AllegroGraph 39
1.5. Explotación de datos (SPARQL) 40
1.6. Visualización y/o Consumo (Special Browser) 42
1.6.1. /facet 42
1.6.2. Lena 42
1.6.3. Jena2 43
1.6.4. OpenLink RDF Browser 43
1.6.5. Disco HiperData Browser 43
1.7. Vocabularios para Linked Data 44
1.7.1. Dublin Core Metadata Iniciative(DCMI) 44
1.7.2. FOAF (Friend to Friend) 45
1.7.3. Creative Commons (CCREAL) 46
1.7.4. AIISO (Academic Institution Internal Strcture Ontology). 47
1.8. Fuentes de Datos OCW/OER 48
1.8.1. OpenCourseWare 48
1.9. Open Educational Resource (OER) 51
1.10. Trabajos Relacionados 52
1.10.1. CourseWare 53
1.10.2. Locah Project 53
1.10.3. MitOpenCourseWare RDFizer 54
2.0. RDF Triple Store System y APIS Comparativa 55

2.1. Introducción 56
2.2. RDF Triple Store System 56
2.2.1. Sesame 57
2.2.2. AllegroGraph 58
2.2.3. OpenLink Virtuoso 60
2.2.4. Oracle Spatial 11g 61
2.2.5. 4Store 63
2.2.6. Jena 64
2.3. Conclusión 68
3.0. Identificación Y Selección De Fuentes De Datos 69
3.1. Introducción 70
3.2. OERs y Linked Data 70
3.3. Descripción del problema de Datos 71
3.3.1. El problema de la Interoperabilidad entre repositorios OCW 73
3.4. ¿Qué datos necesito? 75
3.5. Proceso de Obtención de Datos 75
4.0. Descripción De Vocabularios OCW-OER-Licencias 79
4.1. Introducción 80
4.2. Uso de Vocabularios para Describir Datos 80
4.3. Reutilizando Vocabularios Existentes 81
4.3.1. Vocabulario OCW 82
4.4. Vocabulario OCW/OER/Licence. 83
5.0. Generación De RDFS y RDF Data. 94
5.1. Introducción. 95
5.2. Descripción del Problema 95
5.3. Loxa VocabRDFAdmin 95
5.4. Arquitectura de la Aplicación. 97
5.5. Desarrollo de la Aplicación. 98
5.6. Resultados Obtenidos. 103
6.0. Implementación De Un Modelo De Visualización Para Datos RDF 109
6.1. Introduction 110

6.2. Arquitectura de la Aplicación 110
6.3. Resultados Obtenidos 114
8.0. Conclusiones, Recomendaciones y Trabajos Futuros 126
8.1. CONCLUSIONES 127
8.2. RECOMENDACIONES 129
8.3. TRABAJOS FUTUROS 130
8.0. BIBLIOGRAFÍA 131
9.0. GLOSARIO 137
ANEXOS 142
ANEXO 1: MANUAL DE INSTALACIÓN DE ORACLE SPATIAL 11g EN UBUNTU 10.04 143
ANEXO 2: LEVANTAR ORACLE SPATIAL 11g EN UBUNTU 11.4 155
ANEXO 3: CONFIGURACIÓN DEL ENTORNO SEMÁNTICO 157
ANEXO 4: INSTALACIÓN DE ORACLE 10G EN MAC OS X SERVER 10.5.4 160
ANEXO 5: CONFIGURACIÓN DEL ENTORNO SEMÁNTICO EN MAC OS X SERVER 10.5.4 171
ANEXO 6: INSTALACIÓN DE VIRTUOSO SOBRE OPEN-SOURCE EN UBUNTU 10.4 174
ANEXO 7: INSTALACIÓN DE ALLEGRO GRAPH SOBRE UBUNTU 11.4 176
ANEXO 8: INSTALACIÓN DE 4STORE SOBRE UBUNTU 10.4 32 bits 182
ANEXO 9: INCORPORAR LIBRERÍA DE JENA A NETBEANS 186
ANEXO 10: PAPER CIENTÍFICO 188

ÍNDICE DE FIGURAS

FIGURA 1 EVOLUCIÓN DE LA WEB 20
FIGURA 2 TORRE SEMÁNTICA (W3C) 25
FIGURA 3 MODELO RDF (BIZER, 2010) 31
FIGURA 4 MODELO RDF CON HTTP URIS (BIZER, 2010) 31
FIGURA 5 LINK RDF (BIZER, 2010) 31
FIGURA 6 EJEMPLO DE URI 32
FIGURA 7 THE LAYERCAKE (W3C) 41
FIGURA 8 CONSULTA SPARQL SOBRE DBPEDIA 41
FIGURA 9 CREATIVE COMMONS OVERVIEW 46
FIGURA 10 USUARIOS DE OCW (VALKENBURG, 2011) 50
FIGURA 11 OER Y ELEMENTOS (NELSON & ROMERO, 2010) 52
FIGURA 12 NUBE DE LINKED DATA 53
FIGURA 13 CAPACIDADES DE ORACLE SEMÁNTICO 63
FIGURA 14 ARQUITECTURA DE JENA (TORRE, 2009) 65
FIGURA 15 OSI/OSI MODEL AND ENVISIONED LINKED DATA INTEROPERABILITY LAYER (EURECOM, 2010) 71
FIGURA 16 METADATOS DE LA UTPL 72
FIGURA 17 METADATOS SOBRE EL CANCILLER DE LA UTPL 73
FIGURA 18 METADATOS DE UN CURSO OCW DE LA UPM 76
FIGURA 19 PROCESAMIENTO DE DATOS 77
FIGURA 20 PROCESO DE OBTENCIÓN DE DATOS 78
FIGURA 21 USO DE VOCABULARIO COMUNES (BIZER ET AL, 2011) 81
FIGURA 22 DUBLIN CORE Y RDF 82
FIGURA 23 VOCABULARY OCW/OER/LICENCE 84
FIGURA 24 LOXA VOCABRDFADMIN 96
FIGURA 25 ARQUITECTURA DE LA APLICACIÓN 98
FIGURA 26 CADENA SQL 99
FIGURA 27 CADENA SQL DE CONSULTA 100
FIGURA 28 CADENA CON CONSULTA SPARQL 101
FIGURA 29 LOXA VOCABRDFADMIN APP 103
FIGURA 30 CARGAR ARCHIVO DE CONEXIÓN 104
FIGURA 31 TABLAS RECUPERADAS DE LA BASE DE DATOS 105
FIGURA 32 COMPONENTE AUTOCOMPLETE 105
FIGURA 33 COMPONENTE DRAG AND DROP 106

FIGURA 34 CARGAR VOCABULARIO RDF 106
FIGURA 35 RDFS/RDF GENERADO 107
FIGURA 36 ARQUITECTURA DE VISUALIZADOR RDF LUD 111
FIGURA 37 INTERFAZ PRINCIPAL DE VISUALIZADOR RDF LUD 115
FIGURA 38 DETALLE DEL RECURSO 115
FIGURA 39 DETALLE DEL CURSO 117
FIGURA 40 DETALLE DEL OPEN COURSE WARE 118
FIGURA 41 LISTA DE AUTORES 119
FIGURA 42 BUSCAR OER 120
FIGURA 43 CONFIGURACIÓN DE DIRECTORIOS 143
FIGURA 44 CREACIÓN DE GRUPOS 143
FIGURA 45 ASIGNACIÓN DE PERMISOS 144
FIGURA 46 CREACIÓN DE DIRECTORIOS 144
FIGURA 47 CREACIÓN DE ENLACES SIMBÓLICOS 144
FIGURA 48 CONFIGURACIÓN DE ARCHIVOS DEL SISTEMA 145
FIGURA 49 CONFIGURACIÓN DEL DISPLAY 146
FIGURA 50 INSTALLER ORACLE 147
FIGURA 51 CONFIGURACIÓN DE VARIABLES 148
FIGURA 52 COMPROBACIÓN DE REQUISITOS 148
FIGURA 53 SOLUCIONAR PROBLEMAS DE REQUISITOS 149
FIGURA 54 SOLUCIONAR REQUISITOS (SOLUCIONADO) 149
FIGURA 55 SOLUCIONAR PROBLEMA DE RDBMS 150
FIGURA 56 SOLUCIONAR PROBLEMAS DE RDBMS (SOLUCIONADO) 150
FIGURA 57 CONFIGURACIÓN DEL LISTENER 151
FIGURA 58 CONFIGURACIÓN DEL ENTERPRISE MANAGER 152
FIGURA 59 CONFIGURACIÓN DE LOS PARÁMETROS DE INICIALIZACIÓN 152
FIGURA 60 INICIALIZACIÓN DE LA CREACIÓN DE LA BASE DE DATOS 153
FIGURA 61 FINALIZACIÓN DEL ASISTENTE DE CONFIGURACIÓN DE BASE DE DATOS 153
FIGURA 62 ASIGNACIÓN DE PERMISOS 154
FIGURA 63 FINALIZACIÓN DE LA INSTALACIÓN 154
FIGURA 64 CONFIGURACIÓN DE DIRECTORIOS 155
FIGURA 65 START LISTENER 155
FIGURA 66 SQLPLUS 156
FIGURA 67 SGA 156
FIGURA 68 USUARIO SYSDBA 157
FIGURA 69 EJECUCIÓN DE SCRIPTS 157
FIGURA 70 USUARIO MDSYS 157
FIGURA 71 HABILITAR ENTORNO SEMÁNTICO 158
FIGURA 72 CREACIÓN DE TABLESPACE 158
FIGURA 73 CREACIÓN DE TEMPORARY TABLESPACE 158
FIGURA 74 ASIGNACIÓN DEL TABLESPACE A UNA RED SEMÁNTICA 159

FIGURA 75 CREACIÓN DE UN USUARIO 159
FIGURA 76 ALMACENAMIENTO DE DATOS RDF. 159
FIGURA 77 CREACIÓN DE USUARIO PARA ORACLE 162
FIGURA 78 CREACIÓN DE USUARIO PARA ORACLE CON WORKGROUP MANAGER 162
FIGURA 79 CREACIÓN DE USUARIO ORACLE OWNER CON WORKGROUP MANAGER 163
FIGURA 80 GRUPO OINSTALL 163
FIGURA 81 CAMBIO DE PARÁMETROS DEL KERNEL 164
FIGURA 82 SHELL MAC OS X 165
FIGURA 83 ORACLE INSTALLER 166
FIGURA 84 SELECCIÓN DE MÉTODO DE INSTALACIÓN 167
FIGURA 85 CONFIGURACIÓN DE PARÁMETROS DE LA BASE DE DATOS 167
FIGURA 86 ORACLE UNIVERSAL INSTALLER 168
FIGURA 87 EXECUTE CONFIGURACIÓN SCRIPTS 169
FIGURA 88 MODIFICACIÓN DE BASH_PROFILE 169
FIGURA 89 LEVANTAR INSTANCIA DE ORACLE 170
FIGURA 90 CREACIÓN DEL TABLESPACE 171
FIGURA 91 EJECUCIÓN DE PROCEDIMIENTO 171
FIGURA 92 CREACIÓN DE USUARIOS 171
FIGURA 93 OTORGACIÓN DE PRIVILEGIOS 172
FIGURA 94 CREACIÓN DE OBJETOS PARA RDF 172
FIGURA 95 CREACIÓN DE MODELO RDF 172
FIGURA 96 INSERCIÓN DE TRIPLETAS 172
FIGURA 97 CONSULTA DE DATOS RDF 173
FIGURA 98 REQUERIMIENTOS PREVIOS 174
FIGURA 99 EJEMPLO PARA INSTALAR UN PAQUETE 174
FIGURA 100 EJECUCIÓN DE AUTOGEN.SH 174
FIGURA 101 ESTABLECIMIENTO DE VARIABLES DE ENTORNO 174
FIGURA 102 OPEN LINK VIRTUOSO 175
FIGURA 103 VMWARE INSTALL 176
FIGURA 104 MONTAR ALLEGRO GRAPH 177
FIGURA 105 ALLEGRO GRAPH VIRTUAL MACHINE 177
FIGURA 106 ALLEGRO GRAPH WEBVIEW 178
FIGURA 107 CREACIÓN DE UN TRIPLE STORE 178
FIGURA 108 ABRIR AG TRIPLE DATA 179
FIGURA 109 CREACIÓN DE USUARIOS Y ROLES 179
FIGURA 110 CREACIÓN DE REPOSITORIOS 180
FIGURA 111 LOAD FILES 180
FIGURA 112 CONSULTAS SPARQL 180
FIGURA 113 ECLIPSE (JAVA Y JENA) 181
FIGURA 114 CONFIGURACIÓN DE SOURCES.LIST 182
FIGURA 115 CONFIGURACIÓN DE INSTALACIÓN DE RAPTOR 183

FIGURA 116 INSTALACIÓN DE RAPTOR 183
FIGURA 117 INSTALACIÓN DE RASQAL 184
FIGURA 118 CREACIÓN DE UNA BASE DE DATOS CON 4STORE 184
FIGURA 119 CREACIÓN DE BIBLIOTECA JENA 186
FIGURA 120 AGREGAR BIBLIOTECA JENA 187

ÍNDICE DE TABLAS

TABLA I ESTADÍSTICAS DE INTERNET 21
TABLA II COMPONENTES Y FUNCIONES DE LA WEB 2.0 23
TABLA III PARTES DE UNA EXPRESIÓN RDF 33
TABLA IV RDF EN RDF/XML CORRESPONDIENTE A LA EXPRESIÓN RDF ANTERIOR 33
TABLA V XML VS RDF 35
TABLA VI COMPARATIVA ENTRE LOS RDF TRIPLE STORE SYSTEM 40
TABLA VII CATEGORÍAS DE UN CURSO OCW 50
TABLA VIII CARACTERÍSTICAS RDF TRIPLE STORE 67
TABLA IX DESCRIPCIÓN DE CLASES DEL VOCABULARIO OCW/OER/LICENCE 86
TABLA X DESCRIPCIÓN DE RELACIONES DEL VOCABULARIO OCW/OER/LICENCE 87
TABLA XI DESCRIPCIÓN DE ELEMENTOS DEL VOCABULARIO OCW/OER/LICENCE 93
TABLA XII LIBRERÍAS UTILIZADAS 102
TABLA XIII REQUERIMIENTOS DE ORACLE PARA MAC OS X 160

RESUMEN

La web de datos se ha convertido en un potencial considerable para varios sectores entre ellos el sector educativo, permitiendo que los datos educativos se conecten con más datos relacionados.

El presente proyecto tiene por objetivo lograr que los Dataset de los OpenCourseWare de las principales Universidades de Iberoamérica y los Dataset de los Open Educational Resource estén disponibles como RDF, tomando en cuenta los principios de Linked Data, con el objetivo de mejorar la experiencia de uso de los OCW y OERs, analizando para ello las principales tecnologías semánticas, componentes, estándares, motores de representación RDF y herramientas de visualización y consumo.

Se analizó varios motores de representación RDF como Oracle, Virtuoso, Sesame, AllegroGraph y 4store, en varias plataformas, con el propósito de determinar más adecuado para el dominio de la investigación.

Se construyó una aplicación web que permite la generación de RDFS, y RDF a partir de fuentes de datos estructurados (MySQL). Además se construyó un visualizador de datos donde se muestran los datos generados con el aplicativo, esto con el objetivo de demostrar el consumo de datos RDF.

Keywords-component: *RDF, metadatos, ontología, RDF Schema, Web Semántica, Linked Data, SPARQL, Anotación semántica, URI, HTML, DBPEDIA, XML, HTTP, Motores de Representación de RDF, Jena, Sesame, SPARQL end point. Java. Prefix. Tripletas, Recursos, Propiedades y Objetos.*

INTRODUCCIÓN

La cantidad de datos vinculados publicados en la web ha experimentado un gran crecimiento, y la lista de recursos disponibles como datos enlazados crece día a día, sin embargo la mayoría de datos se encuentran sueltos en la web convirtiéndose en datos pobres, controlados por aplicaciones, las mismas que se auto mantienen, es por esto que los datos enlazados es la forma que tiene la web semántica para vincular todos estos datos sueltos en la web, de forma que se referencian como lo hacen los enlaces de las páginas web con el objetivo de facilitar la búsqueda de información, ofreciendo de esta manera una mejor accesibilidad y brindar mayor significado a los datos.

Actualmente en la web de datos encontramos, alguna información vinculada relacionada a OCWs (Open Course Ware) como autores, organizaciones, licencias, áreas de conocimiento, entre otras, esta información puede ser enlazada y consumida desde una fuente de datos basada en OERs (Open Educational Resource) y OCW, añadiendo de esta manera conocimiento y contexto a los OERs.

La solución que se plantea es contar con un Dataset rico, abierto e integrado de RDF de las principales organizaciones OCWs de Iberoamérica, tomando como clave la desambiguación de los OERs, con identificadores únicos en este caso URIs, a través de un vocabulario consensuado, teniendo como resultado final la obtención de una serie de metadatos involucrados a un curso OCW en particular.

Estos Objetivos se cumplirán a través de 5 capítulos; en el **Primer Capítulo** se presenta el marco teórico sobre tecnologías, herramientas, estándares, principios e iniciativas de Linked Data, enmarcado en un estado del arte. En el **Segundo Capítulo** se presenta una comparativa entre los principales RDF Triple Store disponibles, con el objetivo de analizar su rendimiento y posteriormente seleccionar el más adecuado para el dominio de esta investigación. En el **Tercer Capítulo** se detalla el primer proceso del círculo de vida del proyecto la identificación y selección de las fuentes de datos en nuestro caso de OCW y OER, analizando la importancia de la vinculación de datos en el campo de los recursos educativos abiertos. En el **Cuarto Capítulo** se realiza la descripción de vocabularios tanto para OCW, OER, y Licencias, utilizando vocabularios ya establecidos. En el **Quinto Capítulo** se detalla la generación de RDFS y RDF mediante una aplicación web, que recibe de entrada una cadena de conexión a una

base de datos relacional (MySQL), con dichos datos el usuario interactúa para formar las tripletas y posteriormente generar el RDFS y RDF. En el **Sexto Capítulo** se detalla la construcción de un módulo que permite la visualización de datos del componente Linked Data University, la misma que consiste de una aplicación web transparente para el usuario.

Los resultados obtenidos permitieron corroborar los objetivos del presente trabajo, por consiguiente se establece que la utilización de los principios de Linked Data pueden ser usados para publicar datos enlazados que sirvan a estudiantes y profesores a auto educarse, además se demuestra que Linked Data permite la interoperabilidad entre sitios OCW heterogéneos. Para finalizar también se pudo establecer algunos trabajos futuros que pueden partir a raíz de este trabajo, dejando claro en que aplicaciones puede ser factible esta tendencia.

1.0. Estado Del Arte

... sin causa y sin principio es imposible que algo exista o se lleve a cabo.

Aristóteles

El arte de la retórica, Libro I, Capítulo VII

1.1. Introducción

La web fue creada hace 15 años aproximadamente y ha experimentado una evolución asombrosamente rápida, revolucionando la vida de las personas de todo del mundo, proporcionando nuevas oportunidades tanto sociales, económicas y tecnológicas, gracias a que se ha convertido en el mayor repositorio de información. Paralelamente al crecimiento espectacular de la Web, las tecnologías que la hacen posible han evolucionado rápidamente, desde las tecnologías básicas como HTML[1] y HTTP[2], hasta las tecnologías de hoy en día como Java[3], JavaScript[4], PHP[5], Flash[6], XML[7], etc. En la Figura 1 se plasma la predicción expuesta por (Spivack, 2007), sobre el pasado, presente y futuro de la web, cuya perspectiva es bastante razonable.

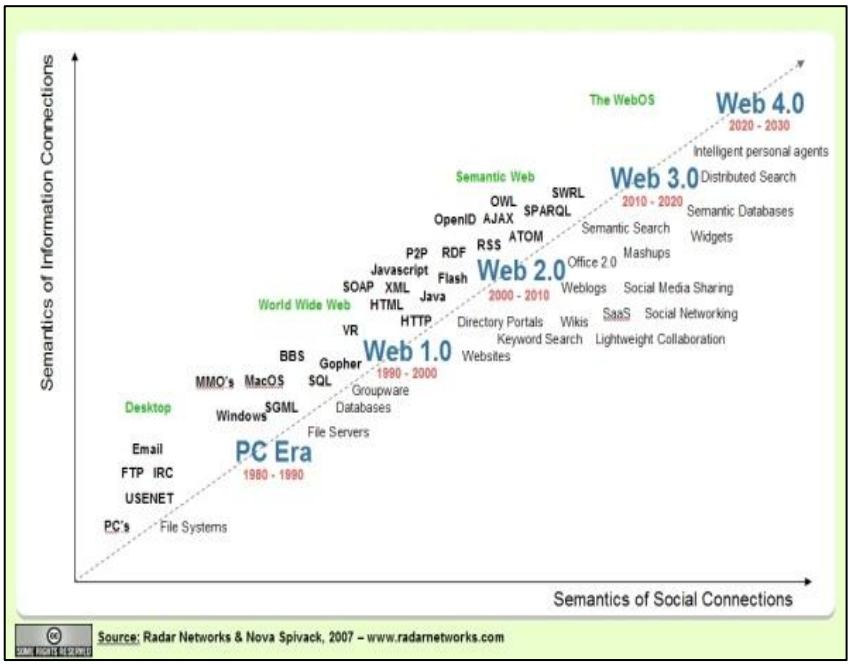


Figura 1 EVOLUCIÓN DE LA WEB

En el eje de las x se encuentran las diferentes evoluciones de la web, y en el eje de las y podemos apreciar las distintas tecnologías utilizadas.

[1] Hyper Text Mark Language http://www.w3.org/MarkUp/
[2] http://www.w3.org/Protocols/
[3] www.oracle.com/technetwork/java/index.html
[4] http://www.mozilla.org/js/
[5] http://php.apache.org/
[6] http://www.macromedia.com/
[7] Extensible markup Language http://www.w3.org/XML/

1.2. Evolución de la web

La web ha tenido un crecimiento continuo desde su creación iniciando como un proyecto para el manejo de información en un laboratorio hasta convertirse en un sistema global de información, evolución que ha sido posible gracias a que es relativamente fácil escribir una página web y es fácil de enlazar a otras páginas (Berners-Lee, 2007), además los estándares como URIs, HTTP y HTML han permitido en gran medida el crecimiento de la misma.

La historia del crecimiento de la WWW puede ser medida por el número de páginas web que se publican y el número de enlaces entre las páginas. En la actualidad existen 128.538.602 sitios web de los cuales el 72 % son comerciales (Domain Counts, 2011). La Tabla I muestra las estadísticas de manera más detallada.

Active	**New**	**TLD**
93,312,817	112,709	.Com
13,810,119	17,196	.Net
9,428,245	13,507	.Org
7,894,594	17,693	.Info
2,230,747	3,473	.Biz
1,863,398	2,676	.Us
128,671,754	**167,254**	**Total**

Tabla I ESTADÍSTICAS DE INTERNET

1.2.1. Fundamentos de la Web

Hablar de los fundamentos de la web o web 1.0 conlleva retroceder hasta donde la potencia de los ordenadores y el ancho de banda no permitían páginas como las actuales, las mismas que eran páginas de texto estáticas solo de lectura, programadas en HTML, donde la participación del usuario era pasiva permitiéndole obtener solamente información, la misma que era unidireccional y su publicación requería de conexión, a esto se suma que las aplicaciones web 1.0 son propietarias, las compañías desarrollan software que los usuarios podían descargar pero que no podían ver cómo trabajan o no podían cambiarlas (howstuffworks, 2008).

1.2.2. Web Social

Tomando la idea de Berners-Lee donde señala que: "Deberíamos ser capaces no sólo de encontrar cualquier tipo de documento en la Web, sino también de crear cualquier clase de documento fácilmente. Deberíamos no sólo poder interactuar con otras personas, sino crear con otras personas. La intercreatividad es el proceso de hacer cosas o resolver problemas juntos" (Berners-Lee, 2000), están son las ideas fundamentales acerca de la web social, término que fue acuñado por (O'Reilly, 2005), principal promotor de esta ideología. El menciona que los principios constitutivos de la web social son siete:

1. *La WWW como plataforma de trabajo:* Las compañías ofrecen software gratuito, utilizando las web como plataforma de esta manera las aplicaciones se ejecutan sobre la web y no en la computadora del usuario.
2. *El fortalecimiento de la inteligencia colectiva*: Los hipervínculos constituyen los cimientos de la web. A medida que los usuarios agreguen nuevo contenido, y nuevos sitios web estos se enlazan con la estructura de la web gracias a otros usuarios que descubren el contenido y lo enlazan.
3. *La gestión de las bases de datos como competencia básica:* lo valioso de las aplicaciones web 2.0 son los datos.
4. *El fin del ciclo de las actualizaciones de versiones del software:* nuevas aplicaciones web 2.0 se desarrollan con el objeto de sustituir a productos comerciales y de tratar a los clientes como co-desarrolladores, para aprender del consumo de los mismos, de esta manera mejorar la herramienta en forma constante y sin costes adicionales para el consumidor.
5. *Los modelos de programación ligera junto a la búsqueda de la simplicidad:* los modelos de programación ligera ponen énfasis en la reducción de la complejidad, dando lugar a la simplicidad la cual genera formas creativas de reutilizar la programación hacia la estandarización.
6. *El software no limitado a un solo dispositivo:* la web 2.0 no se limita al uso de las computadoras solamente, hoy en día dispositivos de tercera generación han hecho posible el uso de la web desde cualquier dispositivo móvil.

7. *Las experiencias enriquecedoras de los usuarios:* aplicaciones de google[8] como gmail[9], googleDocs[10], googlemaps[11] entre otras proporcionan interfaces de usuario ricas y con interactividad equivalente a la del computador.

Cristóbal Cobo y Hugo Pardo en su libro digital planeta web 2.0 (Cobo & Pardo, 2007), proponen cuatro pilares fundamentales de la Web 2.0:

1. *Social Networking:* redes sociales que facilitan la conformación de comunidades e instancias de intercambio social.
2. *Contenidos:* Herramientas que favorecen la lectura, escritura, distribución e intercambio de información en línea.
3. *Organización Social e inteligente de la información*: herramientas y recursos que facilitan el orden y el almacenamiento de la información, por medio de la etiquetación e indexación.
4. *Aplicaciones y servicios:* Herramientas, y plataformas en líneas que ofrecen servicios de valor añadido al usuario final.

En la tabla II se ejemplifica los pilares de la web 2.0.

Componente	Función	Ejemplos
Usuarios crean los contenidos	Creación	www.youtube.com, www.flicker.com, box.net.
Redes sociales y recomendadores	Social	www.facebook.com, www.delicious.com, http://www.linkedin.com/
Aplicaciones sobre la web	Creación	docs.google.com
Sistemas de colaboración	Cooperara	www.wikipedia.org, wikis

Tabla II COMPONENTES Y FUNCIONES DE LA WEB 2.0

Es verdad que la web 2.0 está marcada por el uso de etiquetas que si bien es cierto contribuye de gran manera a la organización de la información; es el uso de XML en la construcción de los metadatos que describan contenidos y relacionen datos entre sí de forma sistemática es lo que

[8] www.google.com
[9] www.gmail.com
[10] www.docs.google.com
[11] www.maps.google.es/

verdaderamente impulsara la evolución hacia una web más inteligente la llamada "Web Semántica".

1.2.3. Web Semántica

Tim Berners-Lee define a la Web Semántica (WS) como: "La WS no es una web independiente, sino una extensión de la web actual en donde a la información se le da un significado bien definido, facilitando para que las personas y las computadoras puedan trabajar en cooperación." (Berners-Lee, et al, 2001). Cuando se dice que la información debe tener un significado bien definido es hablar de relaciones con significados entre conceptos para que otorguen un alto grado de automatización, esto debido a que actualmente existen cosas que no podemos hacer de forma automática en la web, por ejemplo si utilizamos un navegador para buscar el texto "La dirección web del blog de Tim Berners-Lee", obtendremos varios resultados, la mayoría de ellos no tendrán nada que ver con lo que estamos buscando, y nos vemos en la necesidad de utilizar "técnicas de búsqueda" como utilizar sinónimos (sinonimia), o inclusive buscamos utilizando palabras en otro idioma (multilingüismo), en otras palabras realizamos una búsqueda sintáctica[12], siendo este un proceso tedioso y largo, de allí la necesidad de evolucionar en la WS.

En conclusión la web semántica es añadir metadatos semánticos a los contenidos que existen en la web, estos metadatos deben ser entendibles tanto por los humanos como por las máquinas. (Cueva, 2008).

La infraestructura de tecnologías y lenguajes necesarios para la implementación de la WS se esquematiza en una torre de siete capas según Tim Berners-Lee, como se muestra a en la Figura a continuación:

[12]Búsqueda sintáctica es un procedimiento de búsqueda donde el resultado son varios documentos que contienen los términos de la consulta.

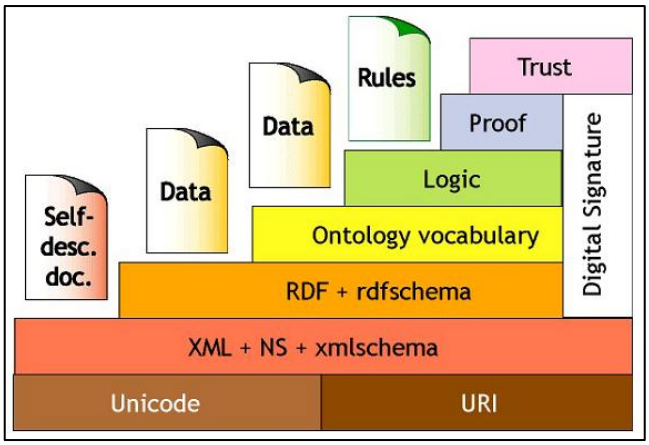


Figura 2 TORRE SEMÁNTICA (W3C[13])

La arquitectura de la WS está estructurada por niveles que establecen una jerarquía entre los mismos:

- *Nivel de recursos:* Identificación de recursos Uniformes, (URIs), su importancia está en que se puede señalar a cualquier documento o recurso de cualquier tipo en el universo de la información (Berners-Lee, 1996).
- *Nivel sintáctico:* Definición de lenguajes de etiquetado para añadir semántica a las páginas web, mediante las tecnologías XML y XMLSchema.
- *Nivel de Descripción de Recursos:* Descripción de recursos de la web, utilizando para ello el estándar RDF.
- *Nivel de Ontologías:* El conocimiento es consensuado y reutilizable a través del uso de ontologías.
- *Resto de niveles:* La lógica nos ayuda a inferir conocimiento basado en ontologías y la seguridad nos da un cierto grado de confianza con el uso de firmas digitales.

A continuación se describe el rol de cada una de las capas de la arquitectura de la WS:

- *Capa Unicode:* Es un estándar que permite la codificación de texto, que garantiza que la información no contenga símbolos extraños.
- *Capa URI:* Son cadenas de caracteres que identifican inequívocamente un recurso físico o abstracto.
- *Capa XML + NS + Xmlschema:* Es la capa más técnica de la WS ya que posibilita la comunicación entre agentes.

[13]http://www.w3.org/2001/09/21-orf/hagino-sw/slide8-0.html

- *Capa RDF + RDFSCHEMA:* Define un lenguaje universal con el que podemos expresar diferentes ideas en la WS.
- *Capa Ontología:* Sirve para clasificar la información agregando clases y propiedades a los recursos.
- *Capa Lógica:* Se especifican reglas de inferencias para las ontologías.
- *Capa de Pruebas:* Se intercambiarán "pruebas" escritas en el lenguaje unificador de la WS.
- *Capa de Confianza:* Especifica que antes de que no se compruebe la veracidad de las fuentes de información los agentes no ejecuten alguna acción.
- *Capa de Firma digital:* Las firmas digitales nos sirven para que los computadores y los agentes estén seguros que la información proviene de una fuente confiable.

A continuación se explica más detalladamente cada uno de los componentes de la WS:

- *Ontologías:* Gruber menciona que una "Ontología es una especificación formal y explícita de una conceptualización compartida". (Gruber, 1993), se dice que debe ser formal por que debe ser entendible por un computador, explicita porque, se refiere a su descripción en términos de conceptos, atributos y relaciones, y conceptualización debido a que es un modelo abstracto de un fenómeno del mundo que se desea representar, además Gruber menciona conceptos (ideas básicas que se intentan formalizar, describen y representan conceptos de un dominio específico), relaciones (representan la interacción y enlace entre los conceptos del dominio), atributos (expresan cualidades de un concepto), funciones (son relaciones donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.), instancias (utilizadas para representar objetos determinados de un concepto) y axiomas (teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología) como los componentes de una ontología para representar el conocimiento (Gruber, 1993). Ademas una ontología se compone de una taxonomía y un conjunto de reglas de inferencia para un dominio especifico de conocimiento. Una taxonomía define clases de objetos y relaciones entre ellos y una regla de inferencia es una condición sobre las relaciones entre los objetos.
- *XML (Extensible Markup Language):* Es un lenguaje que permite representar, manipular y transmitir datos estructurados y documentos, dicha estructura está constituida en forma de árboles de etiquetas con atributos, posee una relación con los metadatos, los cuales son datos estructurados que describen el significado de los

recursos como el contenido, la calidad y la condición de los mismos, facilitando de esta manera la formulación de preguntas más concretas a los motores de búsqueda pero para lograr este tipo de metadatos es necesario aplicar RDF.

- *XML Schema:* Es una extensión del XML que permite proporcionar y dictar ciertas restricciones sobre el contenido de los recursos o contenidos disponibles definiendo la sintaxis con respecto al orden, tipos de datos y formatos dentro de un documento XML.
- *OWL (Web Ontology Languaje):* Lenguaje utilizado para la descripción semántica de recursos en la web usando ontologías las mismas que tienen como objetivo proporcionar un modelo en RDF codificado en XML además se encargan de definir los términos utilizados para describir y representar un área de conocimiento, son utilizadas por los usuarios, base de datos y las aplicaciones que necesitan compartir información específica, el OWL es una extensión del RDF, por lo que este posee todas sus características y otras adicionales como son la aplicación de operaciones lógicas, además atribuye a las relaciones ciertas propiedades como la cardinalidad, simetría, transitividad o relaciones inversas.

1.2.4. Anotaciones Semánticas

Debido a que es necesario darle una descripción formal y procesable por los computadores a los recursos de la web, se debe realizar el proceso de adición de metadatos semánticos a dichos recursos este proceso es conocido como anotación semántica , las mismas que implican un proceso de análisis, extracción y marcado de la información para enriquecerla semánticamente, decimos que es un proceso de análisis debido a que se analiza el contenido de un recurso para encontrar entidades semánticas y relaciones entre ellas, para hacerlas encajar dentro de una ontología (mapping), para posteriormente realizar una descripción con metadatos utilizando el estándar RDF creado por la W3C. Existen herramientas de anotación que permiten añadir contenido semántico a las páginas web, estructurando la información publicada mediante su clasificación en base a conceptos semánticos, estas herramientas pueden ser de dos clases, las herramientas de anotación externa que permiten asociar meta información a las páginas web existentes, esta meta información se almacena en repositorios generalmente gestores de RDF, y las herramientas de anotación de autor que permiten incluir la información estructurada dentro de las páginas web mediante XML o RDF.

1.2.5. Web de Datos

La web semántica no solo trata de publicar los datos en la web si no de que estos se puedan vincular a otros dando lugar a la web de datos o Linked Data término que se refiere al conjunto de mejores prácticas para la publicación y la conexión de datos estructurados en la web (Bizer et al, 2009), este proceso de vincular datos es importante porque nos permite publicar datos usando el estándar RDF y nos permite conectar datos de diferentes repositorios de datos usando URIs, convirtiéndose en una solución práctica para la anotación semánticas de los datos de la web.

1.2.6. Linked Data

TimBerness-Lee en su artículo DesingIssues (Berners-Lee, 2006), menciona que la web semántica no solo significa poner datos en la web, también implica enlazar datos, para que una persona o una máquina pueda explorar la web de datos, permitiendo encontrar datos relacionados, por ejemplo si una organización publica datos como su nombre, tamaño, área, etc. es posible que desee indicar información acerca de su localización, gracias a que en la web existen bases de datos geográficas, la organización puede hacer referencia a los datos geográficos de esa fuente externa, enriqueciendo de esta manera sus datos.

1.2.7. Principios de Linked Data

En el mismo artículo (Berners-Lee, 2006), se delinean 4 principios para la publicación de datos en la web.

1. *Utilizar URIs (UniformResourceIdentifier) como nombres únicos para los recursos:* Si se utiliza los URIs para nombrar conceptos se consigue una abstracción del lenguaje natural y evitamos ambigüedades, por ejemplo si deseamos referirnos a un país por su topónimo este puede variar en función de su representación e idioma (EEUU, United States, United States of América, Estados Unidos).
2. *Utilizar el protocolo HTTP para nombrar y resolver la ubicación de los datos identificados mediante esas URIs:* los URIs no son solo direcciones son identificadores de los recursos, por ejemplo http://dbpedia.org/page/United_States.
3. *Representar los datos en RDF y utilizar SPARQL, como lenguaje de consulta de dichos datos:* URIs y HTTP son complementadas por el uso del estándar RDF, una tecnología que es crítica en la web de datos bebido a que provee un modelo genérico de grafos

basado en datos para estructurar y vincular datos que describan cosas en el mundo (Bizer et al, 2009).

4. *Incluir enlaces a otras URIs para permitir la localización de más datos enlazados:* los datos no deben quedar aislados, se deben enlazar a fuentes externas.

1.2.8. Publicación de Datos con RDF

Existen varios mecanismos para publicar datos vinculados en la web, mecanismos que tienen en común el uso de RDF, esto debido a que RDF provee un modelo de datos que es extremadamente simple por un lado, pero a la vez es estrictamente dura hacia la arquitectura de la WS, permitiendo así que los usuarios puedan buscar todos los URIs en un grafo RDF a través del Internet para obtener información adicional.

Según (Bizer et al, 2009) Los principios básicos de Linked Data para la publicación de datos son:

1. Asignar URIs a las entidades descritas por el conjunto de datos.
2. Usar link RDF para interconectar datos desde diferentes fuentes de datos en la web.
3. Proporcionar metadatos sobre los datos publicados.

(Corcho & Gomez, 2010), mencionan que el proceso de publicación de datos enlazados en la web incluye los pasos citados a continuación:

1. *Identificación de fuentes de datos:* se responde preguntas como ¿dónde están los datos, en que formato y que tipo de repositorio?
2. *Desarrollo de vocabularios:* se desarrollan vocabularios que se ajuste a las necesidades del proyecto, para lo cual se usan los vocabularios ya establecidos. Entre los vocabularios bien formulados para describir recursos según (Bizer, et al, 2008), se pueden citar FOAF[14], DublinCore[15],DOAP[16], CC[17], entre otros.
3. *Generación de datos RDF*: desde las diferentes fuentes de datos ya identificadas se deben generar datos RDF, para lo cual existen varias tecnologías como NOR20 y

[14]http://xmlns.com/foaf/spec/, Friend of a Friend es un vocabulario para describir personas.
[15] http: //dublincore.org/documents/dcmes-xml/, define metadatos generales de atributos.
[16] http: //trac.usefulinc.com/doap, Description of a Project, es un vocabulario para describir proyectos.
[17]http://creativecommons.org/ns CreativeCommons, es un vocabulario para describir términos de licencia.

OdeMapster[18], el mismo que es un marco integrado para la expresión formal, la evaluación, verificación y explotación de correspondencias semánticas entre ontologías y bases de datos relacionales (Ontology Engineer IngGroup, 2010).

4. *Publicación de datos RDF:* según (Bizer, 2010) es necesario usar RDF para publicar estructuras de datos en la web y crear enlaces entre los datos de diferentes Data Sources, estos enlaces permiten que los navegadores Linked Data naveguen entre los diferentes Data Sources (Representa todo lo relativo a una fuente de datos configurada por el usuario para conectarse a una Base de datos) y descubran datos adicionales. En el siguiente ejemplo se muestra como los links RDF nos adentran más profundamente en la web, partiendo de un modelo RDF flexible (Figura 3), los datos son identificados por medio de HTTP URIs (Figura 4), se resuelve más URIs (Figura 5). Estos link RDF se pueden ajustar manualmente utilizando servicios como Urigr[19] o Sindice[20], que nos permiten buscar URIs para ser reutilizadas o pueden ser generados por algoritmos automatizados de vinculación. El dominio de la aplicación es quien determinara que propiedades RDF se usan como predicados, para ligar con los Data Sources necesarios. En este punto cabe mencionar a los Data Sources y es que si los datos están almacenados en una base de datos relacional es una buena idea dejarlos allí y publicar una vista de datos enlazados en la base de datos existente, una alternativa para realizar este proceso es D2R Server[21].
5. *Desambiguación y cleaning de datos*: es importante antes de publicar testear y depurar esto con el objetivo de probar si la información será correctamente accedida, para esto existen varias herramientas que permiten trabajar directamente con datos publicados como Datos Enlazados: navegadores de RDF (Disco, OpenLinkBrowser[22]), motores de búsqueda de RDF (SWSE[23], Swoogle[24]), y librerías (SemWebClient Library[25], SWIC).
6. *Enlazar datos RDF con otras fuentes en la Web de datos:* Se debe enlazar los datos RDF con otros datos, de esta manera obedecemos al cuarto principio de Linked Data y logramos enriquecer nuestros datos.
7. *Habilitar un descubrimiento efectivo:* las aplicaciones Linked Data deben ser libres, de manera que contribuyan con la tendencia.

[18] http://neon-toolkit.org/wiki/ODEMapster
[19] http://dev.uriqr.com/
[20] http://www.sindice.com/
[21] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/index.html
[22] http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html
[23] http://swse.org/
[24] swoogle.umbc.edu
[25] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/

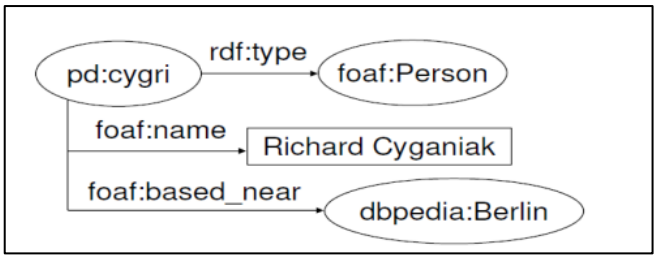


Figura 3 MODELO RDF (Bizer, 2010)

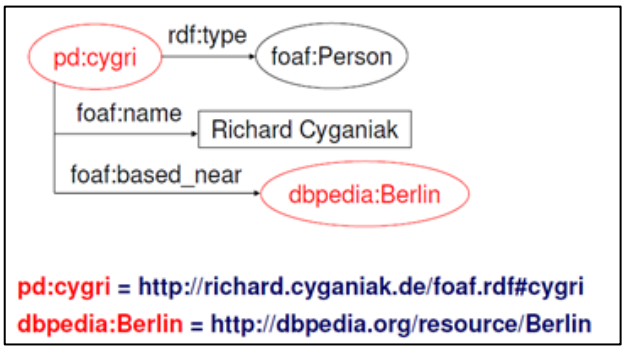


Figura 4 MODELO RDF CON HTTP URIS (Bizer, 2010)

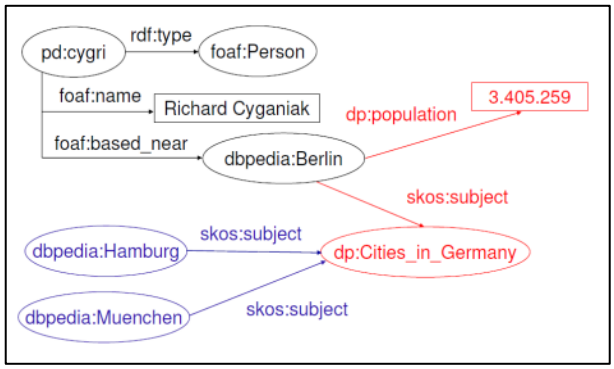


Figura 5 LINK RDF (Bizer, 2010)

1.3. Tecnologías

Existen varias tecnologías que son muy útiles a la hora de vincular y publicar datos entre las más importantes se mencionan las siguientes.

1.3.1. Identificadores Uniformes de Recursos (URI´s)

Una URI puede identificar cualquier cosa, en otras palabras provee un simple y un extensible significado para identificar un recurso (Berners-Lee, et al, 2005). A menudo se confunde el termino URI y URL, la diferencia radica en que un URI puede ser clasificado como un localizador, un nombre o ambos y permite incluir en la dirección URL una subdirección, a diferencia de una URL que se refiere a un subconjunto de URIs que además de la identificación de un recurso, provee un significado de localización del recurso mediante la descripción de su mecanismo de acceso primario. En la Figura a continuación se representan los componentes de una URI en la cual se aprecia la diferencia entre una dirección URL.

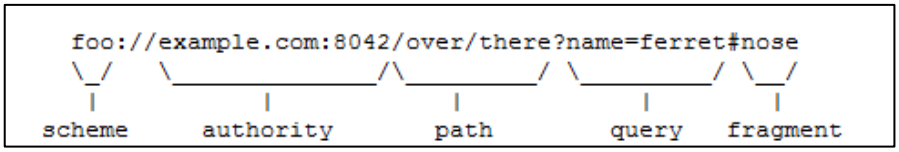


Figura 6 EJEMPLO DE URI

1.3.2. Representación de Datos RDF

RDF o Marco de Descripción de Recursos es el estándar de la W3C para los datos interoperables legibles por máquinas, este modelo se basa en la idea de convertir las declaraciones de los recursos en expresiones en tripletas[26], las mismas que se componen de sujeto, predicado y objeto.

- *Sujeto:* Es el recurso al cual nos estamos refiriendo.
- *Predicado*: Es el recurso que indica lo que estamos definiendo.
- *Objeto:* Puede ser un recurso que puede considerarse el valor definido.

En la tabla III y IV, se ilustra un ejemplo de los componentes de una expresión RDF y su representación en el lenguaje RDF (W3C, 1999).

Leonardo Sarango es autor del recurso http://leodansr.wordpress.com/2010/12/17/instalar-virtuoso-open-source-en-ubuntu/

[26]Una tripleta describe propiedades de un recurso identificado por una URI.

Sujeto (Recurso)	http://leodansr.wordpress.com/2010/12/17/instalar-virtuoso-open-source-en-ubuntu/
Predicado(propiedad)	Autor
Objeto (literal)	“Leonardo Sarango”

Tabla III PARTES DE UNA EXPRESIÓN RDF

RDF también provee una sintaxis basada en XML (llamada RDF/XML), en la Tabla IV observamos cómo queda la expresión de la tabla III representada en RDF/XML.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/">
    <rdf:Description about="http://leodansr.wordpress.com/2010/12/17/instalar-virtuoso-open-source-en-ubuntu/”>
   <s:Autor>Leonardo Sarango</s:Autor>
  </rdf:Description>

</rdf:RDF>
```

Tabla IV RDF EN RDF/XML CORRESPONDIENTE A LA EXPRESIÓN RDF ANTERIOR

Es importante realizar las siguientes explicaciones en cuanto al sujeto y al objeto, y es que ambos se indican mediante URI (Universal Resource Identifier), mientras que el objeto puede ser URI o un literal. El siguiente ejemplo muestra la sintaxis RDF en tres sentencias que indican el nombre de un recurso (Instalar 4Store sobre Ubuntu 10.4), otro recurso (leonardo_sarango) que representa el creador de tal artículo y se incluye una meta sentencia (rdf:type), que indica el tipo del recurso leonardo_sarango (foaf:Person).

```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:dc="http://purl.org/dc/elements/1.1/">

    <rdf:Description
```

```
        rdf:about="http://leodansr.wordpress.com/2011/05/23/intalar-4store-sobre-
        ubtutu-10-4/">
        <dc:title> Intalar 4Store sobre Ubtutu 10.4
        </dc:title>
        <dc:creator
                rdf:resource="http://www.example.org/leonardo_sarango"/>
    </rdf:Description>
    <rdf:Description
        rdf:about="http://www.example.org/leonardo_sarango"/>
        <rdf:type rdf:resource="foaf:Person"/>
        <foaf:name>Leonardo Sarango</foaf:name>
    </rdf:Description>
</rdf:RDF>
```

Como podemos observar esta forma de representación tiene la desventaja que las sentencias pueden ser muy largas y muy repetitivas, por eso es necesario la utilización de Qname (abreviaturas para los URIref´s), espacios de nombres o Namespace usados para agrupar nombres y de esta manera evitar ambigüedad a la hora de usarlos. (W3C, 2004 a), como se ilustra a continuación.

PREFIX rdf: namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
PREFIX dc: namespace URI: http://purl.org/dc/elements/1.1/
PREFIX foaf: URI: http://xmlns.com/foaf/0.1/>

En el artículo (W3C, 2004 b) se menciona que existen algunas características que se deben considerar al realizar un diseño de RDF, las cuales se citan a continuación:

- *Independencia:* Cualquier persona o empresa independiente puede diseñarla.
- *Intercambio:* Una de las características de este estándar es que permite el intercambio de información de forma masiva, esto gracias a la utilización del lenguaje de marcas XML.
- *Escalabilidad:* RDF nos permite representar expresiones basados en un modelo de tripletas, facilitando el manejo y uso de objetos en la búsqueda en grandes volúmenes de información.
- *Las propiedades son recursos:* las propiedades pueden tener sus propias propiedades a la vez, lo que permite ser manipuladas como cualquier otro recurso.
- *Los valores pueden ser recursos:* los valores de las propiedades debe incluir recursos.
- *Las sentencias pueden ser recursos:* Las sentencias deben tener sus propias propiedades.

Pero porque no utilizar solamente XML que también permite definir datos de forma estructurada para que puedan ser compartidos y procesados automáticamente, la respuesta es sencilla y es que XML no permite especificar la semántica de dicha estructura, de una forma formal que pueda ser interpretada por un computador.

1.3.3. XML vs RDF

En la sección 1.3.2 se mencionó que porque no utilizar solamente XML que también permite definir datos de forma estructurada para que puedan ser compartidos y procesados automáticamente. En la siguiente tabla se amplía una comparación más específica entre estas dos tecnologías. (Cámara, 2002).

XML	RDF
No permite especificar la semántica de una estructura.	Permite especificar la semántica para una estructura de una forma formal.
Lenguaje que permite modelar datos	Lenguaje para especificar metadatos
Escalabilidad de datos limitada, debido a que el orden de los datos no es adecuado.	Permite la interoperabilidad entre aplicaciones.
El modelo de datos XML es un árbol etiquetado, hay menos flexibilidad para expresar los metadatos.	El modelo de datos consiste en arcos etiquetados.
Los nodos que se refieren a un XML Schema son localizados dentro del mismo documento, en una localización específica, dentro de la estructura de ese documento.	Los nodos no se encuentran dentro del documento, pero el recurso posee un URI y la mayor parte del tiempo reside fuera de la referencia del documento.
XML tiene un significado que es sintáctico.	Es usado para construir modelos de conocimiento, para las cuales la representación basada sobre una estructura de árboles no es suficiente.

Tabla V XML vs RDF

Mediante RDF podemos describir recursos o metadatos para ser usados en interoperabilidad en el intercambio de datos, posibilidades que con XML no las podremos conseguir.

1.3.4. RDF Schema

Con RDF se definen instancias hablando en Programación Orientada a Objetos, pero para definir las clases debemos utilizar RDF Schema el mismo que es una extensión semántica de RDF que consiste en un lenguaje primitivo de ontologías que proporciona los elementos básicos

para la descripción de vocabularios (W3C, 2004 c). El modelo RDF junto con el RDF Schema es importante, dado que puede utilizarse como un modelo general para expresar metadatos sobre recursos Web. De acuerdo a (W3C, 2004) c, podemos citar algunas características de RDF Schema:

- Los elementos son un conjunto de recursos y propiedades que facilitan la definición de vocabularios.
- Implementa un modelo de datos orientado a objetos.
- La estructura de los RDF Schema contiene: clases, instancias, relaciones, propiedades y dominio.

1.3.5. Propiedades para la Definición de Esquemas.

Una importante característica de los RDFS es la definición de esquemas debido a que permite la extensibilidad en cuanto a la elaboración de nuevos esquemas (vocabularios de recursos para la utilización con RDF). A continuación se mencionan las propiedades para la definición de esquemas..

- *rdfs:subPropertyOf:* Se aplica a las propiedades que pueden ser interpretadas como un subconjunto de otras propiedades, permitiendo que la inferencia entre conceptos.
- *rdf:Class*: Las clases son recursos que denotan conjunto de recursos.
- *rdf:type:* Denota las instancias que tiene una propiedad valorado por una clase, las clases son estructuradas al igual que las propiedades, en un subconjunto jerárquico denotado por la propiedad.
- *rdf:subClassOf*: Se aplica a las instancias que tienen una propiedad valorada por la clase. Las clases poseen una clase-universal llamada rdf:Resource a la que se aplican las mismas reglas que rdfs:subPropertyOf.
- *rdfs:range*: se aplican a propiedades que son valorados por clases. Una propiedad no puede tener más de un valor para rdfs:range.
- *rdfs:domain:* Es utilizada para efectuar chequeos automáticos de verificación.

1.4. Motores de Representación (RDF Data Store)

Actualmente existen varias herramientas que se utilizan como repositorios de almacenamiento para RDF llamados RDF Triple Store System. (W3C, 2011), son

sistemas que nos permiten almacenar e interactuar con tripletas, entre los más importantes y los que se han utilizado se mencionan los siguientes:

1.4.1. 4Store

Es un RDF/SPARQL Store, escrito en C, diseñado para el desarrollo de aplicaciones web semánticas, de esta manera permite hacer gran cantidad de consultas sobre la información almacenada por los usuarios. Según (Garlik, 2009 a), se citan las siguientes características.

- Soporte para datos RDF.
- Disponible bajo GPL.
- Trabaja sobre sistemas operativos basados en UNIX.
- Soporte para SPARQL.
- Posee buen desempeño, escalabilidad y estabilidad sobre los datos.

Entre los proyectos más importantes que utilizan este Data Store se menciona los siguientes según (Garlik , 2009 b)

- FOAF Index: proyecto que tiene millones de usuarios, almacenados en 4store para realizar consultas SPARQL sobre HTTP SPARQL (FOAF, 2010).
- DataPatrol: proyecto encargado de proteger información personal utilizadas en el comercio electrónico, DataPatrol realiza un monitoreo diario emitiendo una alerta al usuario informándole que su información corre algún riesgo (Garlik, 2010).

1.4.2. OpenLinkVirtuoso

Virtuoso es un Data Store híbrido que combina las funcionalidades de los RDBMS[27], ORDBMS, bases de datos virtuales, RDF, XML y aplicaciones web (W3C, 2011 b). Entre las características más relevantes podemos citar las siguientes, en base a lo que se menciona en (OpenLink Software, 2009 a).

[27] http://www.rdbms.org/

- Posee licencia GLP para el producto conocido como OpenLinkVirtuoso[28]y también posee licencia pagada para otras versiones del producto.
- Posee un diccionario de datos en donde se almacena toda la información de los objetos de los usuarios.
- Soporte para SPARQL.
- Soporte para ISQL[29], lenguaje que permite una fácil administración y mantenimiento de la base de datos.
- Trabaja nivel de privilegios y roles, como mecanismos de seguridad.

Su motor de datos provee conexiones a fuentes XML, ODBC, JDBC, ADO.NET y OLE DB.

1.4.3. Sesame

Sesame es un framework de java, creada para el almacenamiento, consulta y razonamiento de con RDF y RDF Schema, puede ser usada como base de datos para RDF y RDFs o como una librería para aplicaciones java que necesiten trabajar con RDF internamente. (Aduna B.V., 2006). Entre las características más relevantes se mencionan:

- Puede ser implementado en sistemas de almacenamiento como base de datos relacionales, sistemas de ficheros e indexadores de palabras claves.
- Soporta para SeRQL[30] y SPARQL.
- Ofrece las herramientas necesarias para analizar, interpretar, consulta y almacenar información.

Es importante señalar que las tripletas en Sesame se almacén en cuádruplas: sujeto, predicado, objeto y contexto.

1.4.4. Oracle Spatial 11g

Oracle Spatial 11g incluye una plataforma abierta de gestión de RDF escalable, segura y confiable, basado en un modelo de datos de grafos, las tripletas RDF se conservan indexadas y la consulta es similar a otros tipos de datos de bases relacionales, el sistema también

[28] *virtuoso.openlinksw.com*
[29] *isql.sourceforge.net*
[30] http://www.openrdf.org/doc/sesame/users/ch06.html

implementa subconjuntos de OWL Full (W3C, 2011 a). A continuación se describen algunas características como se mencionan en (ORACLE, 2009).

- Toda la información puede ser consultada mediante SQL, al que se añade la función embebida SEM_MATCH, que cubre la mayoría de las funcionalidades de SPARQL.
- Introduce un motor de inferencia nativa para razonamientos sobre subconjuntos (extensos) de OWL.

Los datos se almacenan en cuádruplas: sujeto, predicado, objeto y el cuarto elemento que es un identificador único de la tripleta.

1.4.5. AllegroGraph

AllegroGraph es un sistema de carga y consulta de datos RDF, incluye una interface SPARQL y un servidor RDF basado sobre D2RQ[31] (W3C, 2011 b). Ente las características más relevantes se mencionan las siguientes según (Franz Inc., 2011 a).

- Soporta SPARQL.
- Se puede manipular los datos en triple-Store por diferentes interfaces y lenguajes como JAVA, HTTP y LISP[32].
- Soporta RDFS++.
- Soporta Razonador Prolog.
- Incorpora un cliente java mejorado.
- Carga de archivos en el formato Turtle(Lenguaje breve de tripletas RDF).

En la tabla VI se realiza una comparativa entre los RDF triple Store System, esta comparativa se encuentra ampliada en la tabla. VIII (Características de los RDF Triple Store).

[31] http://www4.wiwiss.fu-berlin.de/bizer/d2rq/
[32] http://www.lisp.org/alu/res-lisp

RDF Triple Store Systems	Plataforma	Tecnología	Lenguaje de consulta
4store	Unix, Mac OS X	RDF	SPARQL
Virtuoso	Windows, Mac OS X, Linux y UNIX	RDF, XML	SPARQL, ISQL
Sesame	Windows, Unix, Solaris, Mac OS X eIRIX	RDF, RDFS	SERQL y SPARQL
Oracle	Windows, Linux, Mac OS X, Solaris	RDF, RDFS	SQL
AllegroGraph	Linux, Apple Mac OS X, Windows	RDF, RDFS	SPARQL, Prolog, Razonador RDFS++

Tabla VI COMPARATIVA ENTRE LOS RDF TRIPLE STORE SYSTEM

1.5. Explotación de datos (SPARQL)

SPARQL es un lenguaje de recuperación web y sirve para recuperar recursos electrónicos desde la Web, esto se logra mediante la estructuración de sentencias las cuales contienen operadores y estructuras que son organizadas según normas lógicas definidas por el lenguaje. Específicamente SPARQL hace posible la recuperación de RDF y RDFS (W3C, 2007), esto debido a que se basa en la comparación de patrones gráficos los cuales están basados en tripletas como las usadas en RDF pero con la opción de que posee una variable de consulta que remplaza el sujeto, objeto y predicado en el término RDF.

Como pudimos observar en la Figura 2, esta tecnología no está contemplada, sin embargo en el artículo (Herman, 2007), se hace una especificación adicional, sobre las capas de la WS, mostrando a SPARQL como un componente importante dentro de la recuperación de información a través de RDF, esto se ilustra en la Figura a continuación citada.

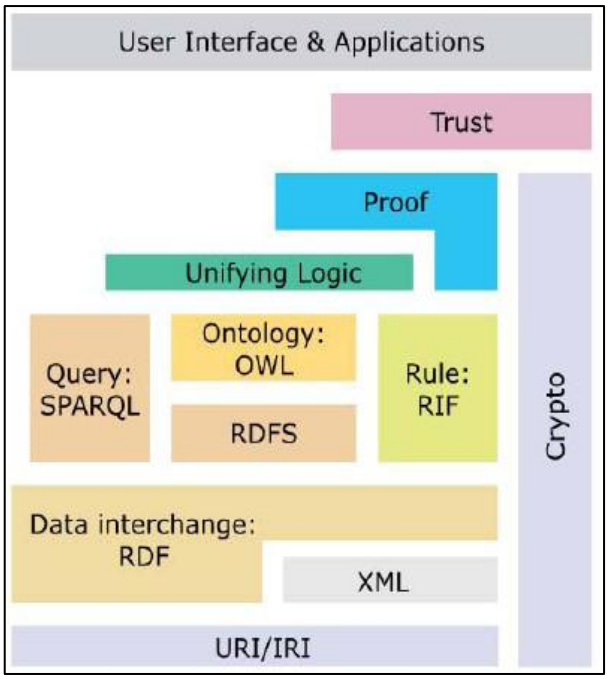


Figura 7 THE LAYERCAKE (W3C[33])

En cuanto a la sintaxis de SPARQL, la podemos observar en el siguiente ejemplo que muestra una consulta SPARQL hecha sobre DBPEDIA[34] , en donde se muestra el nombre, pagina web, de las universidades de la ciudad de Madrid, España.

SPARQL:

```
PREFIX dbpediar: <http://dbpedia.org/resource/>
PREFIX dbp-ont: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?name ?ciudad ?homepage
WHERE {
?university foaf:name ?name.
?university foaf:homepage ?homepage.
?university rdf:type <http://dbpedia.org/ontology/University>.
?university dbp-ont:city ?ciudad.
FILTER regex(?ciudad, "Madrid" , "i")
}
order by ?name
```

Results: Browse Go! Reset

SPARQL results:

name	ciudad	homepage
"Autonomous University of Madrid"@en	:Madrid	<http://www.uam.es>
"CEU San Pablo University"@en	:Madrid	<http://www.uspceu.com>
"Comillas Pontifical University"@en	:Madrid	<http://www.upcomillas.es>
"Complutense University of Madrid"@en	:Madrid	<http://www.ucm.es/>
"EAE Business School"@en	:Madrid	<http://www.eae.es>
"ESCP Europe"@en	:Madrid	<http://www.escpeurope.eu>
"ESERP"@en	:Madrid	<http://www.eserp.com/en>
"ESERP School of Business and Social Sciences"@en	:Madrid	<http://www.eserp.com/en>
"Escuela Superior de Música Reina Sofía"@en	:Madrid	<http://www.escuelasuperiordemusicareinasofia.es/>
"European Institute of Design"@en	:Madrid	<http://www.ied.it/>
"European School of Economics"@en	:Madrid	<http://www.eselondon.ac.uk>
"Francisco de Vitoria University"@en	:Madrid	<http://www.ufv.es>
"IE Business School"@en	:Madrid	<http://www.ie.edu/business>

Figura 8 CONSULTA SPARQL SOBRE DBPEDIA

[33] http://www.w3.org/2007/Talks/0423-Stavanger-IH/Slides.pdf
[34] www.dbpedia.org/snorql, es un conjunto de datos extraído de Wikipedia.

Actualmente esta tecnología está siendo considerada como una recomendación del W3C para crear un lenguaje de consulta dentro de la Web semántica que como lo analizamos anteriormente está implementada en varios lenguajes y bases de datos como Oracle[35], Virtuoso[36], AllegroGraph[37], Sesame[38], entre otros.

1.6. Visualización y/o Consumo (Special Browser)

Los navegadores especiales son medios que me permiten directamente visualizar o explorar datos RDF, estos trabajan directamente en su navegador web habitual (W3C, 2011 a). A continuación se citan algunos de ellos:

1.6.1. /facet

Facet es un navegador genérico para repositorios semánticos heterogéneos, trabaja sobre cualquier conjunto de datos RDFS sin ninguna configuración adicional, entre sus características más importantes según (Hildebrand, 2006) citamos:

- Selecciona y navega por las facetas de los recursos que pueden ser de cualquier tipo.
- Hace selecciones basadas en las propiedades de otros tipos relacionados semánticamente.
- Permite la inclusión de una pantalla de opciones de facetas específicas.

1.6.2. Lena

Lena es sinónimo de un navegador basado en objetivos. Un objetivo representa una visión particular sobre los datos RDF, y es descrito por el vocabulario de pantalla Fresnel[39]. Lena permite ver los datos RDF en un navegador, presentados de acuerdo a la descripción del objetivo que se proporciones. Según (UNIVESITAT KOBLENZ- LANDAU, 2008), se citan las siguientes características:

[35] www.oracle.com
[36] virtuoso.openlinksw.com
[37] www.franz.com/agraph/allegrograph
[38] http://www.openrdf.org/
[39] http://www.w3.org/2005/04/fresnel-info/

- Soporta SPARQL
- Provee una extensión para soportar selectores Fresnel SPARQL.
- Se puede acceder a los datos que serán procesados por Elena mediante un repositorio HTTP Sesame.

1.6.3. Jena2

Jena2 es un framework desarrollado en java para construir aplicaciones para la web semántica, provee un ambiente de programación para RDF, RDFS y OWL, además provee un motor de inferencias basado en reglas, lo que la convierte en un modelo ideal para cualquier proceso automatizado de creación de contenidos destinados a ser usados en canales de información. Entre sus características más importantes podemos citar las siguientes según (Reynolds, 2010).

- Posee un API para RDF, la misma que soporta la creación, manipulación y consulta de grafos RDF.
- Permite realizar lectura y escritura de documentos en formato RDF/XML, N3 y N-Triples.
- Posee un API para OWL
- Almacenamiento persistente y en memoria.
- Posee RDQL, el mismo que es un lenguaje de consulta de RDF.

1.6.4. OpenLink RDF Browser

El OpenLink RDF Browser[40] es un navegador de datos RDF implementado con una interfaz de un manejador de navegador HTML, para demostrar y explotar el poder de los datos enlazados RDF desreferenciables, ha sido implementado dentro de OpenLinked Data Explorer[41] y ha sido construido con OpenLink AJAX Toolkit (Semantic Web, 2009).

1.6.5. Disco HiperData Browser

Disco Hyperdata browser es un simple browser para navegar por la WS. El navegador hace que toda la información pueda ser encontrada en la WS sobre un recurso específico como una página HTML. Esta descripción de recursos contiene hipervínculos que permite que se pueda

[40] http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html
[41] http://ode.openlinksw.com/

navegar entre recursos, mientras se navegada de recurso a recurso, el navegador recupera dinámicamente la información de HTTP URIs desreferenciables (Bizer & GauB, 2007). Entre sus principales características se mencionan las siguientes:

- El navegador es una aplicación que corre del lado del servidor y no necesita instalar ningún componente en el computador del usuario.
- Permite navegar sobre un conjunto de fuentes de datos no enlazados.
- Se puede utilizar el navegador de Disco para permitir que los internautas puedan navegar por los datos RDF que el usuario haya publicado en la web.

1.7. Vocabularios para Linked Data

Actualmente existen varios vocabularios consensuados que permiten la descripción de datos los cuales deben reutilizarse y no reinventar, para garantizar que los datos puedan ser consumidos por aplicaciones en la web facilitando la interoperabilidad, entre los más importantes y utilizados en el desarrollo de esta investigación se mencionan los siguientes:

1.7.1. Dublin Core Metadata Iniciative(DCMI)

Dublin Core es un vocabulario general de metadatos que define 15 definiciones semánticas descriptivas que son necesarias para describir, identificar, procesar, encontrar y recuperar un recurso en la web, esto es importante debido a que si se estandarizara el uso de este vocabulario se lograría facilitar la interoperabilidad no solo entre OER/OCW si no a nivel de cualquier recurso en la Web.

Los elementos de Dublin Core poseen etiquetas descriptivas, donde cada elemento es opcional, puede repetirse y pueden aparecer en cualquier orden. Podemos clasificar el conjunto de elementos Dublin Core en 3 grupos que indican la clase o el ámbito de la información que contienen:

- Elementos relacionados con el recurso:
  - Title (título)
  - Subject (tema)
  - Description (descripción)

- Source (fuente)
- Languaje (lenguaje)
- Relation (relación)
- Coverage (cobertura).

- Elementos relacionados a la propiedad intelectual:
  - Creator (autor)
  - Publisher (editor) y, otras colaboraciones
  - Contributor (otros autores/colaboradores)
  - Rights (derechos).
- Elementos relacionados con la instanciación del recurso.
  - Date (fecha)
  - Type (tipo de recurso)
  - Format (formato)
  - Identifier (identificador)

1.7.2. FOAF (Friend to Friend)

Según lo especificado en (Brickley & Libby, 2010), FOAF describe el mundo con ideas simples inspiradas en la web, cuyo objetivo es ayudar a las personas en la creación de documentos de perfiles FOAF[42] para sí mismos. El principal propósito de un perfil de documentos es permitir a una persona especificar una expresión básica de miembro de una comunidad, estos documentos pueden ser producidos usando un vocabulario RDF que es expresado en la ontología de FOAF y también puede ser procesado por herramientas RDFS.

FOAF permite la creación de una entidad Persona (Person) para representar una persona para quien el documento de perfil FOAF es creado. Esto provee una URI que permite a la persona expresar su identidad única a nivel mundial, esta información puede incluir metadatos como nombre, email, número telefónico, Nick, etc. Luego de definir la información de una persona, se puede usar la propiedad knows, que especifica que una persona puede asociarse con otras personas. Permitiendo de esta manera la especificación de grupos que pueden tener uno o más de una persona como miembros y organizaciones para representar instituciones o sociedades.

[42] http://xmlns.com/foaf/spec/index.rdf

A continuación se listan los principales términos agrupados en categorías. (Brickley & Libby, 2010).

- Core: describen características de personas y grupos sociales que son independientes del tiempo y de la tecnología, dentro de esta categoría los términos más utilizados son Person, name, title, familyName, age, Organization, Group.
- Social Web: Adicionalmente a los términos Core, existe un numero de términos para usarlos cuando se describe cuentas de internet y otras actividades basadas en la web, dentro de esta categoría los términos más frecuentemente utilizados son Nick, mbox, homepage, weblog, topic_interest.
- Linked Data utilities: FOAF es un intento para usar la web con el objetivo de integrar la información factual con información en documentos orientados a humanos como videos, libros, hojas de cálculo, modelos 3d, así como la información que aún permanece en la cabeza de las personas.

1.7.3. Creative Commons (CCREAL)

En (Creative Commons, 2008), se menciona que CC es un proyecto que ofrece licencias libres y herramientas legales para permitir que los creadores de contenidos pueda licenciar sus trabajos, para que otros puedan compartir, re mezclar y utilizarlos comercialmente. Un RDF Schema ha sido definido para especificar las propiedades de una licencia pueda implicar[43] En su trabajo (Newman, 2010), se presenta una figura (Figura 9), que representa las licencia y sus propiedades.

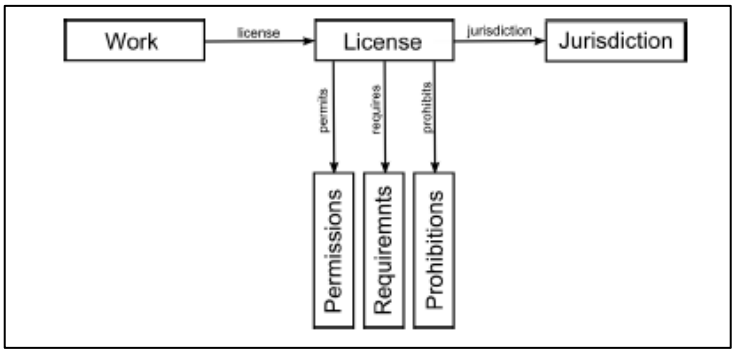


Figura 9 CREATIVE COMMONS OVERVIEW

[43] http://creativecommons.org/schema.rdf

Como se muestra en la Figura 9, y según (Creative Commons, 2008), se pueden citar los siguientes elementos de las clases derivadas de licencia.

- Permissions
  - Reproduction
  - Distribution
  - Derivate Works
  - Sharing
- Requeriments
  - Notice
  - Attribution
  - Share Alike
  - Source Code
  - Copyleft
  - Lasser Copyleft
- Prohibitions
  - Commercial Use
  - High Income Nation Use

1.7.4. AIISO (Academic Institution Internal Strcture Ontology).

Según (Styles & Shabir, 2008), AIISO provee clases y propiedades para describir la estructura organizacional interna de una institución académica. AIISO está diseñado para trabajar en colaboración con Participation[44], FOAF[45] y aiiso-roles[46] para describir los roles juegan las personas dentro de una institución. En esta investigación sin embargo AIISO será utilizado únicamente para describir metadatos de una institución OCW como tal.

AIISO define 15 clases y 10 propiedades de las cuales destacamos las clases que se utilizaran en esta investigación, clases como Course, Knowledge y propiedades como description e ispartof.

[44] http://purl.org/vocab/participation/schema
[45] http://xmlns.com/foaf/0.1/
[46] http://purl.org/vocab/aiiso-roles/schema

1.8. Fuentes de Datos OCW/OER

Existen diversas fuentes de Recursos Educativos de Aprendizaje, sin embargo, el conjunto más amplio, diverso y organizado sin duda alguna son los llamados Cursos Abiertos ofrecidos por las diversas Universidades del Mundo, a continuación se detallan ampliamente estos dos términos.

1.8.1. OpenCourseWare

Las universidades son el repositorio del conocimiento humano y el núcleo principal de expansión del conocimiento científico, en base a este pensamiento, si las instituciones públicas publican los contenidos de los cursos que se dictan a sus estudiantes, en páginas web públicas, se contribuye a la materialización de la educación del siglo XXI, una educación a la cual puedan acceder todas las personas del mundo, pudiendo intercambiar información de calidad entre estudiantes y profesores.

En (OCW Consortium, 2011), se menciona que un OCW es una publicación digital libre y abierta de alta calidad de nivel universitario, estos materiales están organizados como cursos, y frecuentemente incluyen materiales de planificación de cursos y herramientas de evaluación, así como contenido temático. OpenCourseWare es libre y tiene licencia abierta, accesible a cualquiera, en cualquier momento a través de internet. Este proyecto se puso en marcha en el 2002, impulsada por el prestigioso Instituto Tecnológico de Massachusetts (MIT), en colaboración con la 'Fundación William and Flora Hewlett' y la 'Fundación Andrew W. Mellon'.

Cuando una institución deciden contribuir a los proyectos OpenCourseWare, se comprometen a desarrollar y mantener con sus recursos de Internet los materiales educativos de acceso abierto a los usuarios del mundo, estos materiales se publican con licencias abiertas, generalmente con licencia Attribution-NonCommercial-ShareAlike 3.0 Unported[47]. En concreto, un sitio OCW acogido a la licencia Creative Commons[48] conlleva los siguientes derechos y deberes:

- Cede a los usuarios el derecho de usar y distribuir los materiales.

[47] http://creativecommons.org/licenses/by-nc-sa/3.0/
[48] http://creativecommons.org

- Permite a los usuarios crear trabajos derivados mediante su edición, traducción, o añadiéndolos y combinándolos a otros materiales de otros autores.

Pero también obliga a los usuarios a respetar ciertos requisitos en su utilización:

- Restringe a propósitos educativos no comerciales.
- Exige que los materiales utilizados sean atribuidos a la Institución y a los autores originales.
- Requiere que si el original o los materiales derivados son de nuevo publicados o distribuidos, éstos sean ofrecidos libremente a otros usuarios bajo idénticas condiciones de licencia de uso.

Según lo expuesto en (OCW Consortium, 2011), se citan las principales características de un OCW:

- Se ofrecen libremente y son accesibles universalmente en la red.
- No se limita su difusión por temas de propiedad intelectual.
- Se permite: uso, reúso, adaptación y distribución por otros bajo restricciones.

Además, es importante mencionar que un OCW:

- NO es un servicio de educación a distancia.
- NO brinda derechos a reclamar acreditación.
- La institución NO está obligada a disponer medios de interacción entre profesores y estudiantes.

Según lo mencionado en (MITOPENCOURSEWARE, 2011), actualmente existen más de 2000 cursos con acceso libre en este repositorio, en la Figura 10, se plasma los tipos de usuarios y su nivel de uso del proyecto OCW.

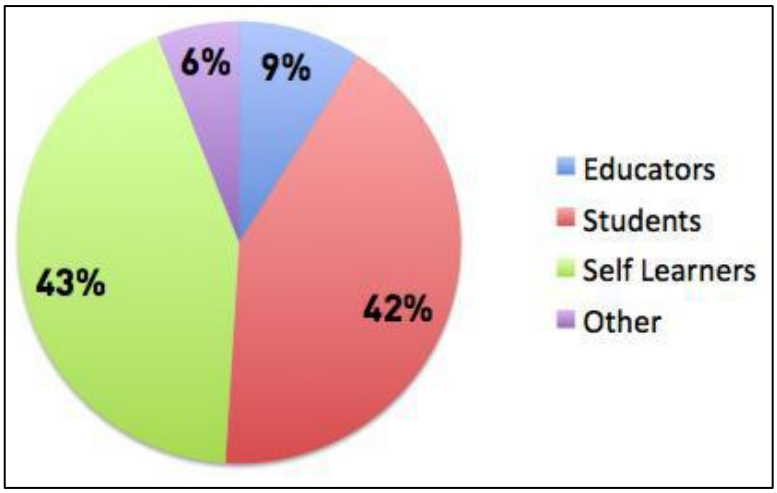


Figura 10 USUARIOS DE OCW (Valkenburg, 2011)

Como se mencionó anteriormente los materiales de un OCW se organizan dentro de cursos, la estructura de dichos cursos se organizan formando un eje constituido por un conjunto de categorías que integran los contenidos del curso, como se muestra en la tabla VII, categorías que no necesariamente son un requisito, esto más bien depende del autor del curso, aunque personalmente pienso que debería ser un requisito para evitar barreras en cuanto a la interoperabilidad entre cursos OCW.

CATEGORÍAS DE UN CURSO OCW
Syllabus
Calendar
Reading
Lecture Notes
Labs
Exams
Study Materials
Image Gallery
Project Video
Projects
Discussion Group

Tabla VII Categorías de un Curso OCW

En el 2007 surge en España un proyecto llamado OCW Universia, la mayor red iberoamericana de colaboración universitaria, que integra a 1.216 universidades socias de 23 países de Iberoamérica (representan 14 millones de estudiantes y profesores universitarios), definición según (Universia, 2011).

1.9. Open Educational Resource (OER)

En (UNESCO, 2011), se dice que los OERs son materiales de aprendizaje, enseñanza o búsqueda que están en un dominio público o relacionado con una licencia de propiedad intelectual que permita su libre uso, adaptación y distribución. Personalmente lo defino como un objeto digital que se encuentra en la web, que puede ser accedido y usado con propósitos educativos. Los beneficios que estos materiales brindan tanto corporativamente como personalmente son varios entre los cuales podemos citar los siguientes:

- Incrementa la visibilidad institucional.
- Minimización de costos.
- Usar-Adaptar-Remezcla.
- Llegar a nuevos grupos de personas que no tienen acceso a una educación superior.
- Mejora la calidad de los materiales de aprendizaje
- Posibilidades de colaboración con instituciones académicas de todo el mundo.

Según (Tóvar, et al, 2011), Un OER es el resultado de un ciclo completo de enseñanza, producto de los recursos que implica una serie de etapas sucesivas, cada una de ellas lleva a cabo un propósito específico, en este ciclo de producción, deben considerarse las necesidades de la comunidad educativa: profesores y estudiantes. Adicionalmente a esto también se menciona que la construcción de un OER involucra uso, reutilización y creación de un recurso digital como videos, texto, presentaciones y el desarrollo de metadatos asociados a estos recursos que permitan el almacenamiento y catalogación en repositorios, así como proporcionar información sobre los derechos de autor que se pueden utilizar para determinar si el contenido se puede consultar, utilizar o editar.

(Nelson & Romero, 2010), Mencionan que la concepción de OER se puede resumir en la siguiente Figura:

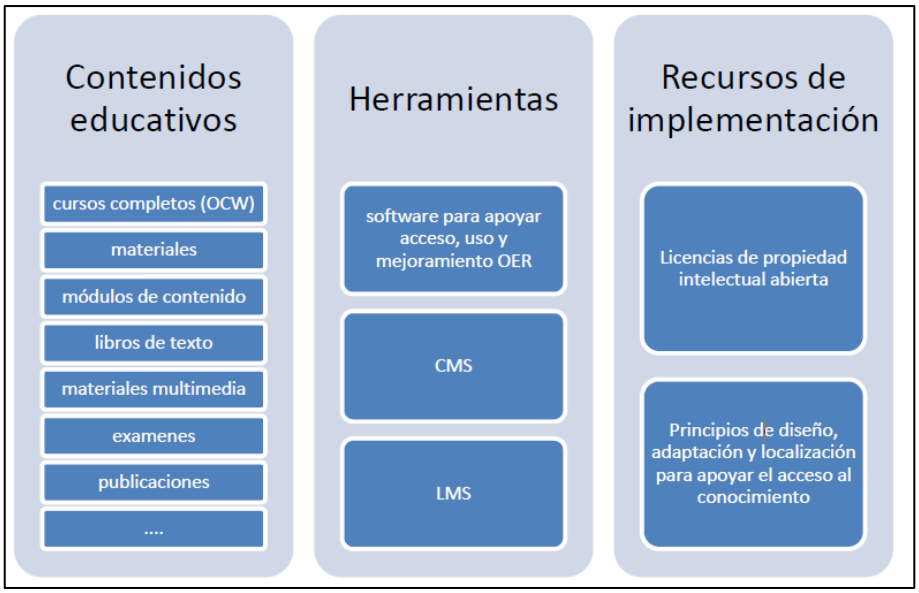


Figura 11 OER Y ELEMENTOS (Nelson & Romero, 2010)

1.10. Trabajos Relacionados

El mejor ejemplo de la adopción y la aplicación de los principios de Linked Data ha sido el proyecto Linked Open Data[49] cuyo objetivo es impulsar la web de datos mediante la identificación de conjuntos de datos existentes que estén disponibles bajo licencias abiertas, convirtiéndolas en RDF bajo los principios de Linked Data y publicarlos en la web, En la Figura 12 se presenta el crecimiento del proyecto representada por distintos conjuntos de datos de diversos tipos, organizados mediante colores por dominios, los mismos que están conectados entre sí, formando así la nube de Linked Data (W3C, 2011 c).

[49]http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

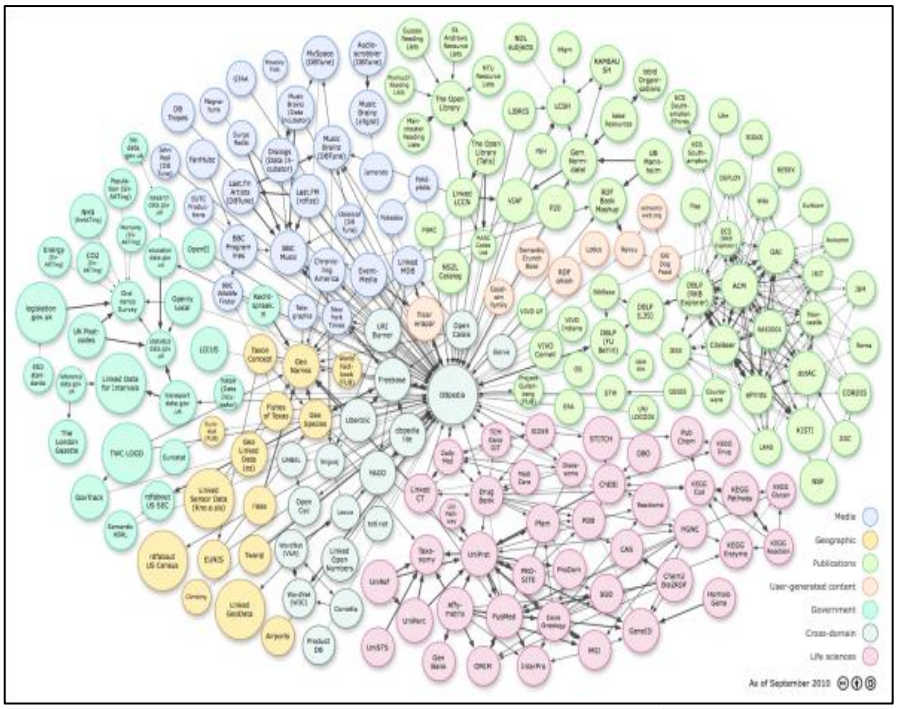


Figura 12 NUBE DE LINKED DATA[50]

En base a esta figura, la investigación tiene por objetivo publicar datos Universitarios enlazados observando los principios de Linked Data, cuyo objetivo más ambicioso es contribuir a algún trabajo futuro que permita la inclusión de nuestro proyecto en la Nube de Linked Data.

1.10.1. CourseWare

Courseware[51] es un repositorio semántico que contiene y publica datos vinculados RDF, los datos presentes son provistos desde el proyecto ReSIST[52], el mismo que posee más de 50 millones de ítems, recomendados para todos los involucrados con la enseñanza de los temas relacionados.

1.10.2. Locah Project

Organismos como Mimas[53] y UKOLN[54] están trabajando juntos en un proyecto interesante financiado por JISC[55] (organización que promueve el uso el uso de tecnologías digitales en las

[50] http://en.wikipedia.org/wiki/File:Lod-datasets_2010-09-22_colored.png
[51] http://courseware.rkbexplorer.com/
[52] http://www.resist-noe.org/
[53] http://mimas.ac.uk/

universidades y colegios (JISC, 2011)), para hacer Archives Hub[56] (proyecto que provee acceso de miles de archivos del Reino Unido de más de 180 instituciones de todo el país, convirtiéndose en una manera eficaz de descubrir fuentes de información únicas que apoyen la investigación (Archives Hub, (n.d)) y Copac[57] (catálogo de una colección de libre acceso a los catálogos en línea de las principales librerías universitarias Nacionales (Copac, 2011)), para que los datos estén disponibles como una estructura Linked Data, en beneficio de la educación y la investigación. Se desea poner archivos y datos bibliográficos en el corazón de la web de Linked Data para habilitar nuevos links entre diversas fuentes de contenidos diversos para permitir la exploración libre y flexible de datos, para que los investigadores puedan crear nuevas conexiones entre sujetos, personas, organizaciones y lugares (LOCAH Project, 2011).

1.10.3. MitOpenCourseWare RDFizer

RDFizer es una herramienta de utilidad para convertir los metadatos contenidos en el sitio web Mit OpenCourseWare[58] a una representación RDF, de las ontologías IEEE LOM[59], RDFizer requiere una máquina virtual de java y apache maven[60] instalada en la máquina del usuario (SMILE, 2007).

[54] http://www.ukoln.ac.uk/
[55] http://www.jisc.ac.uk/
[56] http://archiveshub.ac.uk
[57] http://copac.ac.uk/about/
[58] http://ocw.mit.edu/
[59] http://projects.ischool.washington.edu/sasutton/IEEE1484.html
[60] http://maven.apache.org/

2.0. RDF Triple Store System y APIS Comparativa

Pedid, y se os dará; buscad, y hallaréis

Evangelio según San Mateo 7:7

2.1. Introducción

En el presente capítulo se presenta un estudio detallado de las características de varios RDF Triple Store System instalados, configurados y testeados para el desarrollo de este proyecto. RDF Triple Store System es un nombre otorgado a los sistemas que nos permiten un almacenamiento persistente y la manipulación de RDF, se estudiarán Triple Store como Oracle, Virtuoso, Sesame, AllegroGraph, 4store y Jena, esto con el objetivo de realizar una comparativa entre cada uno de ellos, sus diferencias, ventajas y desventajas en un determinado punto.

Para el desarrollo de este capítulo se analizó los tutoriales, manuales de usuario y manuales de desarrolladores de los gestores RDF mencionados anteriormente, con el objetivo de realizar un breve estudio sobre las características principales de cada uno de ellos, enfocándonos principalmente en características como entorno, arquitectura, interacción, almacenamiento, tratamiento de la información, consumo de la información, inferencia. Además se citan algunas investigaciones que tratan sobre aspectos no probados en esta investigación.

En los anexos de esta investigación, se adjuntan informes técnicos de las instalaciones, configuraciones y pruebas realizadas en lo relacionado a los RDF Triple Store System.

2.2. RDF Triple Store System

Mientras que pequeñas cantidades de datos RDF pueden ser manejados y administrados eficientemente en la memoria principal del computador, cantidades grandes de datos RDF requieren un sistema de almacenamiento persistente. En el trabajo de (Haslhofer, et al, 2011), se menciona que un almacén de RDF son bases de datos construidas con el propósito de almacenar y recuperar cualquier tipo de datos expresados en RDF.

En el mismo documento citado anteriormente se realiza una clasificación que para mi opinión es la más adecuada acerca de los tipos de almacenes de RDF basados en su arquitectura, estos se clasifican en:

- *Almacenes nativos:* implementan un completo motor de base de datos que es optimizado para procesar y operar RDF independientemente de cualquier otro Sistema de Administración de Base de Datos, los datos se almacenan directamente sobre archivos de sistema.

- *Almacenes de respaldo DBMS*: hacen uso del almacenamiento y la funcionalidad de recuperación de bases de datos relacionales, y debido a que están respaldados por DBMS relacionales pueden aplicar diferentes tipos de modelos de almacenamiento para representar el modelo RDF en el esquema relacional subyacente.
- *RDF wrappers:* es un ligero componente de software que se establece en la parte superior de un origen de datos y expone a los datos almacenados en el mismo como RDF sin afectar a las estructuras de almacenamiento existente.
- *Almacenes híbridos*: soportan ambos estilos de arquitectura (nativos y de respaldo DBMS).

De acuerdo a esta clasificación se citan los siguientes RDF Triple Store System.

2.2.1. Sesame

Sesame es un framework RDF, libre desarrollado en Java que soporta inferencia y búsqueda sobre RDF Schema y OWL y de acuerdo a la clasificación encaja dentro de los RDF wrappers por lo que pueden trabajar sobre varias base de datos relacionales, ofrece una API de acceso a los datos que soporta comunicación local y remota a través de RESTfull HTTP, maneja varios lenguajes de consulta entre ellos SeRQL y SPARQL. En base a lo analizado en (Sesame, 2010) se pueden citar las siguientes características:

- *Entorno:* para el uso de las bibliotecas de programación se requiere JRE (Java Runtime Environment), SLF4J (Simple Logging Facade for Java) y para el uso de Sesame como servidor un contenedor de Java Servlet, se recomienda el uso de Apache Tomcat 7.0.
- *Arquitectura:* posee subsistemas como
    - *RDF Model*: que define interfaces e implementación de todas las entidades básicas RDF (URL, nodo blanco, literales y sentencias).
    - *RDF I/O*: que consiste en un conjunto de parsers y escritores para varios formatos RDF.
    - *Sail Repository:* que es una API de bajo nivel para almacenamiento de RDF e inferenciadores.
    - *Repository API:* que es una API de alto nivel que ofrece un gran número de métodos orientados al desarrollador para el manejo de datos RDF.
    - *Http Server:* consiste en una serie de Servlets en Java que implementan un protocolo para acceder a los repositorios de Sesame a través de HTTP.

- *Interacción:* la interfaz de usuario proporcionada por Sesame es en modo consola y a través de la aplicación web Client que se ejecuta utilizando Apache Tomcat.
- *Almacenamiento:* existen cuatro modos de almacenamiento, los cuales se mencionan a continuación:
    - *En memoria:* configuración más simple y rápida de almacenamiento, simplemente almacena los datos RDF en la memoria principal.
    - *En repositorio nativo:* no mantiene los datos en memoria sino los guarda directamente en disco en formato binario optimizado para almacenamiento comprimido y recuperación rápida.
    - *En base de datos:* a través de Sail para bases de datos MySQL, PostgreSQL y Oracle 11g.
    - *Repositorio remoto:* repositorio que las veces de proxy para otro repositorio en un servidor Sesame.
- *Tratamiento de información:* soporta carga masiva de datos, así como también la inserción y borrado de sentencias individuales a través de la API mediante métodos como createStatement, add o remove de clases como RepositoryConnection, pero no especifica la manipulación directa de esquemas. Además soporta formatos de escritura N3, N-Triple, RDF/XML, TRIG, TRIX y Turtle, RDF, RDFS y OWL. Según (W3C, 2011 b), Sesame ha sido testeado hasta con 70 millones de tripletas.
- *Inferencia:* (Torre, 2009), menciona que Sesame admite razonamientos a nivel de RDFS, utilizando un algoritmo forwars-chaining. Además admite jerarquía de tipos que añade la posibilidad de calcular de herencias directas, mediante los métodos directSubClasssOf y directSubPropertyOf.
- *Consumo de Información:* Sesame soporta SPARQL y SeRQL su propio lenguaje de consultas.

En esta investigación se ha utilizado el conector de Sesame para Java y ha sido testeado con SPARQL, MySQL y Oracle 11g, obteniendo mejores resultados al trabajar con Oracle.

2.2.2. AllegroGraph

Según (Franz Inc, 2011 b), AllegroGraph es una base de datos comercial, desarrollada por Franz Inc, para almacenar y consultar datos RDF. Puede ser implementada como un servidor de base de datos independiente y ofrece interfaces para acceso remoto donde la comunicación entre el Servidor AllegroGraph y el proceso cliente es realizado a través de HTTP.

AllegroGraph está disponible para Windows, Linux y Mac OS X para plataformas de 32 bits y 64 bits. La versión libre está limitada al almacenamiento de menos de 50 millones de tripletas, la edición de desarrollo soporta hasta 600 millones de tripletas y la edición Enterprise es limitada únicamente por el hardware que se posea. De acuerdo a nuestra clasificación encaja dentro de los almacenes nativos. En base a lo analizado en (Franz Inc., 2011 b), se citan a continuación las siguientes características:

- *Arquitectura:* AllegroGraph proporciona una arquitectura de protocolo REST, esencialmente es un súper conjunto de clientes Sesame HTTP. En su nivel más bajo se encuentra el código ejecutable en los sistemas soportados por la aplicación, sobre esta capa se encuentra la base de datos, en otra capa se encuentran las API en java o en Lisp utilizadas para la manipulación de información, en otra capa se localizan los razonadores por encima de las API y de la base de datos, encargados de recibir las consultas en su capa superior, la cual procesa las solicitudes de búsqueda desde un servidor, aplicaciones locales o de acceso remoto.
- *Interacción: p*osee un navegador AGwebWiew Server, la misma que es una interfaz para explorar, almacenar, consultar y administrar tripletas. También se puede interactuar mediante RESTfull HTTP para añadir y borrar sentencias, así como para hacer búsquedas (SPARQL y Prolog). (Torre, 2009) menciona que también permite interacción en modo Lisp, a través de TopBraid Composer y TacerPorter.
- *Extensibilidad:* soporta un protocolo cliente Sesame HTTP, para interactuar a través del HTTPClient de Sesame 2.0, permite conexión con TopBraid Composer para modelar y editar ontologías y soporta integración con RacerPro y RacerPorter.
- *Almacenamiento:* al ser un almacén nativo no utiliza base de datos relacionales, en cambio utiliza archivos con sentencias simples (N-triple) para el almacenamiento de los datos, permitiendo rapidez para la carga y actualización masiva de datos. Según (Heymans, et al., 2007), AllegroGraph puede cargar datos RDF a una velocidad superior a las 10000 sentencias por segundo. En cuanto a la indización AllegroGraph construye índices para acelerar las búsquedas.
- *Tratamiento de información: s*oporta carga masiva de datos mediante funciones como load-ntriple, así como también la inserción y borrado de sentencias individuales a través de add-triple y delete-triple, además soporta formatos de entrada RDF/XML, N-Triple y Trix y de salida RDF/XML y N3.

- *Consumo de Información:* soporta SPARQL y utiliza twinql[61]
- *Inferencia:* soporta inferencia RDF++ y soporta reglas en Prolog.

En esta investigación se ha instalado la versión Libre de Allegro Graph sobre una máquina virtual con Linux y se ha realizado operaciones DDL y DML sobre las tripletas RDF, con Java, obteniendo excelentes resultados tanto en procesamiento, inserción, visualización y consumo de RDF.

2.2.3. OpenLink Virtuoso

OpenLink Virtuoso[62] es una solución de almacenamiento híbrido para una amplia gama de modelos de datos incluyendo datos relacionales, RDF, XML y documentos de texto libre. Virtuoso ha ganado un interés significativo desde que es usado para alojar importantes conjuntos de Linked Data como por ejemplo DBpedia[63]. Virtuoso es ofertado como una versión open-source, para propósitos comerciales existen varios modelos de licencia, se puede implementar como un servidor en las principales plataformas. En base a lo analizado en (OpenLink Software, 2009 a), se citan las siguientes características:

- *Arquitectura:* posee varios módulos como:
  - *Data Access Drivers:* Es el punto de entrada a los servicios del motor.
  - Security Manager: módulo responsable de proteger los datos y su transmisión.
  - *Query Manager:* provee análisis léxico/sintáctico de las consultas.
  - Meta Data Manager: proporciona al Query Manager información respecto de las entidades sobre las que trabajara el plan de ejecución.
  - *Transaction Manager:* Garantiza atomicidad, consistencia, aislamiento y durabilidad en las transacciones.
  - *Concurrency Manager:* Permite que los clientes y servicios puedan abrir múltiples sesiones concurrentes.
  - *Load Data I/O Manager:* maneja la lectura y escritura desde fuentes de datos externas.
  - *Replication Manager:* Controla la migración de datos, su transformación y la sincronización a través de dos o más servidores.

[61] http://www.holygoat.co.uk/projects/twinql/
[62] http://virtuoso.openlinksw.com/
[63] www.dbpedia.org

- *Extensibilidad:* no proporciona ninguna API para que sus métodos sean invocados, sino funciones y procedimientos almacenados invocados a través de la implementación de estándares como ODBC, JDBC, OLE-DB, ADO.NET, XMLA.
- *Interacción:* Posee una interfaz web accesible para la administración, y para el uso como cliente proporciona un intérprete SQL interactivo (ISQL).
- *Almacenamiento:* Según (Erlin & Mikhailo, 2008), Virtuoso permite almacenamiento propietario en el sistema de archivos, así como en gestores de base de datos, gracias a sus características como servidor multiprotocolo que provee diferentes API para acceder a una base relacional.
- *Tratamiento de información:* Soporta carga masiva de datos y carga de sentencias individuales, para este último se emplea una extensión de SPARQL denominada SPARUL, que permite inserción, actualización y eliminación (OpenLink Software, 2009 a). Según (W3C, 2011 b), Virtuoso ha sido testeado hasta con 15.4 Billones de tripletas.
- *Inferencia*: Según (Torre, 2009), Virtuoso SPARQL infiere sentencias que no han sido almacenadas físicamente, utilizando un mecanismo backward-chaining. Utiliza una base de reglas para sus razonamientos.
- *Consultas:* Soporta SPARQL y SPARUL.

En esta investigación se ha instalado Virtuoso Open Source sobre Ubuntu, se realizó operaciones DML y DDL sobre las tripletas RDF, obteniendo excelentes resultados.

2.2.4. Oracle Spatial 11g

El módulo Spatial de Oracle 11g permite el soporte para la administración de datos RDF. Oracle es un producto comercial, pero es libre para propósitos no comerciales. Se despliega como un servidor de base de datos y provee interfaces de programación para SQL (SQLplus, PL/SQL) y aplicaciones basadas en java (Jena/Sesame y JDBC), en esta investigación se ha probado con Jena y Sesame. Dentro de nuestra clasificación encaja dentro de los almacenes de respaldo DBMS al ser una base de datos relacional. En base a lo analizado en (ORACLE, 2010) se pueden citar las siguientes características:

- *Arquitectura:* En la figura 13 se muestra la arquitectura de Oracle 11g donde resaltan los siguientes componentes:
    - *Store:* permite el almacenamiento, carga y la aplicación de operaciones DML.

- *Infer:* permite hacer inferencia OWL, RDF/S. sobre la base de datos, además nos permite definir reglas de inferencias por parte de los usuarios.
- *Query:* establece consultas sobre la información que se ha almacenado en el modelo de datos usando SQL, a estas sentencias se agrega la función SEM_MATCH la cual facilita el uso de la funcionalidad SPARQL.
- *Enterprise:* Permite la adaptación con diferentes ontologías.

- *Extensibilidad:* Permite adaptar varias aplicaciones como adaptador para Jena, adaptador para Sesame, SQL* Loader (carga archivos N-Triple), Plugin protege para Oracle.
- *Accesibilidad:* Según (Torre, 2009), en cuanto a privilegios y seguridad de modelos que se crean en la base de datos se pueden destacar algunos aspectos como:
  - El creador de un usuario puede dar privilegios a otros usuarios sobre su modelo.
  - El usuario puede observar solos los modelos sobre los cuales tiene privilegios.
  - Solo el creador de un modelo puede eliminarlo.
- *Almacenamiento:* Es un almacenamiento normalizado, por lo que los literales y las URI largas se almacenan en una tabla aparte. Además soporta tipos de datos xsd.
- *Manipulación de datos:* Para poder manipular los datos RDF y OWL que son almacenados en la base de datos Oracle utiliza el paquete SEM_APIS en el cual se define una serie de funciones y procedimientos que permiten trabajar sobre los datos, funciones y procedimientos que se detallan a continuación (ORACLE, 2010):
  - *SEM_APIS.CREATE_SEM_NETWORK:* sirve para crear la red que permite soportar RDF, de esta forma permite definir los objetos necesarios para trabajar con esta tecnología.
  - *EXECUTE SEM_APIS.CREATE_RDF_MODEL:* sirve para crear un modelo de datos y para esto se debe estructurar una sentencia.
- *Inserción:* soporta carga masiva de datos mediante utilización de funciones y procedimientos utilizando el adaptador para Sesame por ejemplo. Además también se lo puede hacer por medio de consultas PL/SQL.
- *Inferencia:* Soporta una serie de niveles de inferencia, inclusión de reglas y de razonadores DL, dicha inferencia es forward-chaining, calculando las sentencias antes de almacenarlas en la base de datos.
- *Consultas:* soporta SPARQL embebido en SQL, lo que tiene la gran ventaja de ser compatible con todas las capacidades de SQL, incluyendo búsqueda de patrones y funciones de agregación.

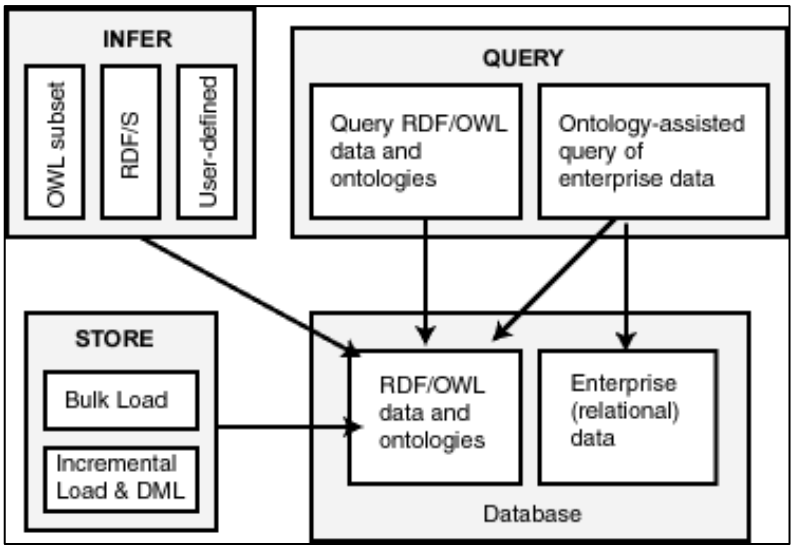


Figura 13 Capacidades de Oracle Semántico

En este proyecto Oracle Spatial 11g ha sido instalado sobre varios sistemas operativos, y se ha realizado operaciones de DML y DDL en forma masiva utilizando para ello las APIs de Sesame y Jena, obteniendo excelentes resultados en cuanto al procesamiento, consulta y almacenamiento de RDF, pero sobre todo al momento de aplicar inferencia sobre el modelo de datos RDF.

2.2.5. 4Store

4Store es una base de datos RDF, disponible bajo licencia GLP, diseñado para sistemas UNIXs y corre como un servidor sobre un cliente o en modo clúster. Los conectores clientes están disponibles para PHP. Ruby, Python y Java, para este último posee librerías propiamente desarrolladas para el caso. (W3C, 2011 b), menciona que 4Store soporta hasta 15 Billones de tripletas, y dentro de nuestra clasificación encaja dentro de los almacenes nativos.

4Store es ideal para CPUs de 64 bits, aunque corre con pequeñas bases de datos sobre CPUs de 32 bits, en nuestra investigación ha sido testeado en Ubuntu 10.10 de 32 y 64 bits, en cuanto a la dependencia utiliza raptor[64] (software libre que provee un conjunto de analizadores y serializadores que generan tripletas RDF para el análisis de sintaxis o de serializar las tripletas dentro de una sintaxis, (Beckett, 2011 a)) y rasqal[65](librería que se maneja de la sintaxis del lenguaje de consulta RDF como creación y ejecución de consultas (Beckett, 2011 b)) y varias

[64] http://librdf.org/raptor/
[65] http://librdf.org/rasqal/

librerías que varían de acuerdo a la distribución. En cuanto al respaldo en (Garlik, 2009 a), se menciona que hay herramientas de copias de seguridad a través de HTTP, pero a la fecha de la investigación no existía documentación al respecto.

En este proyecto 4Store ha sido instalado sobre varios sistemas operativos, y se ha realizado operaciones de DML y DDL en forma masiva obteniendo excelentes resultados en cuanto al procesamiento, consulta y almacenamiento de RDF, sin embargo para aplicar inferencia no es muy recomendable, además cuenta con funciones muy básicas.

2.2.6. Jena

Según (SourceForge, 2009), Jena es un framework para la construcción de aplicaciones de tipo web semántica, implementa APIs para trabajar con bloques de construcción de la WS como RDF y OWL. En este contexto Jena ha sido utilizado como API, y no como Base de Datos, puesto que se lo ha testeado usando JDBC para poder ser enlazado con MySQL[66]. El funcionamiento de Jena posee una clase fundamental que es el Model (Modelo), una API para hacer frente a una serie de tripletas RDF, este modelo puede ser creada desde los archivos del sistema o desde un archivo remoto. SDB es un componente de Jena y provee almacenamiento y consultas SPARQL, de conjuntos de datos usando bases de datos SQL convencionales. Además Jena proporciona acceso a un razonador, configurado para realizar razonamientos a varios grados. A continuación se mencionan las siguientes características (Torre, 2009):

- *Arquitectura:* En la Figura 14 se muestra la arquitectura de Jena. Donde las aplicaciones interactúan con un modelo abstracto que traslada operaciones de alto nivel a operaciones de bajo nivel sobre las sentencias que se almacenan en grafos RDF, permitiendo de esta manera un almacenamiento persistente, lo que facilita que se divida la información en grafos lógicos, para cada uno de los cuales se optimiza su almacenamiento, según una serie de subgrafos especializados, permitiendo que las búsquedas en cada grafo especializado fuera concurrente respecto a los demás. Una de las principales fortalezas de Jena es que ha ido creciendo gracias a diferentes contribuciones en forma de clases, paquetes, bibliotecas, aplicaciones u ontologías, que se han integrado en su código. (sourceforge, 2008).
- *Extensibilidad:* Al ser Open Source Jena continúa creciendo en medida a la modificación de sus fuentes publicadas y la inclusión de nuevas bibliotecas.

[66] http://www.mysql.com/

- *Interacción:* Jena provee un conjunto de utilidades en línea de comandos para el procesamiento de RDF, pero no posee por sí mismo características de cliente servidor.
- *Almacenamiento:* Jena implementa la interfaz Model para base de datos relacionales, este tipo de almacenamiento será el utilizado en nuestra investigación. El cual consiste en un almacenamiento genérico, con una tabla principal y los atributos son referencias del sujeto, predicado y objeto que conforman las tripletas. Aplica des normalización de tablas para aumentar su velocidad de almacenamiento, y se lo puede integrar con MySQL, Oracle, PostgreSQL, HSQLDB, entre otros.
- *Inserción:* Jena permite la carga masiva de tripletas, mediante métodos de la clase FileManager. (SourceForge, 2009).
- *Inferencia:* Soporta razonadores basados en reglas RDFS, OWL y Razonadores transitivos.
- *Consultas:* Soporta SPARQL y RDQL, sin embargo se lo utilizo solamente con SPARQL.

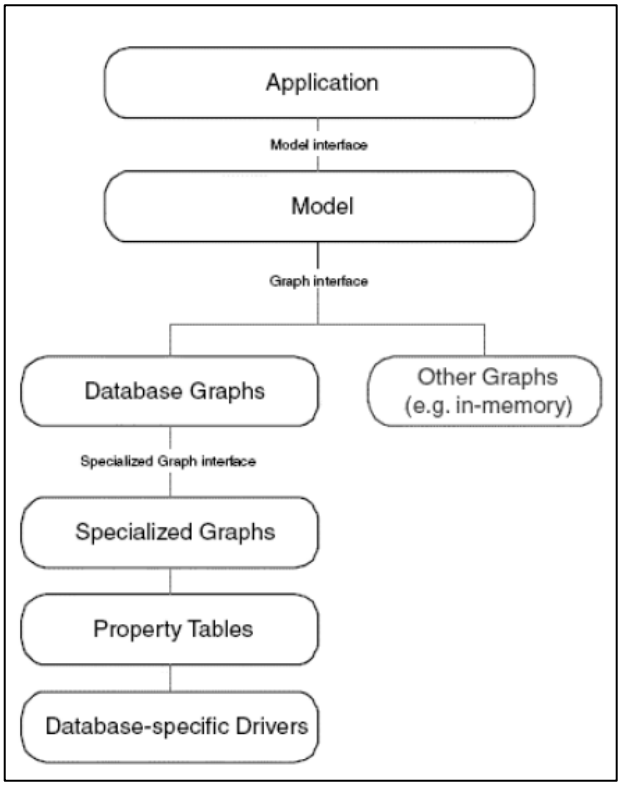


Figura 14 Arquitectura de Jena (Torre, 2009)

En la tabla VIII se resume las principales características de los gestores RDF mencionados anteriormente.

RDF Triple System	Inferencia	Soporte para Jena	Conectores Clientes	Escalabilidad	Lenguaje de Consulta	SPARQL End Point	Plataformas	Licencia	Analizadores	Transacciones	Backup y Restauración
4store	No posee clases que faciliten inferencia	NO	PHP, Ruby, Python, Java	15 Billones	SPARQL	SI	Windows, Linux, or Mac OS X	Open	RDF	NO	Archivos de Sistema de la Base de datos
Sesame	SI - RDFS	SI	Java	70 Millones	SPARQL, SeRQL	SI	Windows, Linux, or Mac OS X	Open	RDF/XML, Turtle, N-Triples, TriG and TriX.	SI	SI
Virtuoso	SI - RDFS y OWL limitado	SI	Java	15.4 Billones	SPARQL y SQL	SI	Windows, Linux, or Mac OS X	Open and Comercial	RDF, XML	SI	SI
Allegro Graph	SI Spatio-Temporal, RDFS	SI Jena 2.5	Java, Python, Lisp, Clojure, Ruby, Perl, C#, Scala	1+ Trillón	SPARQL	SI	Windows, Linux, or Mac OS X	Comercial	RDF/XML y N-Triples	Transaccion Logs	Archivos de Sistema de la Base de datos
Oracle 11g	SI, RDFS, OWL, Rules	SI	Java, SQL	1068 Millones	SPARQL y SQL	SI a través de APIS	Windows, Linux, Mac (Oracle 10g)	Comercial	RDF, RDFS	SI – total soporte para transacciones	SI
Jena	SI, RDFS, OWL, DAML, Rules	SI	Java, SQL	1.7 Billones	SPARQL Y RDQL	SI	Windows, Linux, Mac (Oracle 10g	Open	RDF, RDFS	SI – total soporte para transacciones	SI

RDF Triple System	Almacenamiento	Carga Masiva	DML de sentencias	Índices	Interfaz	Cliente/Servidor	Rendimiento Carga Tripletas
4store	Memoria, archivos y SGBD externos.	SI	SI	tokenising, double metaphones, y stemming.	Modo consola, y aplicación web cliente para consultas	SI	3500 – 0,001 min
Sesame	Memoria, archivos y SGBD externos.	SI	SI	Opsc, ospc, psoc, sopc	Modo consola, y aplicación web cliente	SI	3500 – 0,003 min
Virtuoso	Archivos y SGBD externos	SI	SI	Genérico denormalizado	ISQL y Conductor	SI	3500 – 0,004 min
Allegro Graph	Persistente.	SI	SI	Indexación automática y manual	Aplicación web Cliente, TopBraid Composer y TacerPorter.	SI	3500 – 0,00004 min
Oracle 11g	Oracle	SI	SI	Oracle B-Tree	Consola, SQLDeveloper y programas afines	SI	3500 – 0,6116 min.
Jena	Persistente y Memoria	SI	SI	Indexación automática y manual	Consola	SI	3500 – 0,018 min

Tabla VIII CARACTERÍSTICAS RDF TRIPLE STORE

2.3. Conclusión

En base a la investigación presentada en este capítulo, el API de Jena se ajusta perfectamente al dominio de la investigación, por sus características mencionadas y recalcadas en la tabla VIII. Es importante mencionar que para seleccionar un RDF Triple Store System, el factor más importante es el dominio de la aplicación o sistema que se desea implementar, por ejemplo si se no se requiere almacenar una gran cantidad de datos, se puede utilizar los conectores de Jena o Sesame, que permiten generar y administrar tripletas RDF a través de Java, y como base de datos se puede trabajar con MySQL para almacenar las tripletas generadas, cabe recalcar que Jena o Sesame almacenan directamente las tripletas en MySQL, en el capítulo 5 de esta investigación se detalla el proceso para la generación de RDF. Se ha elegido MySQL para el almacenamiento de RDF, de la misma manera que se seleccionó Jena, y está ligada al dominio de nuestro sistema. Cabe recalcar que Oracle posee varias ventajas sobre MySQL para trabajar con RDF, como se lo analizo anteriormente, sin embargo MySQL, es suficiente para el dominio de la investigación, a esto le agregamos que MySQL es Open Source y Oracle posee licencia Privativa.

3.0. Identificación Y Selección De Fuentes De Datos

Pondera la situación y luego actúa

Sun Tzu (El arte de la guerra)

3.1. Introducción

En el presente capítulo se detalla el proceso de identificación y selección de fuentes de datos, obedeciendo al primer paso que propone (Corcho & Gomez, 2010). En esta investigación las fuentes de datos son básicamente metadatos de OCW (OpenCourseWare) y OER (Open Educational Resource), esto con el objetivo de poder ayudar a las personas a obtener más información usando la URL de un recurso educativo abierto determinado, enlazándolo automáticamente con la información disponible en la nube de datos, logrando con esto añadir conocimiento y contexto a los OERs. La selección de datos se extiende únicamente a OCW y sus contenidos educativos abiertos como libros de texto, papers, presentaciones, material fuente, video/audio, tutoriales o demostraciones de temas específicos, para el propósito de esta investigación estos datos serán procesados y almacenados usando Jena y MySQL, esta parte será abordada en el capítulo 5.

3.2. OERs y Linked Data

Una cantidad significativa de información está expuesta en páginas web públicas de varias instituciones OCW alrededor de todo el mundo, esta información puede ser publicada en RDF como Linked Data, terminó que en este contexto se convierte en la mejor practica para exponer, compartir, conectar e interoperar con OCW/OER, ayudando a usuarios e instituciones a romper barreras limitantes de la web tradicional. Sin embargo actualmente no existe un medio que permita lograr la interoperabilidad a nivel de datos, debido a esto en (EURECOM, 2010) se menciona una vista conceptual en la arquitectura de redes llamado la capa OSI/OSI 7, un punto de vista posible de Linked Data en el futuro como una capa independiente en la arquitectura del Internet, ubicada en la parte superior de la capa de red, pero por debajo de las capas de aplicación debido a que proporciona un modelo de datos común para todas las aplicaciones como se ilustra en la Figura a continuación.

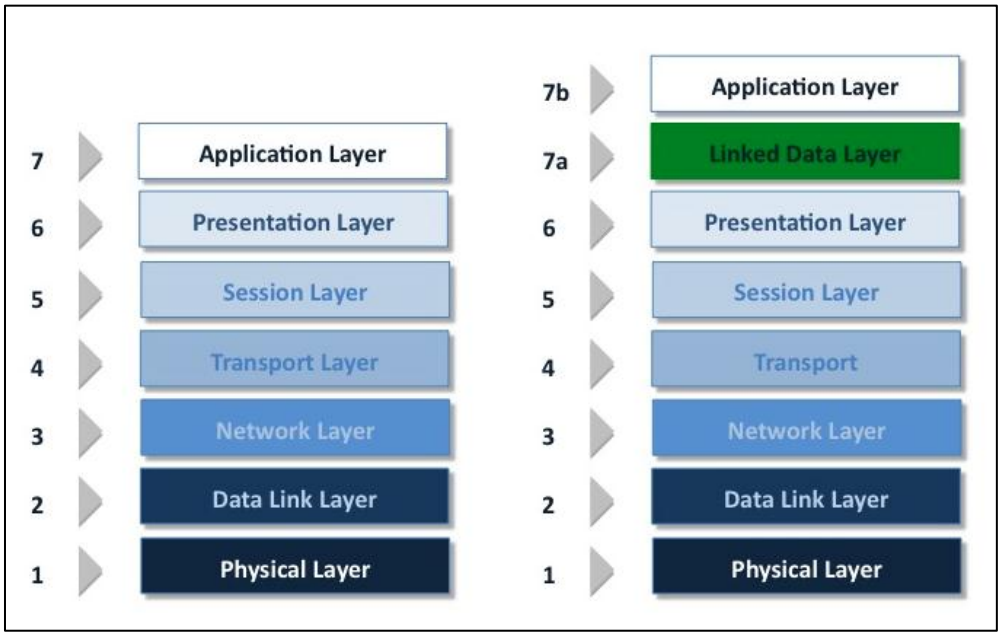


Figura 15 OSI/OSI Model and envisioned Linked Data Interoperability Layer (EURECOM, 2010)

Sin embargo alejándonos un poco de la conceptualización citada actualmente existen muchas barreras para materializar la idea de la interoperabilidad en la web y puntualmente en el contexto de esta investigación la interoperabilidad entre OCW, barreras que en mi opinión no son relevantes sino más bien obedecen a la aplicación de prácticas uniformes a la hora de publicar un recurso en la web, a continuación se citan algunas de las barreras actuales:

- *La naturaleza heterogénea de los OERs*: esto se aplica básicamente al estándar de metadatos que utilizan las instituciones OCW para describir los metadatos de un determinado OER.
- *El contenido de los OERs sobre sitios web frecuentemente está cambiando:* los OERs son dinámicos, y están cambiando continuamente tanto en sus contenidos como en sus metadatos por ejemplo el título de un OER es actualizado en algunos OCW dependiendo del periodo institucional de clases , algunos después de un tiempo relativamente corto son dados de baja.

3.3. Descripción del problema de Datos

Actualmente los datos (OERs) se encuentran en repositorios de diferentes OCW del mundo, el problema radica en que estos datos no están enlazados y su recuperación es bastante tediosa, esto debido a la heterogeneidad de los diferentes repositorio de OERs. Sin embargo hasta el momento el movimiento Open Linked Data ha realizado un

excelente trabajo y ha demostrado que la publicación de datos enlazados abiertos en la web es técnicamente posible, esto según (Shabir & Clarke, 2009). Los repositorios de OERs deben cumplir con estándares de lo contrario solo las búsquedas locales en los distintos servidores pueden revelar los "tesoros escondidos", los repositorios deben intercambiar la descripción (metadatos) de los recursos educativos almacenados en sus bases de datos (Wenk, 2010).

El hecho de que los datos estén enlazados es importante para obtener mucha más información sobre un recurso determinado, por ejemplo el URI de DBpedia http://dbpedia.org/page/Universidad_T%C3%A9cnica_Particular_de_Loja, muestra varios metadatos sobre esta Institución como lo podemos apreciar en la Figura 16, sin embargo algunos metadatos no están enlazados o complementados con más información por ejemplo la URI http://dbpedia.org/page/Ph.D._Jos%C3%A9_Barbosa_Corbacho correspondiente a la propiedad de canciller y este no está asociado a un vocabulario especifico como FOAF que describa al recurso, como se muestra en la Figura 17.

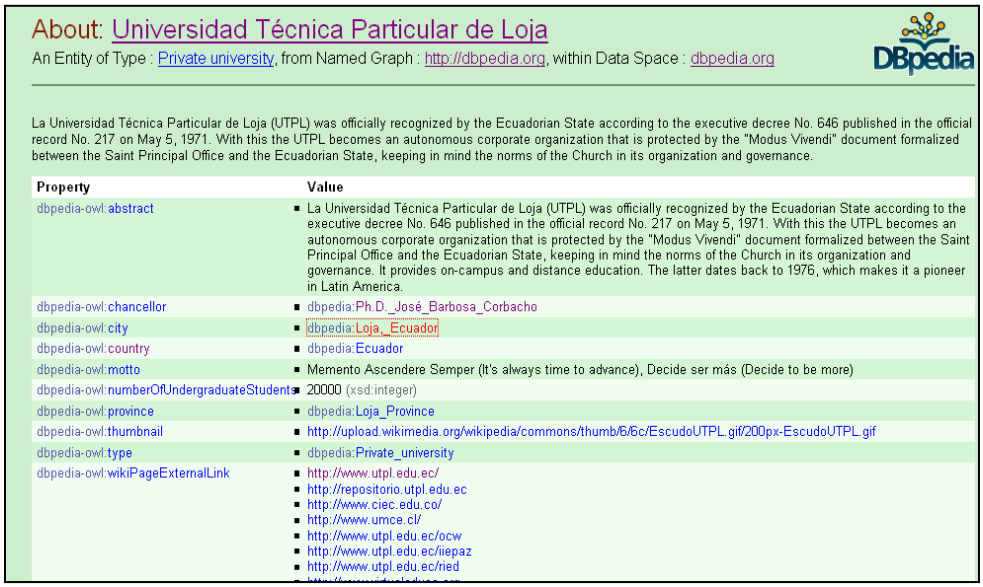


Figura 16 Metadatos de la UTPL[67]

[67] http://dbpedia.org/page/Universidad_T%C3%A9cnica_Particular_de_Loja

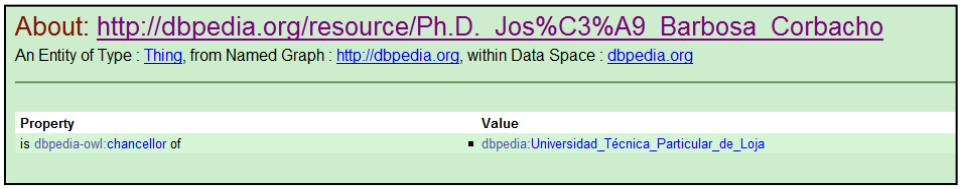


Figura 17 Metadatos sobre el canciller de la UTPL[68]

3.3.1. El problema de la Interoperabilidad entre repositorios OCW

Dentro de un sitio OCW existe un gran número de recursos educativos a los cuales no se puede acceder fácilmente ya que la mayoría de estos sistemas emplean sus propios modelos de datos para la descripción de sus recursos educativos, por esta razón las tareas de búsqueda, recuperación e integración que se vuelen temas difíciles de alcanzar. Nace así la idea de la interoperabilidad entre sitios heterogéneos, la interoperabilidad es definida por la (IEEE, 1991) como la habilidad de dos o más sistemas o componentes para intercambiar información y para usar la información que ha sido intercambiada. En los sistemas de eLearning, la interoperabilidad permite el intercambio y reutilización de recursos educativos (cursos, documentos, videos, tutoriales, etc.) que han sido desarrollados en plataformas educativas heterogéneas, lo cual permite:

- Incrementar la calidad y variedad de recursos educativos disponibles en el mercado.
- Preservar el capital invertido en tecnología y desarrollo de recursos educativos, ya que un recurso educativo podrá ser intercambiado o usado sin la necesidad de realizar costosas modificaciones.
- Garantizar que los usuarios con diferentes plataformas hardware y software puedan acceder a recursos educativos de fuentes heterogéneas, con pérdidas mínimas tanto de contenido como de funcionalidad.

Para lograr la interoperabilidad en eLearning es necesario definir una sintaxis y semántica común para la descripción de los recursos educativos (estándares, tecnologías de la SW y Linked Data como RDF que proporciona un soporte para metadatos flexible y abierto que permitirá la interconexión con otros sistemas similares).

[68] http://dbpedia.org/page/Ph.D._Jos%C3%A9_Barbosa_Corbacho

(Nelson & Romero, 2010), presentan un modelo de calidad de los contenidos OCW, formado por un conjunto de áreas que garantizan el aseguramiento de calidad de contenidos OCW/OER, estas áreas se las menciona a continuación de detallará el área de requerimientos tecnológicos e interoperabilidad:

- Modelos de desarrollo, distribución y licenciamiento.
- Alcance académico.
- Presentación al usuario.
- Evaluación y materiales de soporte.
- Requerimientos tecnológicos e interoperabilidad: Según lo mencionado en (Nelson & Romero, 2010), se presentan categorías como:
    - Estándares de interoperabilidad: se refiere a promoción de estándares que faciliten la interoperabilidad de servicios e intercambio de contenidos para mejorar el acceso a repositorios OCW/OER.
    - Formato de entrega del curso, la misma trata del método de entrega del contenido para acceder al curso.
    - Uso de protocolos para sindicación: Utilizar protocolos estandarizados como XML y RSS, para permitir a los usuarios hacer uso de datos de un sitio en otro contexto.
    - Características y técnicas SLATES[69] Sitios web y repositorios OCW/OER: Los sitios OCW deben incluir algunas de las categorías y técnicas derivadas de la utilización de tecnologías web 2.0
    - Consideraciones de retroalimentación: Es importante implementar mecanismos de retroalimentación a los usuarios OCW.
- Accesibilidad.

Interesantes iniciativas en el área de la interoperabilidad semántica de repositorios educativos se han desarrollado. Proyectos como Elena[70], OKI (Open Knowledge Initiative)[71] plantean interesantes arquitecturas de integración de repositorios educativos heterogéneos que se encuentran localizados en forma distribuida.

[69] Search Links Authoring Tags Extensions and Signalling
[70] http://www.elenaproject.org/
[71] http://www.okiproject.org/

3.4. ¿Qué datos necesito?

Como se mencionó anteriormente en esta investigación se trabajó con Open Educational Resources (documentos, gráficos, imágenes, videos, ejercicios de laboratorio, etc.). Por Educacional entendemos que el material es producido para su uso en contextos educativos formales, a pesar que los OER pueden ser usados para el aprendizaje no formal fuera del sistema educativo no oficial y por Open Educational Resources se entiende materiales digitalizados ofrecidos libremente y abiertamente para educadores, estudiantes y autodidactas para ser usados y reusados para la enseñanza, aprendizaje y la investigación (OECD, 2007). En (Wenk, 2010) se menciona que un OER puede ser:

- Contenido de aprendizaje (por ejemplo, cursos, módulos, objetos de aprendizaje, ejercicios, referencias a colecciones y archivos).
- Software (por ejemplo, herramientas de desarrollo, herramientas para organizar contenido, herramientas de simulación).
- Hardware (por ejemplo, prototipos de plataforma electrónica como Arduino)
- Un recurso de implementación (por ejemplo, licencias Creative Common)
- Un estándar de interoperabilidad (por ejemplo, SCORM , IMS)
- Media (por ejemplo, imágenes, grabaciones de audio, videos).

3.5. Proceso de Obtención de Datos

El proceso para la obtención de metadatos para ser procesados se detalla a continuación:

1. Los metadatos que se obtendrán serán de las páginas principales de un curso OCW, debido a que en la presentación de un curso, encontramos metadatos como autores, título del curso, fecha de creación del curso, objetivos, área de conocimiento entre otros, metadatos que no los encontramos en el RSS[72], pues la información que aquí se presenta es muy limitada. En la siguiente Figura se muestra la página inicial de un curso OCW y los metadatos a extraer.

[72] Really Simple Syndication

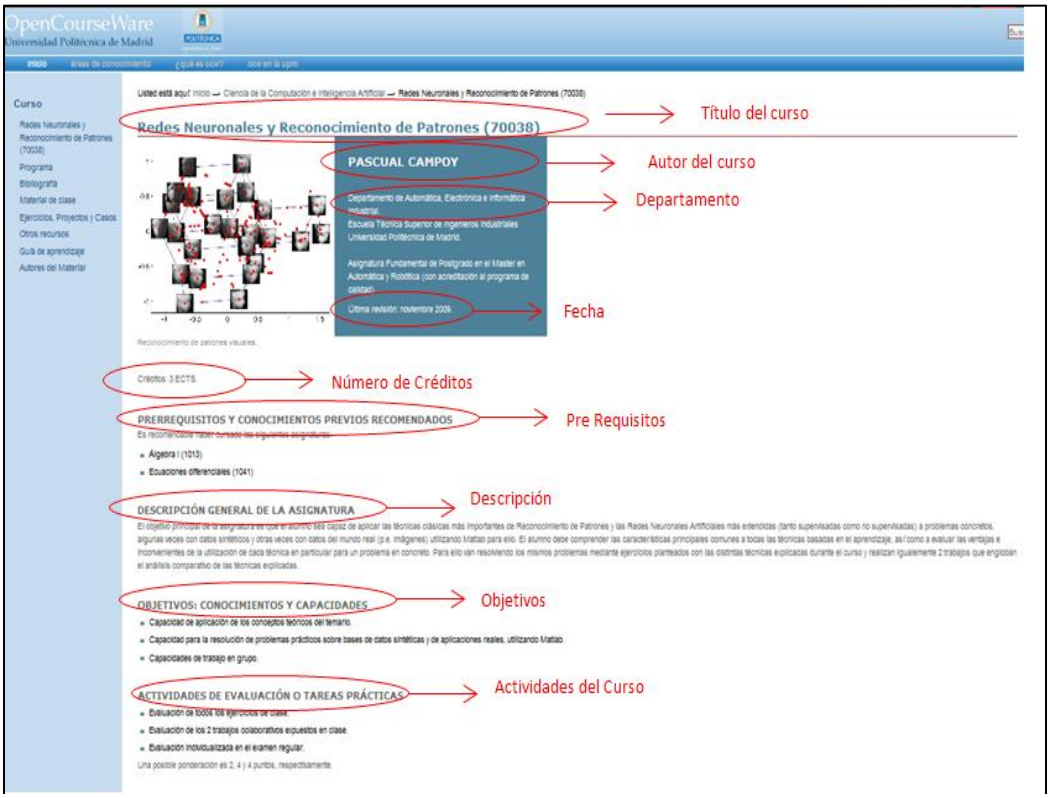


Figura 18 Metadatos de un Curso OCW de la UPM

2. Se copió toda la información de la página principal de 1500 cursos de diversos OCW de las principales Universidades de Iberoamérica, sin embargo, debido al nivel de datos obtenidos se decidió solamente procesar 20 cursos, 10 cursos pertenecientes al OpenCourseWare de la Universidad Politécnica de Madrid[73] y 10 cursos pertenecientes al OpenCourseWare de la Universidad Carlos III de Madrid[74], en un procesador de textos como Word.
3. Por motivos de encontrar patrones que nos faciliten la homogeneidad de información, se eliminó información innecesaria como por ejemplo los gráficos de los cursos, caracteres especiales. La figura a continuación, muestra el resultado de un simple procesamiento de limpieza de datos.

73 http://ocw.upm.es/

74 http://ocw.uc3m.es/ocwuniversia

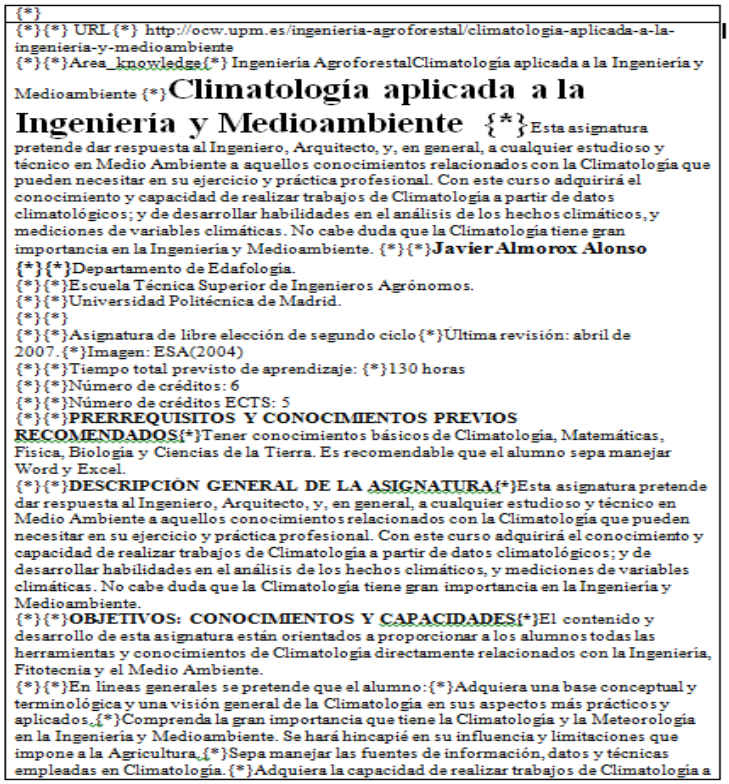

{*}
{*}{*} URL{*} http://ocw.upm.es/ingenieria-agroforestal/climatologia-aplicada-a-la-ingenieria-y-medioambiente
{*}{*}Area_knowledge{*} Ingeniería AgroforestalClimatología aplicada a la Ingeniería y Medioambiente {*}**Climatología aplicada a la Ingeniería y Medioambiente** {*}Esta asignatura pretende dar respuesta al Ingeniero, Arquitecto, y, en general, a cualquier estudioso y técnico en Medio Ambiente a aquellos conocimientos relacionados con la Climatología que pueden necesitar en su ejercicio y práctica profesional. Con este curso adquirirá el conocimiento y capacidad de realizar trabajos de Climatología a partir de datos climatológicos; y de desarrollar habilidades en el análisis de los hechos climáticos, y mediciones de variables climáticas. No cabe duda que la Climatología tiene gran importancia en la Ingeniería y Medioambiente. {*}{*}**Javier Almorox Alonso**
{*}{*}Departamento de Edafología.
{*}{*}Escuela Técnica Superior de Ingenieros Agrónomos.
{*}{*}Universidad Politécnica de Madrid.
{*}{*}
{*}{*}Asignatura de libre elección de segundo ciclo{*}Ultima revisión: abril de 2007.{*}Imagen: ESA(2004)
{*}{*}Tiempo total previsto de aprendizaje: {*}130 horas
{*}{*}Número de créditos: 6
{*}{*}Número de créditos ECTS: 5
{*}{*}**PRERREQUISITOS Y CONOCIMIENTOS PREVIOS RECOMENDADOS**{*}Tener conocimientos básicos de Climatología, Matemáticas, Física, Biología y Ciencias de la Tierra. Es recomendable que el alumno sepa manejar Word y Excel.
{*}{*}**DESCRIPCIÓN GENERAL DE LA ASIGNATURA**{*}Esta asignatura pretende dar respuesta al Ingeniero, Arquitecto, y, en general, a cualquier estudioso y técnico en Medio Ambiente a aquellos conocimientos relacionados con la Climatología que pueden necesitar en su ejercicio y práctica profesional. Con este curso adquirirá el conocimiento y capacidad de realizar trabajos de Climatología a partir de datos climatológicos; y de desarrollar habilidades en el análisis de los hechos climáticos, y mediciones de variables climáticas. No cabe duda que la Climatología tiene gran importancia en la Ingeniería y Medioambiente.
{*}{*}**OBJETIVOS: CONOCIMIENTOS Y CAPACIDADES**{*}El contenido y desarrollo de esta asignatura están orientados a proporcionar a los alumnos todas las herramientas y conocimientos de Climatología directamente relacionados con la Ingeniería, Fitotecnia y el Medio Ambiente.
{*}{*}En líneas generales se pretende que el alumno:{*}Adquiera una base conceptual y terminológica y una visión general de la Climatología en sus aspectos más prácticos y aplicados.{*}Comprenda la gran importancia que tiene la Climatología y la Meteorología en la Ingeniería y Medioambiente. Se hará hincapié en su influencia y limitaciones que impone a la Agricultura.{*}Sepa manejar las fuentes de información, datos y técnicas empleadas en Climatología.{*}Adquiera la capacidad de realizar trabajos de Climatología a

Figura 19 Procesamiento de Datos

4. Se construyó una aplicación utilizando la librería POI[75], una librería de java que permite procesar documentos de Microsoft. La aplicación lee el documento de Word con los datos procesados, crea y escribe un documento de Excel, con los metadatos requeridos por el ámbito de este proyecto. Es importante recalcar que esta aplicación no es óptima debido a la heterogeneidad entre los metadatos de los cursos de un mismo OCW, como por ejemplo no todos los cursos poseen objetivos o área de conocimiento, entre otros, es por esto que es muy difícil establecer un patrón para la extracción de metadatos, sin embargo el aplicativo nos reduce el tiempo requerido para la extracción de metadatos.

Básicamente la aplicación utiliza los siguientes métodos de la librería POI. (Apache, 2011).

- WordExtractor: Clase que nos permite extraer el texto desde un Documento Word.
- HSSFWorkbook: Clase publica que implementa de WordBook, el cual es el primer objeto que se crea al momento de leer el documento.

75 http://poi.apache.org/

- HSSFSheet: Clase publica que es una representación de alto nivel de una hoja de cálculo.
- HWPFDocument: Clase que actúa como el bucle que se lanzan todas las estructuras de datos en la palabras.
- getText: método interno para acceder al texto del documento.
- Split: método para manejar cadenas. Este método no pertenece a la librería POI, pero que se lo utilizo para cortar cadenas en base a un carácter en nuestro caso "*" y posteriormente almacenar estas cadenas en un documento de Excel.

La figura 20 ilustra el procedimiento de obtención de metadatos descrita en este capítulo.

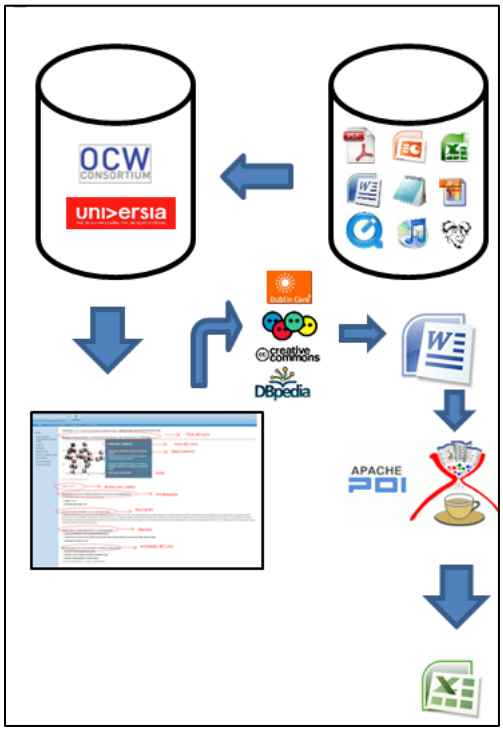


Figura 20 Proceso de obtención de datos

4.0. Descripción De Vocabularios OCW-OER-Licencias

La formulación de un problema es más importante que su solución.

Albert Einstein

4.1. Introducción

En este capítulo se presenta la reutilización de vocabularios, el tercer paso para la publicación de datos en la web, según lo mencionado por (Corcho & Gomez, 2010), en el proceso de publicación de datos enlazados en la web, además es una recomendación según (Healh & Bizer, 2011). Se ha reutilizado vocabularios consensuados como FOAF (para describir los autores de los OER), Dublin Core (para describir las propiedades genéricas de un OER), AIISO (para describir las propiedades de las instituciones OCW) y Creative Commons (para describir el tipo de licencias asociadas a un determinado OER), en otras palabras en base a las fuentes de datos seleccionadas en el capítulo anterior. Se reutilizara vocabularios existentes que se adapten a las necesidades del proyecto para posteriormente poder generar el RDFS y el RDF.

4.2. Uso de Vocabularios para Describir Datos

(Healh & Bizer, 2011), mencionan que el uso del estándar RDF proporciona un modelo genérico, abstracto de datos para la descripción de recursos, pero no proporciona un dominio especifico de términos para describir clases de objetos en el mundo y como estos se relacionan entre sí. Esta función es brindada por taxonomías, vocabularios y ontologías expresados en SKOS[76] un modelo para expresar jerarquías conceptuales, RDFS[77] y OWL[78] que proveen vocabularios para describir modelos conceptuales en términos de clases y sus propiedades.

Colectivamente SKOS, RDFS y OWL, proporcionan un continuo de expresividad. SKOS es utilizado para representar tesauros, taxonomías y jerarquías de tópicos y RDF y OWL se utilizan en los casos en que se requiera la representación de la inclusión de un objeto en la extensión de otro. (Healh & Bizer, 2011), mencionan que en un ambiente de Linked Data frecuentemente es suficiente expresar vocabularios en RDFS, aunque también es frecuente la utilización de RDF++, una combinación de RDFS y ciertas primitivas de OWL como sameAs, para afirmar que dos URIs identifican a un mismo recurso.

[76] http://www.w3.org/TR/skos-reference/
[77] http://www.w3.org/TR/rdf-schema/
[78] http://www.w3.org/TR/2004/REC-owl-features-20040210/

4.3. Reutilizando Vocabularios Existentes

Actualmente existen varios vocabularios consensuados que permiten la descripción de datos los cuales deben reutilizarse y no reinventar, para garantizar que los datos puedan ser consumidos por aplicaciones en la web facilitando la interoperabilidad. En esta investigación se utilizó vocabularios consensuados para describir recursos, personas, instituciones y licencias.

En el trabajo expuesto por (Healh & Bizer, 2011), se menciona que para seleccionar vocabularios para ser reusados se deben tener en cuenta los siguientes criterios.

- Usar y consumir: se debe realizar preguntas como ¿este vocabulario es de uso generalizado?, ¿el uso de este vocabulario, establece un conjunto de datos más o menos accesible a aplicaciones Linked Data?
- Mantenimiento y gestión: ¿este vocabulario es activamente mantenido de acuerdo a un proceso claro de gestionamiento?, ¿Cuándo y sobre qué bases son las actualizaciones realizadas?
- Cobertura: es el vocabulario lo suficientemente cubierto del conjunto de datos para justificar la adopción de sus términos y compromisos ontológicos.
- Expresividad: ¿el grado de expresividad en el vocabulario es el apropiado para el conjunto de datos y para el escenario de la aplicación?, ¿es lo suficientemente expresivo o no?

En la siguiente figura Christian Bizer, presentan una estadista acerca del uso de vocabularios comunes.

Prefix	Namespace	Used by
dc	http://purl.org/dc/elements/1.1/	66 (31.88 %)
foaf	http://xmlns.com/foaf/0.1/	55 (26.57 %)
dcterms	http://purl.org/dc/terms/	38 (18.36 %)
skos	http://www.w3.org/2004/02/skos/core#	29 (14.01 %)
akt	http://www.aktors.org/ontology/portal#	17 (8.21 %)
geo	http://www.w3.org/2003/01/geo/wgs84_pos#	14 (6.76 %)
mo	http://purl.org/ontology/mo/	13 (6.28 %)
bibo	http://purl.org/ontology/bibo/	8 (3.86 %)
vcard	http://www.w3.org/2006/vcard/ns#	6 (2.90 %)
frbr	http://purl.org/vocab/frbr/core#	5 (2.42 %)
sioc	http://rdfs.org/sioc/ns#	4 (1.93 %)

Figura 21 Uso de vocabulario comunes (Bizer et al, 2011)

En base a estas recomendaciones se ha decidido reutilizar los vocabularios detallados en el capítulo 1. A continuación se mencionan algunas razones por las cuales estos vocabularios fueron utilizados.

- Los elementos de Dublin Core se adaptan perfectamente para la descripción de recursos educativos, en base a esto se concluye que el estándar de metadatos Dublin Core o un subconjunto de estos es un fiel candidato para la descripción de OER y en conjunto con RDF facilitan la extracción y publicación de datos enlazados en la web. En la figura 22 se muestra un ejemplo de los metadatos de un OER utilizando Dublin Core y RDF.
- En base a las características que presenta FOAF es recomendable su utilización para describir información acerca de las personas que pertenecen a una institución OCW, identificados como autores y profesores.
- En base a las clases y propiedades que presenta AIISO se utilizó, clases como Course, Knowledge para representar Cursos y OCW respectivamente y propiedades como description, ispartof, entre otras.
- CCREAL, presenta una propiedad útil para representar la licencia que posee un determinado OCW y un OER.

```
<rdf:Description>
        <dc:creator>Asunción Gómez</dc:creator>
        <dc:title>Ontologies and Semantic
        </dc:title>
        <dc:subject>Ciencia de la Computación e Inteligencia Artificial
        </dc:subject>
        <dc:date>2008-04</dc:date>
        <dc:language>EN</dc:language>
        <dc:description>Course about Ontologies
        </dc:description>
</rdf:Description>
```

Figura 22 DUBLIN CORE Y RDF

4.3.1. Vocabulario OCW

Debido, a no todos los vocabularios, se adaptan a las necesidades de nuestro vocabulario, es posible crear un vocabulario propio al cual lo hemos llamado "ocw", cuyo URI es http://loxa.ec/lud/ocw/, y definirá clases y relaciones que no definen otros vocabularios, por ejemplo descripción de propiedades de un Curso OCW como

actividades de curso, actividades de evaluación, glosario entre otros. El prefijo a utilizar para este vocabulario será "ocw".

4.4. Vocabulario OCW/OER/Licence.

En base a lo analizado en la sección 4.3 se ha hecho uso de vocabularios consensuados, para construir un vocabulario hibrido que nos permita, contar con un esquema de RDF para, posteriormente en base a este esquema poder generar el RDF en base los metadatos obtenidos de OCW/OER. Para el desarrollo del vocabulario se implementó la herramienta COE Cmap Tools[79] y Altova SemanticWorks2011[80], para representar mediante mapas conceptuales las clases, sus respectivas propiedades y restricciones. En la Figura 23 se muestra el RFDS del dominio, conceptualizado mediante mapas conceptuales.

[79] http://cmapdownload.ihmc.us/coe/Web_Installersv411.b117/install.htm

[80] http://www.altova.com/semanticworks.html

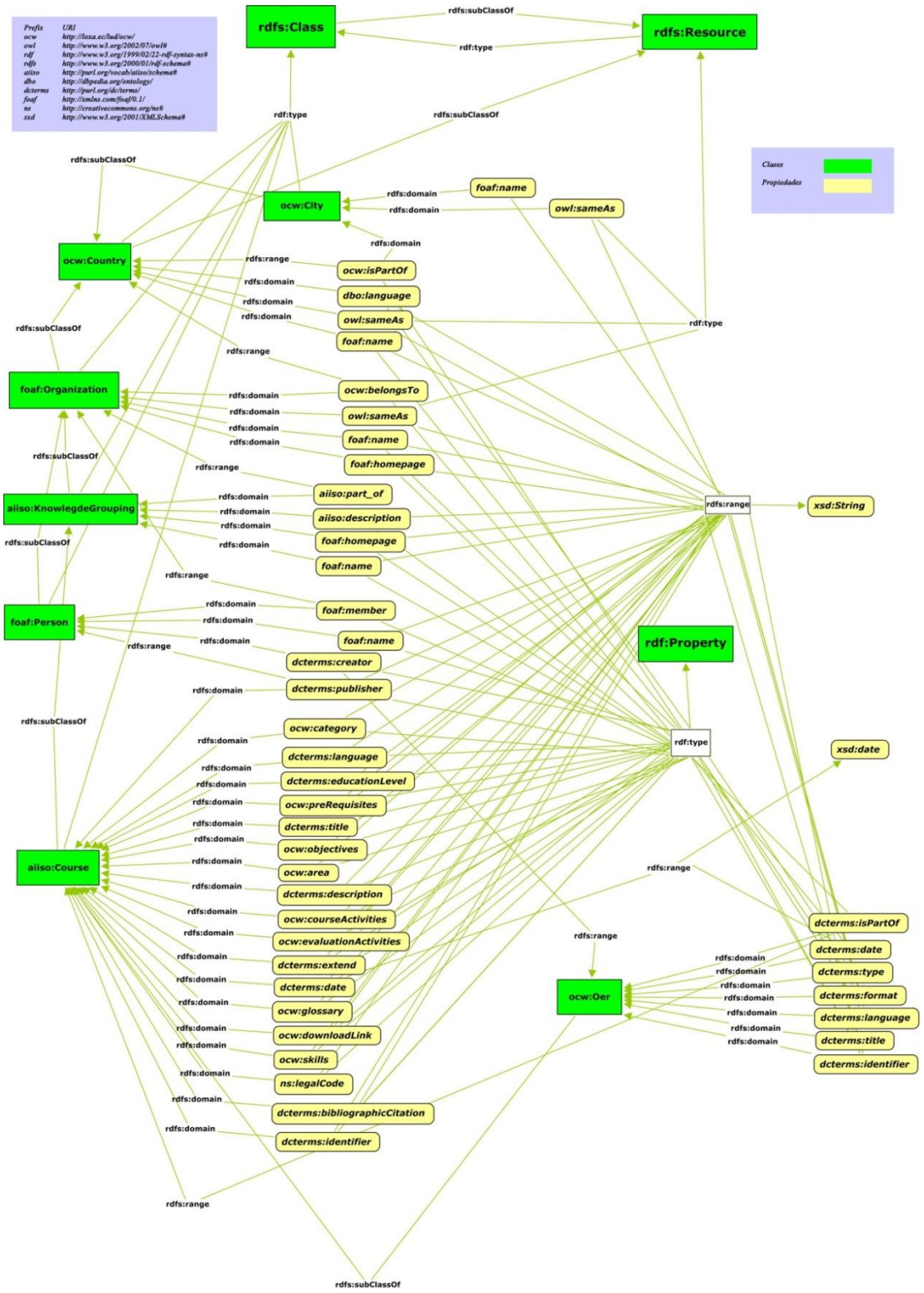


Figura 23 VOCABULARY OCW/OER/LICENCE

Como se puede apreciar en la Figura 23, se ha creado un vocabulario, propio que se ajuste a las necesidades del proyecto, en otras palabras en base a los metadatos que se extrajeron. A continuación en las siguientes tablas se detallan, cada uno de las clases y propiedades de los vocabularios utilizados.

DESCRIPCIÓN DE CLASES DEL VOCABULARIO OCW/OER/LICENCE	
foaf:Organization	
URI:	http://xmlns.com/foaf/0.1/Organization
Label:	Organization
Definition:	Clase que representa a una Organización, en nuestro dominio representa a Universidades.
ocw:Country	
URI:	http://loxa.ec/lud/ocw/Country
Label:	Country
Definition:	País.
Comment:	Utilizado para describir el país al que pertenece una Organización.
ocw:City	
URI:	http://loxa.ec/lud/ocw/City
Label:	City
Definition:	Ciudad
aiiso:KnowlegdeGrouping	
URI:	http://purl.org/vocab/aiiso/schema#KnowlegdeGrouping
Label:	KnowlegdeGrouping
Definition:	Representa una colección de recursos, objetivos de aprendizaje, horarios, entre otros materiales.
Comment:	Representa un sitio OCW
foaf:Person	
URI:	http://xmlns.com/foaf/0.1/Person
Label:	Person
Definition:	Representa a una persona.
Comment:	Representa una persona que pertenece a una organización en nuestro dominio se refiere a un Profesor que es creador de

	un OER.
aiiso:Course	
URI:	http://purl.org/vocab/aiiso/schema#Course
Label:	Course
Definition:	Courso OCW.
ocw:Oer	
URI:	http://loxa.ec/lud/ocw/Oer
Label:	Oer
Definition:	Recurso educativo de aprendizaje

Tabla IX DESCRIPCIÓN DE CLASES DEL VOCABULARIO OCW/OER/LICENCE

DESCRIPCIÓN DE RELACIONES DEL VOCABULARIO OCW/OER/LICENCE	
ocw:isPartOf	
URI:	http://loxa.ec/lud/ocw/isPartOf
Label:	isPartOf
Definition:	Relación que especifica que una ciudad es parte de un País
range	ocw:Country
domain	ccw:City
ocw: belongsTo	
URI:	http://loxa.ec/lud/ocw/belongsTo
Label:	belongsTo
Definition:	Relación que especifica que una Organización pertenece a un País.
range:	ocw:Country
domain	foaf:Organization
aiiso:part_of	
URI:	http://loxa.ec/lud/ocw/part_of
Label:	part_of
Definition:	Relación que especifica que una Grupo de conocimiento pertenece a una Organización.
range:	foaf:Organization

domain	aiiso:KnowlegdeGrouping
foaf:member	
URI:	http://xmlns.com/foaf/0.1/member
Label:	member
Definition:	Relación que especifica que una Persona es miembro de una Organización.
range:	foaf:Organization
domain	foaf:Person
dcterms:publisher	
URI:	**http://purl.org/dc/terms/publisher**
Label:	publisher
Definition:	Relación que específica que una Persona publica un Curso.
range:	aiiso:Course
domain	foaf:Person
dcterms:creator	
URI:	http://purl.org/dc/terms/#creator
Label:	creator
Definition:	Relación que especifica que una Persona es creador de un Recurso educativo de aprendizaje determinado.
dcterms:isPartOf	
URI:	http://purl.org/dc/terms/#isPartOf
Label:	isPartOf
Definition:	Relación que determina que un OER es parte de un Curso OCW.
range:	aiiso:Course
domain	Ocw:Oer

Tabla X DESCRIPCIÓN DE RELACIONES DEL VOCABULARIO OCW/OER/LICENCE

ETIQUETA DEL ELEMENTOS UTILIZADOS EN VOCABULARY OCW/OER/LICENCE	
TermName: dcterms:title	
URI:	http://purl.org/dc/elements/1.1/title

Label:	Title
Definition:	Nombre que se le otorga a un recurso.
Comment:	Por ejemplo Title="The Sound of Music"
domain	aiiso:Course, ocw:Oer
range	xsd:String
TermName: dcterms:type	
URI:	http://purl.org/dc/elements/1.1/type
Label:	type
Definition:	Género o naturaleza del recurso.
Comment:	Se recomienda seleccionar un valor de un vocabulario controlado como DCMITypevocabulary[81]. Para describir la manifestación física o digital del recurso, se usa el elemento Formato. Por ejemplo Type=”image”.
domain	ocw:Oer
range	xsd:String
TermName: dcterms:creator	
URI:	http://purl.org/dc/elements/1.1/creator
Label:	creator
Definition:	La entidad primariamente responsable de la creación del contenido intelectual del recurso.
Comment:	Puede ser una persona, organización o servicio. Por ejemplo Creator="Shakespeare, William"
domain	foaf:Person
range	ocw:OER
TermName: dcterms:date	
URI:	http://purl.org/dc/elements/1.1/date
Label:	Date
Definition:	Una fecha asociada con un evento en el ciclo de vida del recurso.
Comment:	La fecha será asociada con la creación o disponibilidad del recurso. Se recomienda utilizar un valor de datos codificado definido en el documento Date and Time Formats[82], por ejemplo Date="1998-02-16".
domain	aiiso:Course, ocw:Oer

[81] http://www.dublincore.org/documents/dcmi-type-vocabulary/
[82] http://www.w3.org/TR/NOTE- datetime

range	xsd:String
TermName: dcterms:format	
URI:	http://purl.org/dc/elements/1.1/format
Label:	Format
Definition:	La manifestación física o digital del recurso.
Comment:	El formato puede incluir el tipo de media o dimensiones del recurso. Podría usarse para determinar el software, hardware u otro equipamiento necesario para ejecutar u operar con el recurso. Ejemplos de las dimensiones son el tamaño y la duración. Se recomienda seleccionar un valor de un vocabulario controlado (por ejemplo, la lista de Internet Media Types[83] que define los formatos de medios de ordenador). Por ejemplo Format="image/gif", Format="4 kB".
domain	ocw:Oer
range	xsd:String
TermName: dcterms:identifier	
URI:	http://purl.org/dc/elements/1.1/identifier
Label:	Identifier
Definition:	Una referencia no ambigua para el recurso dentro de un contexto dado.
Comment:	Se recomienda identificar el recurso por medio de una cadena de números de conformidad con un sistema de identificación formal, tal como una URI[84], por ejemplo Identifier="http://purl.oclc.org/metadata/dublin_core/"
domain	aiiso:Course, ocw:Oer
range	xsd:String
TermName: dcterms:language	
URI:	http://purl.org/dc/elements/1.1/language
Label:	Language
Definition:	La lengua del contenido intelectual del recurso.
Comment:	Se recomienda usar RFC 3066[85] en conjunción con la ISO 639 [ISO639][86] que define las etiquetas de dos y tres letras

[83] http://www.iana.org/ assignments/media-types/
[84] Uniform Resource Identifier
[85] http://www.ietf.org/rfc/rfc3066.txt

	primarias para lenguaje, con sub etiquetas opcionales. Por ejemplo Language="en".
domain	aiiso:Course, ocw:Oer
range	xsd:String
TermName: ns:legalCode	
URI:	http://creativecommons.org/ns#legalcode
Label:	LegalCode
Definition:	La URL de un texto legal de una licencia.
domain	aiiso:Course
range	xsd:String
Property_Name: foaf:name	
Label	name
Definition:	nombre de una persona u organización , por ejemplo Leonardo Sarango
domain	foaf:Person, foaf:Organization, ocw:Country, ocw:City, aiiso:KnowledgeGrouping
range	xsd:String
Property_Name aiiso:description	
Label	Description
Definition:	Descripción de un grupo de conocimiento.
domain	aiiso:KnowledgeGrouping
range	xsd:String
Property_Name: dcterms:description	
Label	description
Definition:	Se usa para describir la descripción de un Recurso, en nuestro caso la descripción de un determinado curso.
domain	aiiso:Course
range	xsd:String
Property_Name: ocw:area	
Label	area
Definition:	Se usa para describir el área de conocimiento en donde se desenvuelve un curso OCW.

[86] http://www.loc.gov/standards/iso639-2/,

domain	aiiso:Course
range	xsd:String
Property_Name: ocw:objetives	
Label	objetives
Definition:	Se usa para describir los objetivos correspondientes a un curso OCW.
domain	aiiso:Course
range	xsd:String
Property_Name: ocw:preRequisites	
Label	preRequisites
Definition:	Se usa para describir los requisitos previos a la toma de un curso OCW como por ejemplo conocimientos académicos necesarios para la comprensión del curso.
domain	aiiso:Course
range	xsd:String
Property_Name: dcterms:educationLevel	
Label	educationLevel
Definition:	Se usa para describir el nivel de estudio que le corresponde al curso OCW, por ejemplo un curso puede tener un nivel básico, medio o avanzado expresado en niveles como los ciclos de una carrera universitaria.
domain	aiiso:Course
range	xsd:String
Property_Name: ocw:curseActivities	
Label	curseActivities
Definition:	Se usa para describir las actividades que propone el curso expresado la mayoría de las veces como syllabus.
domain	aiiso:Course
range	xsd:String
Property_Name: ocw:evaluationActivities	
Label	evaluacionActivities

Definition:	Se usa para describir las actividades de evaluación que propone el curso, por ejemplo autoevaluaciones.
domain	aiiso:Course
range	xsd:String
Property_Name: dcterms:extent	
Label	extent
Definition:	Se usa para describir el tiempo de aprendizaje que propone un curso OCW, por ejemplo puede ser de 3 meses.
domain	aiiso:Course
range	xsd:String
Property_Name: ocw:glossary	
Label	glossary
Definition:	Se usa para describir el glosario asociado al curso.
domain	aiiso:Course
range	xsd:String
Property_Name: ocw:downloadLink	
Label	downloadLink
Definition:	Se usa para identificar la URL de descarga de los OERs de un curso.
domain	aiiso:Course
range	xsd:String
Property_Name: ocw:skills	
Label	skills
Definition:	Se usa para identificar las habilidades que se adquirirán en el desenvolvimiento del curso.
domain	aiiso:Course
range	xsd:String
Property_Name: dcterms:bibliographicCitation	
Label	bibliographicCitation
Definition:	Se usa para identificar la bibliografía citada en un curso OCW.
domain	aiiso:Course
range	xsd:String

Property_Name: owl:sameAs	
Label	sameAs
Definition:	Indica que dos referencias URI en realidad se refieren a la misma entidad: los individuos tienen la misma "identidad".
domain	foaf:Organization, ocw:Country, ocw:City

Tabla XI DESCRIPCIÓN DE ELEMENTOS DEL VOCABULARIO OCW/OER/LICENCE

5.0. Generación De RDFS y RDF Data.

Si buscas resultados distintos, no hagas siempre lo mismo.

Albert Einstein

5.1. Introducción.

En este capítulo se detalla el proceso seguido, para la generación de RDFS y RDF, el tercer paso para el proceso de publicación de datos enlazados en la web según lo mencionado en (Corcho & Gomez, 2010). Los datos son generados a partir de los datos obtenidos (Capítulo 3), y de la reutilización de los vocabularios consensuados (Capítulo 4). Básicamente se desarrolló una aplicación web en JAVA que permita la generación automática de los datos RDF a partir de una cadena de conexión a una base de datos MySQL. No se utilizó para el desarrollo de este capítulo frameworks integrados como R20 u ODEMapster.

5.2. Descripción del Problema

En el capítulo anterior se mencionó que en un ambiente de Linked Data es necesario expresar vocabularios en RDFS, aunque también es frecuente la utilización de RDF++, una combinación de RDFS y ciertas primitivas de OWL como sameAs, para afirmar que dos URIs identifican a un mismo recurso implementando así distintos grados de expresividad en los dominios del modelado, como se lo menciona en (W3C, 2004 c). La idea es utilizar estos vocabularios para que se puedan conectar con tripletas RDF que enlacen clases y propiedades de un vocabulario con los de otro, mapeando vocabularios relacionados.

Al contar ya con los datos debidamente procesados y estructurados, es necesario generar el RDFS junto con los vocabularios analizados en el capítulo anterior. En la Figura 23 podemos observar la reutilización de vocabularios y la utilización de clases y propiedades de RDFS.

5.3. Loxa VocabRDFAdmin

Se ha desarrollado una aplicación web que permita la generación de RDFS y RDF a partir de fuentes de datos estructuradas como bases de datos relacionales. Estos datos serán procesados en un módulo de extracción estructurada, este módulo recibe de entrada una cadena de conexión de una base de datos relacional (MySQL), se recuperara el nombre de las tablas (clases), y atributos (propiedades), en base a esto se crea 2 tablas una con los datos mencionados, y otra tabla que contiene vocabularios codificados como constates en JAVA, los mimos que se recuperan mediante Jena.

Para procesar los datos obtenidos se creó un módulo donde se realiza un match, para posteriormente generar el RDF utilizando el API de Jena. En la Figura a continuación se muestra el dominio de la aplicación.

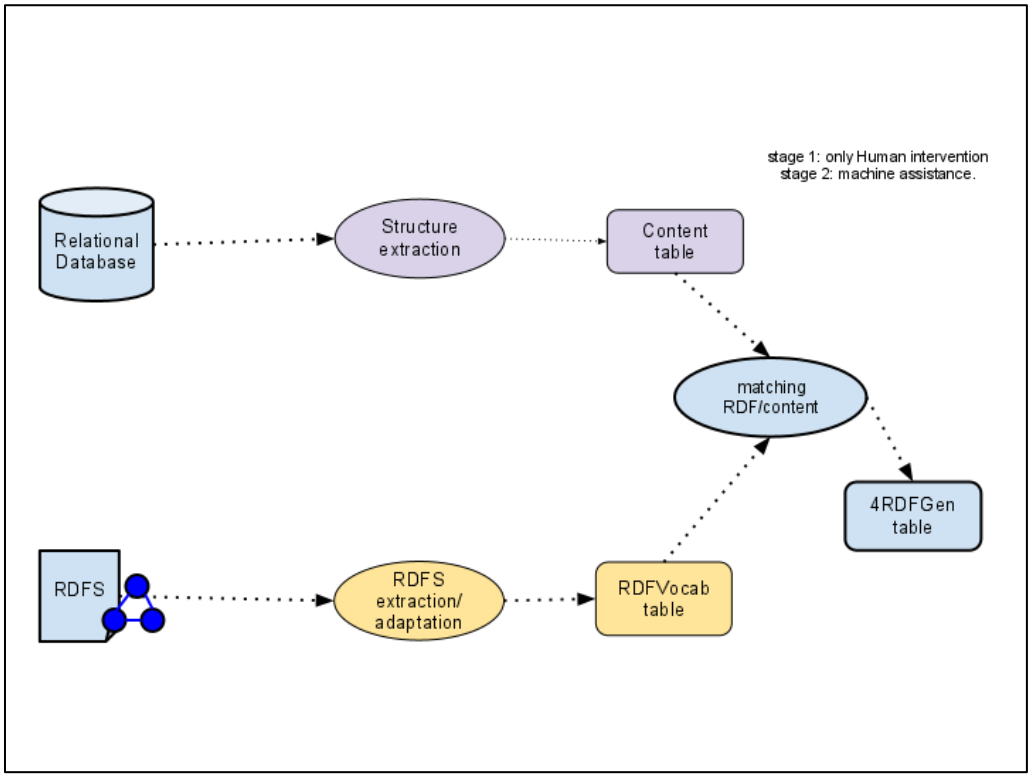


Figura 24 LOXA VOCABRDFADMIN

Como se puede observar la aplicación está conformada por varios módulos, los mismos que se detalla a continuación.

- Módulo para obtener metadata de fuentes de datos estructuradas.
    - Bases de datos en MySQL
- Módulo para incluir vocabularios en la herramienta.
    - RDF
    - RDFS
    - XSD
    - FOAF
    - DC
    - DCTerm
    - AIISO

- Creative Commons

- Módulo para generar RDFS
- Módulo para instanciar filas de fuente de datos estructurada (convertir a RDF, según vocabulario RDFS generado)

A continuación se detalla las tablas a crear.

1. Content (identify entities and attributes)
    - RDB: Tables, column names and datatypes.
    - CONTENT table structure (IdContent, Entity, Attibute, DataType, Comment).
2. RDF Vocabularies
    - RDFVocabularies table structure (IdRDFConcept, Namespaces, Concept, Comment)
3. 4RDFGeneration
    - 4RDFGeneration table structure (IdContent, IdRDFConcept)

5.4. Arquitectura de la Aplicación.

La Aplicación ha sido desarrollada en una arquitectura de 3 capas:

- *Capa de datos*: Donde residen los datos y se encarga de acceder a los mismos, conformada por MySQL como gestor de base datos para el almacenamiento y la recuperación de datos.
- *Capa de negocios:* Capa que almacena los métodos java, y reciben las peticiones del usuario.
- *Capa de Presentación*: Se ha utilizado como framework RichFaces[87], para la construcción del diseño de la aplicación.

En la siguiente Figura se muestra la arquitectura de la aplicación.

[87] Desarrollada por JBoss es una librería de componentes ricos para JSP.

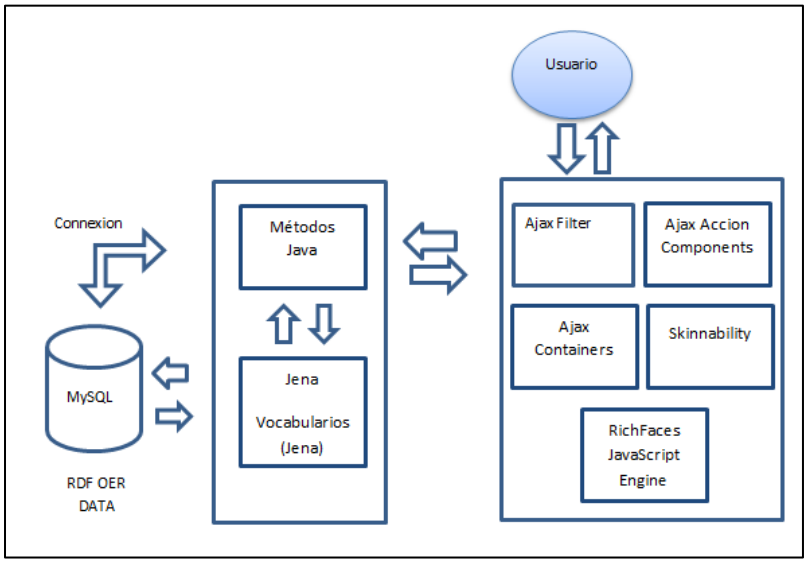


Figura 25 ARQUITECTURA DE LA APLICACIÓN

5.5. Desarrollo de la Aplicación.

Para el desarrollo de esta aplicación se ha utilizado tecnologías como Jena, RichFaces, Java, MySQL entre otras.

MySQL ha sido utilizado como repositorio de información, JAVA como lenguaje de programación, RichFaces como framework de diseño y Jena para manipular y almacenar grafos RDF. Jena interviene en la aplicación para usar sus vocabularios ya establecidos y para generar el RDF. Para esto a Jena recibe valores recuperados directamente desde la base de datos, los mismos que el usuario va formando interactivamente. Jena almacena estos valores como un modelo, este modelo es creado a su vez por ModelFactory sentencia que crea un modelo por defecto. Esto se lo consigue utilizando la siguiente sentencia. Model model = ModelFactory.createDefaultModel(). Este modelo permite que Jena pueda leer los datos que se le envían para que una vez que el modelo sea poblado Jena pueda realizar un conjunto de operaciones.

Definido y creado el Modelo se trabaja con Recursos (Resource) y Propiedades (Property) que son provistas por unas constantes que se manejan, se procesan en listas y son enviadas a una tabla llamada Vocabulary en la base de datos. Con los datos que retorna MySQL y los datos recuperados con la interacción del usuario se forman tripletas (Statements) que son conformados por tres partes:

- Sujeto: recurso
- Predicate: property
- Objetc: recurso o literal.

Una vez que se cuentan con los datos almacenados y cargados en un modelo, resulta sencillo escribir un archivo o crear una tabla en la base de datos con el RDF generado. Para este propósito se utiliza la función: model.write con un formato de abreviatura "RDF/XML-ABBREV".

A continuación se presenta la descripción de los métodos más relevantes del aplicativo:

- *Método Conectar:* A continuación se presenta la descripción de este método.
  - *Tipo*: Connection
  - *Parámetros Recibidos:* Recibe 5 cadenas que poseen la información que el usuario ha llenado en el formulario, datos como dirección IP del ordenador, puerto de comunicación, usuario, contraseña y nombre de la base de datos.
  - *Retorna:* cadena de conexión de tipo Connection.
- *Método CrearTableAllStructure:* En este método se establece una variable con una cadena de sentencias DDL y DML, llama al método conectar y al método armarConsulta, para procesar las sentencias en la base de datos. A continuación se presenta la descripción de este método.
  - *Tipo*: void
  - *Retorna:* no retorna valores, ejecuta una sentencia directamente en la base de datos
  - *Cadena de ejecución query:* variable de tipo String con la que almacena sentencias SQL para la creación de las tablas AllStructure y Vocabulay. En la Figura 26 se muestra la cadena SQL:

```
String query = ""
        + "DROP TABLE IF EXISTS AllStructure; "
        + "CREATE TABLE AllStructure As "
        + "SELECT t.table_name, t.table_schema, "
        + "c.column_name, c.Is_nullable, c.data_type, t.table_comment, c.Column_comment "
        + "FROM information_schema.tables t, information_schema.columns c "
        + "WHERE (t.table_schema = c.table_schema and t.table_name = c.table_name) AND "
        + "c.table_schema = '"+this.getMdbname() + "'; "
        + "DROP TABLE IF EXISTS VOCABULARY_LUD; "
        + "CREATE TABLE VOCABULARY_LUD (idVocab INT AUTO_INCREMENT, PRIMARY KEY(idVocab), Name_Vocabulary VARCHAR(75), "
        + "Resource_Vocabulary VARCHAR(100), Coment_Vocabulary VARCHAR(100)); "
        + this.armarConsulta();
```

Figura 26 CADENA SQL

La variable Mdbname es el nombre de la base de datos que ingreso el usuario.

- *Método armarConsulta:* A continuación se presenta la descripción de este método.
  - *Tipo*: String
  - *Retorna:* cadena con sentencias SQL.
  - *Metodos Llamados:* No llama a método alguno, sin embargo utiliza los valores de una lista con constantes definidas en el método ListVocabulary.
  - Cadena de ejecución query: variable de tipo String con la que almacena una sentencia SQL para la inserción de valores en la tabla Vocabulario.
- *Método ListVocabulary:* Devuelve valores de vocabularios a un constructor de otra clase que recibe 3 parámetros, el nombre del vocabulario, el nombre del recurso y un comentario; por ejemplo (FOAF.NS, FOAF.name.toString(), "name")). A continuación se presenta la descripción de este método.
  - *Tipo*: List
  - *Retorna:* Lista con los valores de las constantes.
- *Método getlistadoResultados:* Retorna una lista de arreglos a un componente RichFaces (DataTable), donde se listarán los valores extraídos de la base de datos. A continuación se presenta la descripción de este método.
  - *Tipo*: ArraysList
  - *Retorna:* ArrayList.
  - *Cadena de ejecución:* se envía una cadena SQL para extraer los datos concatenados de la tabla AllStructure, la cadena de conexión se muestran en la Figura 27.

```
String query = "SELECT concat (table_schema, '@', table_name, '.', "
        + "column_name, '.' ,Is_nullable, '.', data_type, '.', table_comment, '.', Column_comment)"
        + "FROM AllStructure;";
```

Figura 27 CADENA SQL DE CONSULTA

- *Método getDataSelect:* Método usado para traer datos de la tabla AllStructure de la base de datos y los retorna a un set de otra clase, este valor será recuperado por un componente RichFaces. A continuación se presenta la descripción de este método.
  - *Tipo*: List
  - *Parámetros Recibidos*: recibe un String con una consulta a MySQL.
  - *Retorna:* datos a una lista.

- *Método getlistadoResultadosVocab:* Retorna una lista de arreglos a un componente RichFaces (DataTable), donde se listarán los valores extraídos de la base de datos. A continuación se presenta la descripción de este método.
    - *Tipo*: ArraysList
    - *Retorna:* ArrayList.
    - *Cadena de ejecución:* se envía una cadena SQL para extraer los datos concatenados de la tabla Vocabulary.
- *Método getDataSelectVocab:* Método usado para traer datos de la tabla Vocabulary de la base de datos y los retorna valores a varios set de otra clase, estos valores serán recuperados por un componente RichFaces. A continuación se presenta la descripción de este método.
    - *Tipo*: List
    - *Parámetros Recibidos*: recibe un String con una consulta a MySQL.
    - *Retorna:* lista con valores extraídos de la base de datos.

- *Método getPredicateVocab:* Método usado para extraer datos de un vocabulario RDF que será subido por el usuario, e insertar el mismo en la base de datos, para lo cual se utiliza una consulta SPARQL. A continuación se presenta la descripción de este método.
    - *Tipo*: List
    - *Parámetros Recibidos*: recibe el archivo RDF y el nombre del vocabulario el mismo que será ingresado por el usuario.
    - *Retorna:* lista con las tripletas

En la Figura a continuación se muestra la consulta SPARQL para extraer las tripletas.

```
String q = "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> "
        + "PREFIX foaf: <http://xmlns.com/foaf/0.1/> "
        + "PREFIX owl: <http://www.w3.org/2002/07/owl#> "
        + "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
        + "select DISTINCT * where { "
        + "{?s rdf:type rdf:Property "
        + "OPTIONAL{?s rdfs:comment ?coment}"
        + "OPTIONAL{?s rdf:type ?Recurso}"
        + "FILTER (?Recurso = rdf:Property)"
        + "} "
        + "UNION "
        + "{?s rdf:type rdfs:Class "
        + "OPTIONAL{?s rdfs:comment ?coment}"
        + "OPTIONAL{?s rdf:type ?Recurso}"
        + "FILTER (?Recurso = rdfs:Class)"
        + "} "
        + "}";
```

Figura 28 CADENA CON CONSULTA SPARQL

- *Método generarRdf:* Método que genera el RDFS y RDF en base a las tripletas formadas por el usuario.

A continuación se presenta una tabla con las librerías utilizadas para el desarrollo del proyecto.

Librerías a utilizar: Para realizar la aplicación se utilizaron algunas librerías que complementan el funcionamiento de Jena, permiten utilizar componentes de diseño, permiten leer datos de archivos de fuentes estructuradas y permiten realizar conexiones a la base de datos. En la tabla XII se hace una descripción de cada una de ellas,

LIBRARY	DESCRIPCIÓN
JSF 2.0	Tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java, está conformada por: • jsf-api.jar: contiene el paquete javax.faces.* • jsf-impl.jar. contiene el paquete com.sun.faces.*
richfaces-api-3.3.3.Final	Permite utilizar los componentes de RichFaces, mediante, compatible con JSP.
arq-2.8.1	Motor de consulta SPARQL para Jena
icu4j-3.4.4	Permite soporte para Unicode a las aplicaciones, de esta forma las aplicaciones tienen el mismo resultado en cualquier plataforma.
iri-0.7	Realiza la comprobación de cadenas contra los diversos IRI o las especificaciones de las URI.
jena-2.6.2	Framework java construido para aplicaciones web semánticas, este provee un ambiente programable para RDF, RDFS y OWL, SPARQL e incluye un motor de inferencia para reglas base.
log4j-1.2.13	Es una biblioteca desarrollada en Java, que permite a los desarrolladores de software elegir la salida y el nivel de granularidad de los mensajes en tiempo de ejecución.
slf4j-api-1.5.6	Librería que permite tener información sobre un sistema cuando este se encuentra en tipo de ejecución.
stax-api-1.0.1	Estándar para procesamiento de XML, que permite transmitir datos XML desde y hacia una aplicación.
wstx-asl-3.2.9	Procesador XML de alto rendimiento que implementa stax.
xercesImpl-2.7.1	Sirve para la construcción de componentes y configuraciones para el análisis de un programa que es muy modular y fácil de programar.
Ojdbc14	Driver para conectar base de datos con java.

Tabla XII LIBRERÍAS UTILIZADAS

5.6. Resultados Obtenidos.

Los resultados obtenidos, son medidos en cuanto al RDF generado con la aplicación, resultados muy satisfactorios para la primera versión de esta aplicación web que se pretende llegué a ser considerada una herramienta muy útil a la hora de Generar RDF.

A continuación se describen como funciona la aplicación mediante su interfaz gráfica.

Interfaz Principal: Carga una página con 1 botón que permitirá cargar una cadena de conexión de una base de datos.

Figura 29 LOXA VOCABRDFADMIN APP

Botón Cargar Cadena de Conexión: Si el usuario selecciono este botón se le presentara unos cuadros de texto que tendrán que ser llenados con información correspondiente a la conexión a la base de datos como la IP de la maquina (en caso de ser local será localhost), puerto por el cual se conecta la base de datos por defecto es 3306, usuario de la base de datos, su contraseña y el nombre del modelo o base de datos. Este proceso se ilustra en la siguiente Figura.

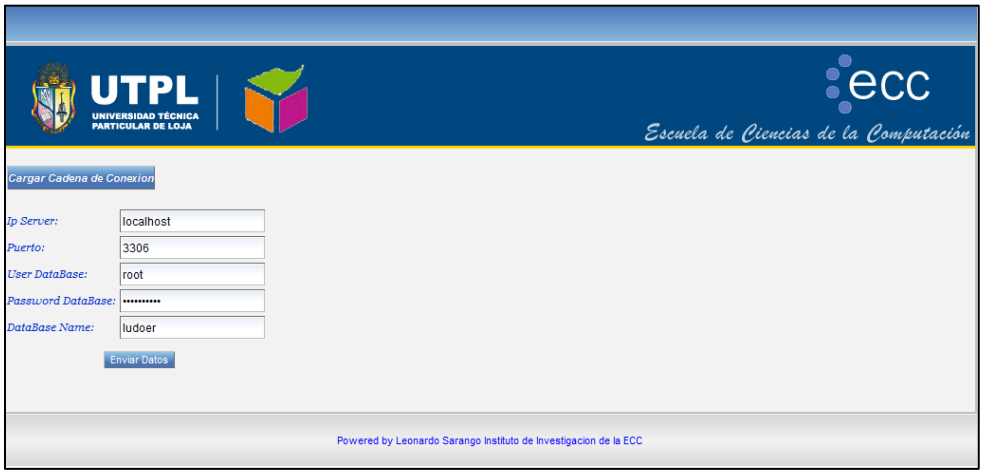


Figura 30 CARGAR ARCHIVO DE CONEXIÓN

Botón Enviar Datos: luego de haber llenado el formulario con los datos correspondientes es necesario seleccionar el botón enviar datos, esto levantará una serie de procesos a nivel de código java como establecer una conexión a la base de datos. Además se ejecutan una serie de sentencias DDL en la base como borrar la tabla en caso de que exista, crear tablas y selección de valores que se crean en la base de datos, estos métodos se encuentran especificados en la sección anterior. Luego se muestra al usuario una tabla con la información concatenada de las tablas creadas con la información acerca de sus columnas, tipo de datos, comentarios a nivel de columnas y de tablas y si el campo acepta valores nulos o no, esto se muestra en la Figura 31. Además a esto también se muestra una tabla con el vocabulario creado, utilizando las constantes de los vocabularios de Jena.

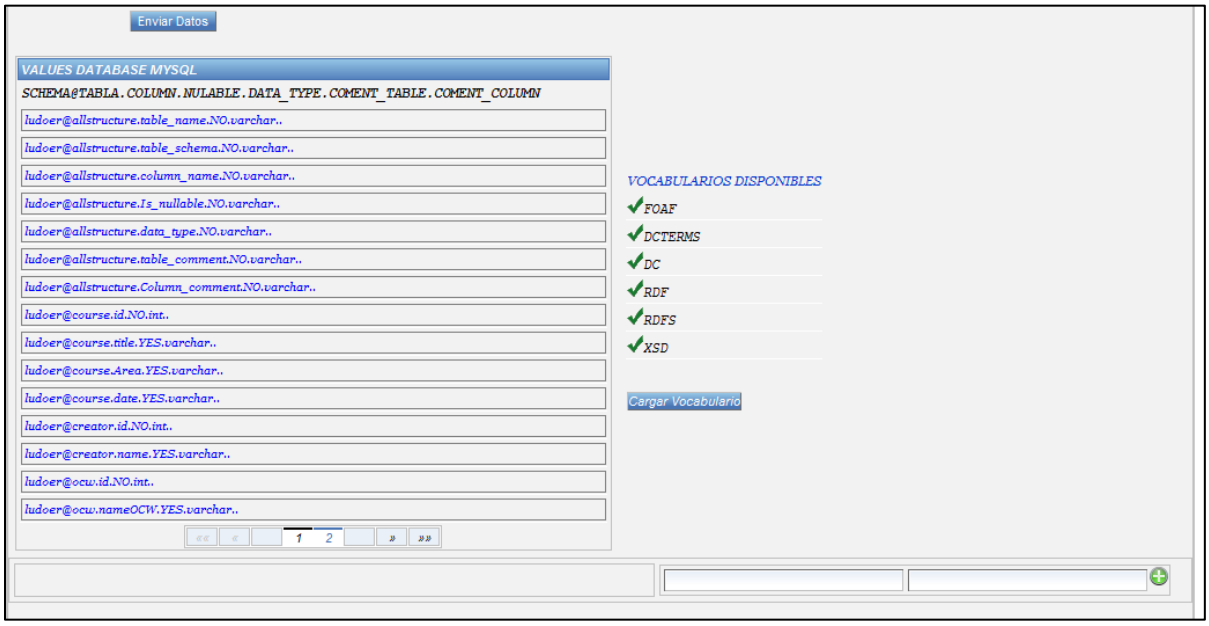


Figura 31 TABLAS RECUPERADAS DE LA BASE DE DATOS

Componentes Drag and Drop y Autocomplete: Los elementos de la Tabla DataBase MySQL, podrán ser arrastrados y pegados en una tabla cuya primera columna estará formada por la concatenación presentada en la primera tabla, y la segunda columna permite realizar un autocomplete para ubicar y relacionar datos y vocabularios; estos datos serán añadidos a un panel secundario por medio de un botón agregar previo a la generación de datos RDF.

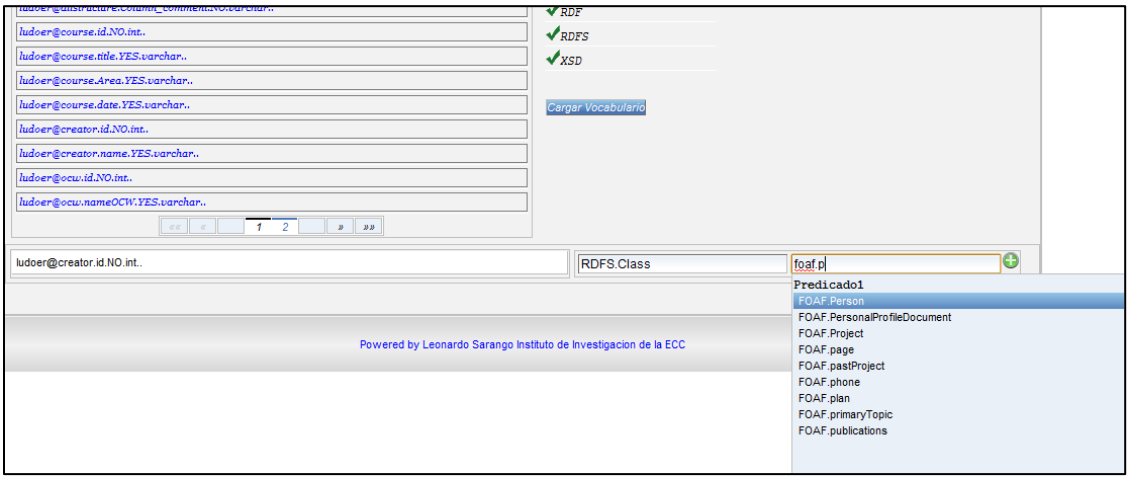


Figura 32 COMPONENTE AUTOCOMPLETE

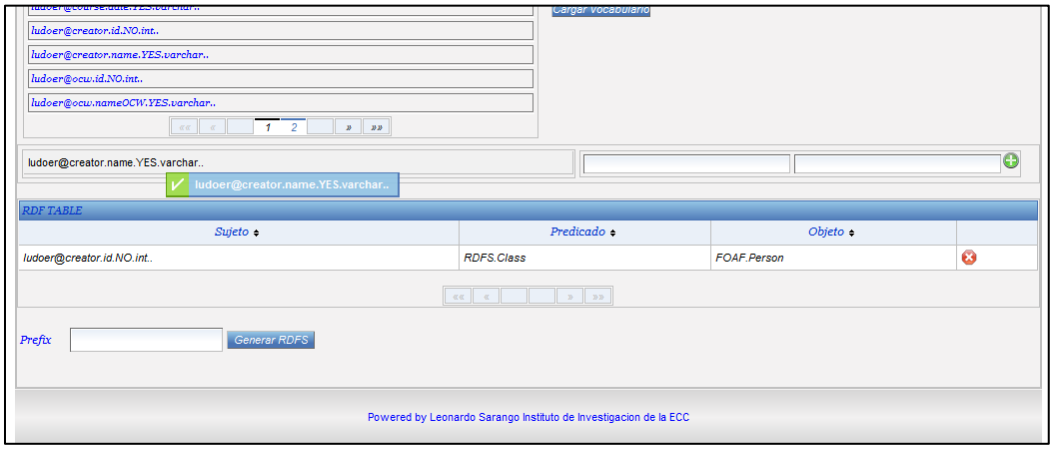


Figura 33 COMPONENTE DRAG AND DROP

Cargar un nuevo vocabulario: Permite al usuario seleccionar un archivo RDF, se extraen las tripletas y se presentan las propiedades y clases del vocabulario para ser utilizadas por el usuario.

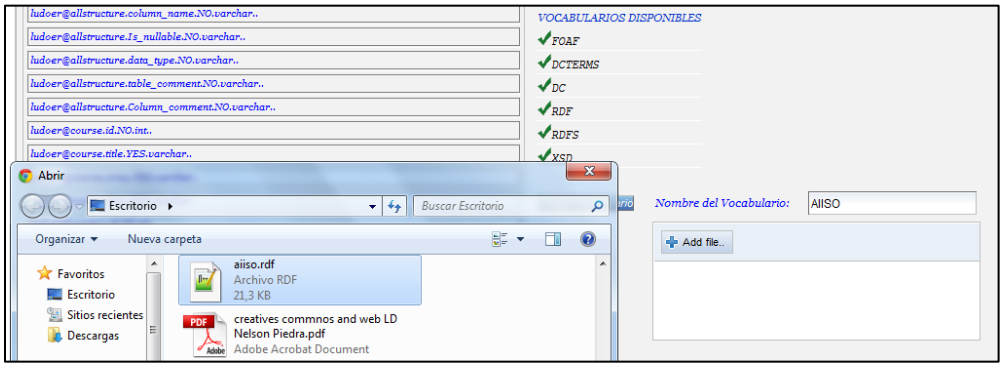
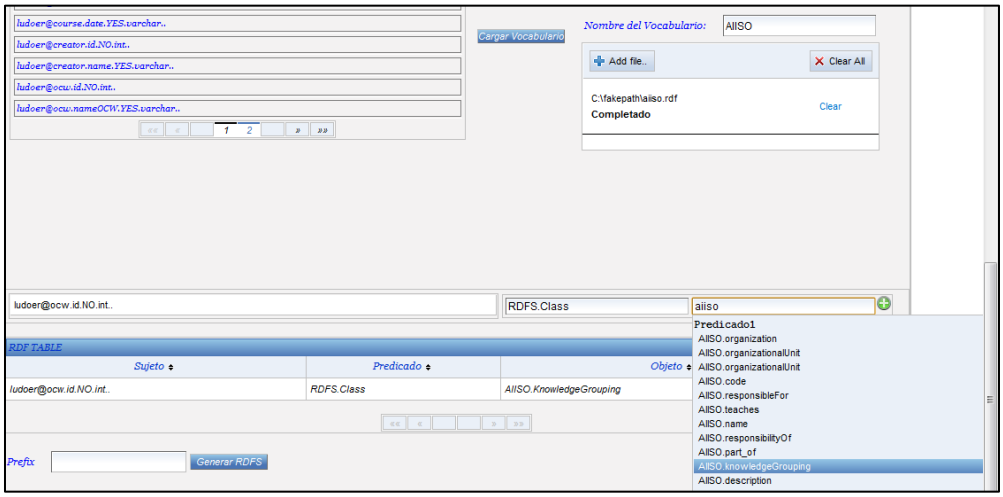


Figura 34 CARGAR VOCABULARIO RDF



Generar RDF: Se genera el RDFS a nivel de ID de las tablas únicamente, caso contrario se genera el RDF a nivel de los datos de las columnas de las tablas de la base de datos, y el resultado se presenta en un cuadro de texto.

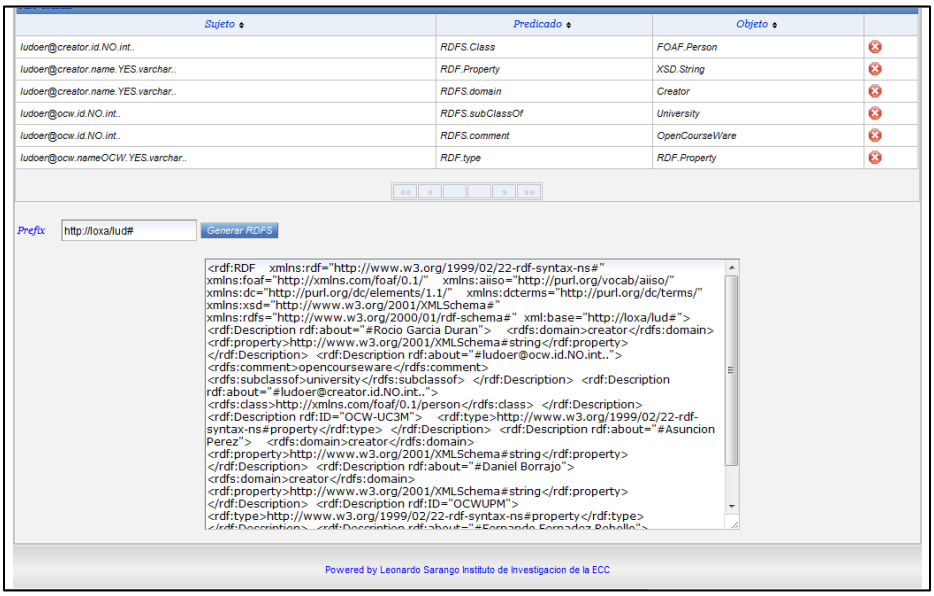


Figura 35 RDFS/RDF GENERADO

Como se puede observar en las figuras se obtiene el RDFS y el RDF, en base a como el usuario va formando las tripletas, el formato del resultado está dado por "RDF/XML-ABBREV". Si bien es cierto que el proceso de generación RDFS y RDF es un proceso tedioso, el aplicativo actualmente permite automatizar el proceso de generación. Actualmente en el mercado existen varias herramientas para la generación de RDF, pero no hay una que permita la generación automática a este nivel. No obstante la calidad del RDF dependerá únicamente de la estructura de las fuentes de datos estructuradas, y del conocimiento de los usuarios en temas de datos RDFS y RDF.

6.0. Implementación De Un Modelo De Visualización Para Datos RDF

"Since it sounded logical, I began by supposing it was very likely possible. But was there any evidence?"

Richard Feynman
What do you care what other people think

6.1. Introduction

En este capítulo se detalla la implementación de un módulo que permita la visualización de datos RDF, a partir de los datos generados con la aplicación Loxa VocabRDFAdmin, es importante mencionar que el dominio de la aplicación solamente funciona con un modelo de datos RDF especifico, esto significa que no es una aplicación que permite la visualización de cualquier dato RDF. Es importante mencionar que es muy complicado realizar una aplicación que muestre al nivel aquí presentado datos RDF, debido a que esta depende íntegramente de las consultas SPARQL aplicadas al modelo de datos RDF.

6.2. Arquitectura de la Aplicación

Para el desarrollo de esta aplicación se ha utilizado tecnologías como Jena, RichFaces, Java, MySQL entre otras.

La Aplicación ha sido desarrollada en una arquitectura de 3 capas y es similar a la de LOXA VOCABRDFADMIN:

- *Capa de datos*: Donde residen los datos y se encarga de acceder a los mismos, conformada por MySQL como gestor de base datos para el almacenamiento y la recuperación de datos.
- *Capa de negocios:* Capa que almacena los métodos java, y reciben las peticiones del usuario, y se utiliza tecnologías como Jena para manipular y consultar grafos RDF, los grafos son almacenados en Jena como un modelo y este modelo se lo crea por un *Factory* como: Model m = ModelFactory.createDefaultModel();
- *Capa de Presentación*: Se ha utilizado como framework RichFaces, para la construcción del diseño de la aplicación.

En la siguiente Figura se muestra la arquitectura de la aplicación.

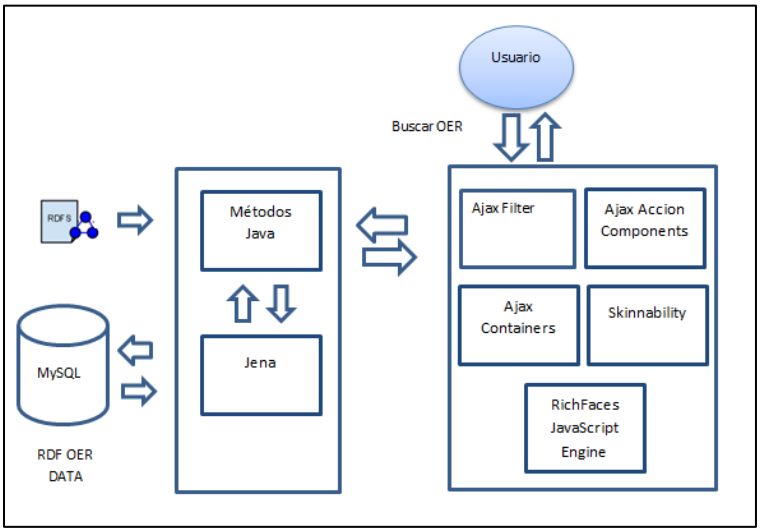


Figura 36 ARQUITECTURA DE VISUALIZADOR RDF LUD

Como se pude observar en la Figura 32. La aplicación permite recuperar y analizar un archivo RDF que contiene un grafo o una colección de grafos, en nuestro dominio los grafos corresponden a los OER, Cursos OCW, Autores y Organizaciones, estos datos son almacenados en una base de datos MySQL, es importante mencionar que Jena almacena directamente los datos en la base de datos mediante un método java que recibe la ubicación del archivo RDF, y el nombre del modelo en la base de datos. Además se implementó métodos para acceder a los datos almacenados en formas de tripletas por Jena en MySQL, y poder consumirlos utilizando para ello SPARQL. Los resultados obtenidos se muestran en una interfaz web, para ello se utilizó los componentes de RichFaces.

Básicamente se trabaja con una programación orientada a objetos, para recuperar y listar datos directamente desde la base de datos, cuando se recupera varios datos a la vez de una misma propiedad como objetivos o requerimientos del curso se trabajó con listas.

A continuación se presentar la descripción de los métodos más relevantes del aplicativo:

- *Método iniciarDB:* A continuación se presenta la descripción de este método.
  - *Tipo*: IDBConnection(Jena)
  - *Parámetros Recibidos:* Recibe 4 constantes definidas en otra clases con información necesaria para realizar la conexión como Host, puerto de comunicación, usuario, contraseña y nombre de la base de datos.

- *Retorna:* cadena de conexión de tipo Connection.

- *Método iniciarModelo:* A continuación se presenta la descripción de este método.
  - *Tipo*: ModelMaker
  - *Parámetros Recibidos:* Recibe una cadena de conexión de tipo IDBConnection.
  - *Retorna:* creación de un modelo, para ello se utiliza la instrucción ModelFactory.createModelRDBMaker(cadena de conexion);
- *Método crearModelo:* Crea un nuevo modelo para el almacen de tripletas. A continuación se presenta la descripción de este método.
  - *Tipo*: boolean
  - *Parámetros Recibidos:* nombre del modelo a crear (modelName).
  - *Retorna:* verdadero si el modelo ha sido creado correctamente y falso si el modelo no pudo ser creado.
- *Método getAllOER:* Método donde se extraen los datos desde MySQL. A continuación se presenta la descripción de este método.
  - *Tipo*: List
  - *Parámetros Recibidos:* acepta como entrada una consulta SPARQL.
  - *Retorna:* retorna una lista con todos los OERS existentes.
- *Método getDataSelect:* Método donde se extraen los datos desde MySQL. A continuación se presenta la descripción de este método.
  - *Tipo*: List
  - *Parámetros Recibidos:* acepta como entrada una consulta SPARQL.
  - *Métodos llamados:* iniciarDB, iniciarModelo.
  - *Retorna:* retorna una lista con todos los OER resultado de la búsqueda.
- *Método getDataSelect:* Método donde se extraen los datos desde MySQL. A continuación se presenta la descripción de este método.
  - *Tipo*: List
  - *Parámetros Recibidos:* acepta como entrada una consulta SPARQL.
  - *Métodos llamados:* iniciarDB, iniciarModelo.
  - *Retorna:* retorna una lista con todos los Cursos disponibles.
- *Método getDataSelectOCW:* Método donde se extraen los datos desde MySQL. A continuación se presenta la descripción de este método.
  - *Tipo*: List

- *Parámetros Recibidos:* acepta como entrada una consulta SPARQL.
- *Métodos llamados:* iniciarDB, iniciarModelo.
- *Retorna:* retorna una lista con todos los OCW disponibles.

- *Método getDataSelectAutor:* Método donde se extraen los datos desde MySQL. A continuación se presenta la descripción de este método.
  - *Tipo*: List
  - *Parámetros Recibidos:* acepta como entrada una consulta SPARQL.
  - *Métodos llamados:* iniciarDB, iniciarModelo.
  - *Retorna:* retorna una lista con todos los Autores disponibles.
- *Método CargarResultados:* A continuación se presenta la descripción de este método.
  - *Tipo*: void
  - *Métodos llamados:* getDataSelect
  - *Retorna:* retorna una lista con todos los OER disponibles.
- *Método buscarResultados:* A continuación se presenta la descripción de este método.
  - *Tipo*: void
  - *Métodos llamados:* getDataSelect y se le envía una consulta armada por medio del método armarSPARQL.
  - *Retorna:* retorna una lista con todos los OER disponibles resultados de la busqueda, cabe mencionar que se controla que no se presenten datos duplicados.
- *Método buscarResultados:* A continuación se presenta la descripción de este método.
  - *Tipo*: void
  - *Métodos llamados:* getDataSelect y se le como filtro el nombre del Recurso.
  - *Retorna:* establece los valores a métodos get y set de otra clase, estas variables serán accedidas para mostrar en un componente RichFaces.
- *Método armarSparql:* A continuación se presenta la descripción de este método.
  - *Tipo*: String
  - *Parámetros Recibidos:* valor obtenido de la caja de texto buscar OER y una consulta SPARQL.
  - *Retorna:* una consulta SPARQL.
- *Método guardar:* Método que se encarga de guardar en RDF en la Base de Datos. A continuación se presenta la descripción de este método.
  - *Tipo*: void
  - *Parámetros Recibidos:* Ubicación del archivo RDF y el nombre del Modelo.

- *Métodos llamados:* iniciarModelo
- *Retorna:* una consulta SPARQL.

- *Método obtenerLatitud:* Método que se encarga de obtener la latitud de un determinado país mediante una consulta SPARQL a DBpedia. A continuación se presenta la descripción de este método.
  - *Tipo*: String
  - *Métodos llamados:* QueryExecution
  - *Retorna:* latitud.
- *Método obtenerLonguitud:* Método que se encarga de obtener la longitud de un determinado país mediante una consulta SPARQL a DBpedia. A continuación se presenta la descripción de este método.
  - *Tipo*: String
  - *Métodos llamados:* QueryExecution
  - *Retorna:* longuitud.

Para realizar una consulta SPARQL sobre DBpedia se implementa la siguiente instrucción. QueryExecution qexec = QueryExecutionFactory.sparqlService("http://dbpedia.org/sparql", query);

6.3. Resultados Obtenidos

Los resultados obtenidos en base a las consultas SPARQL, previamente definidas en el código, fueron satisfactorios y suficientes para demostrar el consumo y enlace de datos.

Interfaz Principal: Muestra todos los recursos disponibles enmarcados en una tabla Como se puede observar en la Figura a continuación presentada, además muestra un Mapa que consume el servicio de Google Maps, y que muestra la ubicación del País relacionado a los Cursos OCW, mediante una consulta SPARQL a DBPEDIA. Los Nombres que están subrayados son Link que muestran metadatos asociados; por ejemplo si el usuario selecciona el nombre del Recurso Ingeniería de Software se muestran los detalles del recurso en un panel, como se lo puede apreciar en la Figura 37; de igual manera si se selecciona el link Forma Parte del Curso, levantara un panel con la información relacionada al curso, esto se demuestra en la Figura 38. Por último si se selecciona Listar Autores esto cargara un panel con los nombre de los autores del Curso, este panel se puede apreciar en la Figura 39. Es importante mencionar que para cerrar el panel es necesario seleccionar en la imagen que se usó para mostrar el panel.

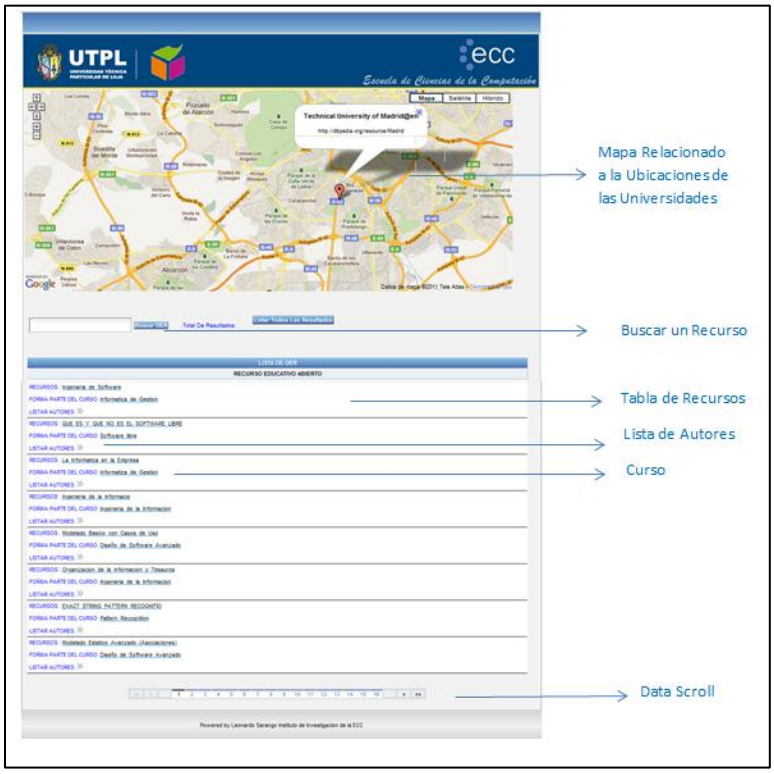


Figura 37 INTERFAZ PRINCIPAL DE VISUALIZADOR RDF LUD

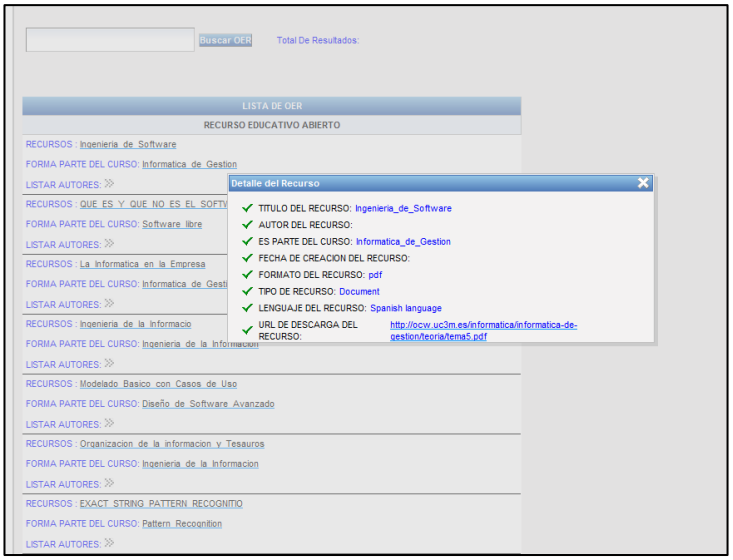


Figura 38 DETALLE DEL RECURSO

A continuación se detalla la definición de cada label presentado en este panel:

- *Autor del Recurso:* En el caso de que el recurso no tenga asociado un autor, este dado no será presentado,
- *Es Parte del Curso:* Muestra a que Curso pertenece dicho OER.
- *Fecha de Creación del Recurso:* Muestra la fecha de creación en caso de tener el dato asociado.
- *Formato del Recurso:* Muestra la extensión de dicho recurso.
- *Tipo de Recurso:* Muestra el tipo del OER, este puede ser documento, video, etc.
- *Lenguaje del Recurso:* Muestra el lenguaje del Recurso.
- *URL de Descarga del Recurso:* Muestra el link para descargar el recurso o en su defecto lo mostrara en otra ventana del navegador.

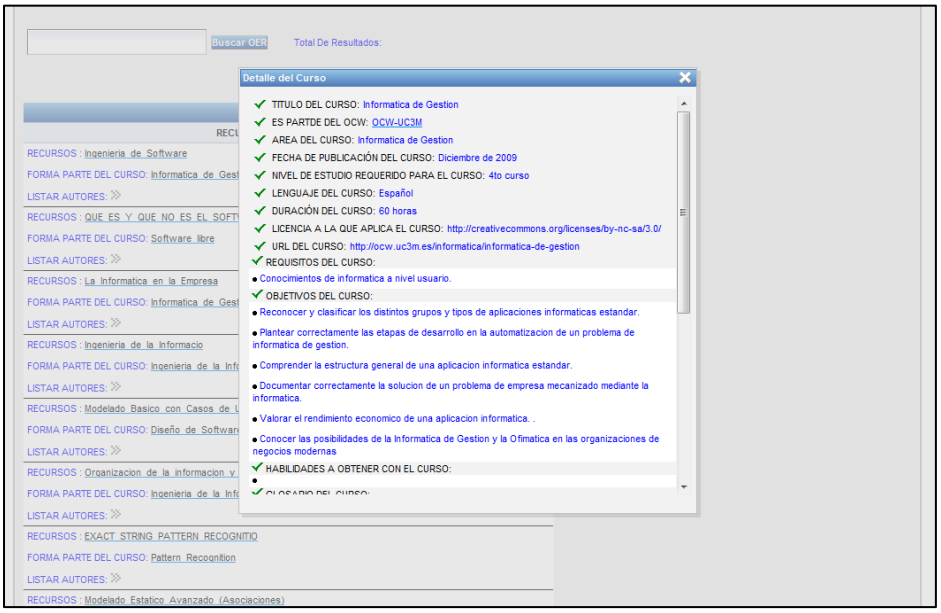


Figura 39 DETALLE DEL CURSO

A continuación se detalla la definición de cada label presentado en este panel:

- *Es Parte del OCW:* Muestra un link con el dato a cual OCW pertenece el curso, el resultado de seleccionar este link se plasma en la Figura 40.
- *Área del Curso:* Es el área de conocimiento a cual pertenece el curso.
- *Fecha de Publicación del Curso:* muestra la fecha de publicación del curso.
- *Nivel de Estudio Requerido para el Curso:* Muestra el nivel de estudio que debe tener el estudiante para tomar el curso.
- *Lenguaje del Curso:* Especifica el lenguaje del curso.
- *Duración del Curso:* Especifica la extensión en términos de tiempo del curso.
- *Licencia a la que Aplica el Curso:* muestra el link de la licencia Creative Commons a la cual aplica el curso.
- *URL del Curso:* Link del curso, si es seleccionado abrirá una nueva ventana en el navegador con la página principal del curso.
- *Requisitos del Curso:* Muestra una tabla con los requisitos sobre temas específicos de estudio, previo al seguimiento del curso.
- *Habilidades a obtener con el curso:* muestra una lista de habilidades a obtener por parte del estudiante al finalizar el curso.
- *Glosario del Curso:* Lista el glosario de términos asociado al curso.

- *Bibliografía del Curso:* Muestra la Bibliografía relacionada al curso.
- *Actividades del Curso:* Muestra las actividades que recomienda seguir el curso.
- *Actividades de Evaluación:* Lista en una tabla las evaluaciones que debe rendir el estudiante.
- *Descripción del Curso:* Muestra una breve descripción del curso seleccionado.

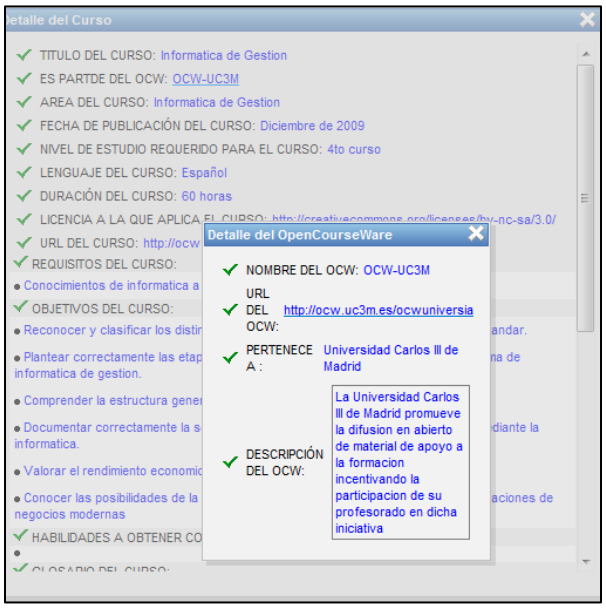


Figura 40 DETALLE DEL OPEN COURSE WARE

A continuación se detalla la definición de cada label presentado en este panel:

- *URL del OCW:* Muestra el Link del OCW, si este link es seleccionado abrirá una nueva ventana en el navegador con la página principal del OCW.
- *Pertenece A:* Muestra el Nombre de la Institución Educativa propietaria del sitio OCW.
- *Descripción del OCW:* Muestra una descripción del sitio OCW.

Figura 41 LISTA DE AUTORES

A continuación se detalla la definición de cada label presentado en este panel:

- Autores del Curso: Muestra los autores del curso.
- Son Miembros de la Universidad: Muestra el nombre de la Universidad a la cual pertenecen los autores.

Buscar Recurso: Permite buscar un recurso determinado, recibe por entrada una parte o todo el nombre de un recurso, por ejemplo si ingresamos en la caja de texto la palabra "inge" obtendremos todos los recursos que se relacionen con ese nombre, además se mostrara el total de recursos listados, y se mostrara un botón para volver a listar todos los resultados. Este proceso se lo puede observar en la Figura citada a continuación.

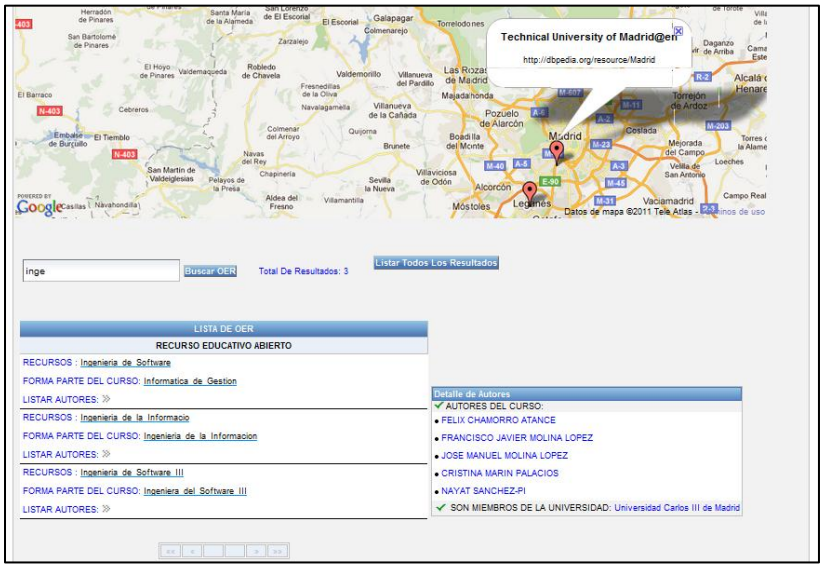


Figura 42 BUSCAR OER

Para obtener este resultado se trabajó con una consulta SPARQL con la siguiente sintaxis FILTER regex(str (?title = “ ” 'i')); para no tener que ingresar todo el nombre completo de un recurso.

7.0. Discusiones Generales

"Todos somos muy ignorantes. Lo que ocurre es que no todos ignoramos las mismas cosas."

Albert Einstein

En lo que respecta a los Gestores de RDF se pueden obtener las siguientes conclusiones.

- Oracle se diferencia de los demás gestores en que no cuenta con una capa específica para el almacenamiento de tripletas RDF, lo que puede ocasionarle problemas de escalabilidad, al trabajar con una gran cantidad de datos a diferencia de 4Store que permite trabajar con una gran cantidad de datos.
- Todos los gestores RDF soportan el lenguaje de recuperación de información SPARQL, pero no todos lo usan.
- Sesame posee una capa de almacenamiento específica para trabajar con RDF, por lo que el coste de trabajar con datos es menor, frente a Oracle cuyo coste es alto y no posee una Arquitectura específica para trabajar con RDF.
- Oracle y Sesame permiten definir el número de índices a crear.
- Oracle por ser un gestor de base de datos relacional, incluye los procedimientos necesarios para garantizar la seguridad de la información de los datos.
- En cuanto a la rapidez de inserción de tripletas 4Store y Sesame son los más óptimos.
- En cuanto al espacio en disco utilizado al almacenar un moderado número de tripletas 4Store, Sesame y Oracle son los más óptimos respectivamente.
- En cuanto al coste del CPU a la hora de realizar consultas, Sesame y Oracle son los más óptimos.
- Oracle permite trabajar con inferencia sobre los datos, lo que constituye una gran ventaja frente a gestores RDF como 4store.
- Oracle ha sido testeado hasta con 1068 millones de tripletas. (Wu, et al, 2008). En esta investigación se lo testeo con 3500 tripletas y Oracle ha demostrado una gran escalabilidad, muy contrario a lo que se afirma en varias investigaciones donde se menciona que Oracle no posee una gran escalabilidad.
- Sesame posee un tiempo de respuesta seis veces más rápido que Jena, sin embargo Jena es capaz de almacenar y trabajar nativamente con casi tres veces más sentencias que Sesame.
- Los sistemas propietarios como Oracle, Virtuoso y AllegroGraph poseen una gran potencia y escalabilidad, según (Torre, 2009), estos sistemas han superado los 1000 millones de sentencias, mientras que los otros no pasa nativamente de los 200, con lo que se puede concluir que las herramientas que mayor escalabilidad aportan se basan en almacenamiento genérico en bases de datos propietarias o en almacenamiento mediante arboles B+.
- Para almacenar pequeñas cantidades de datos, es suficiente la utilización de gestores nativos que almacenen los datos en disco o en archivos.

- En lo relacionado con la instalación, de cada uno de los sistemas, esta varía significativamente dependiendo de la plataforma.
- En cuanto a la facilidad de utilización se recomienda utilizar Jena, o 4Store. Oracle es el gestor RDF que más esfuerzo conllevó, pues se lo testeo en Ubuntu y Mac OS X.
- En cuanto a documentación AllegroGraph, Jena y Sesame destacan.
- Es importante recalcar que se ha probado todos los RDF Triple Store System que en este capítulo se mencionan, en varias plataformas y arquitecturas.
- En varios estudios afines que se han realizado no existe un acuerdo mayoritario sobre cuál es el mejor Gestor RDF.
- No se concluye, cual es el mejor Triple Store, ya que esto depende de varios aspectos como el dominio de la aplicación a implementar, dominio que se debe establecer correctamente, además concluir sobre el mejor Triple Store no es el tema de esta investigación.

En lo relacionado a la obtención de datos OERs de la web se menciona lo siguiente:

- El mayor obstáculo para la obtención y procesamiento de datos, sin duda alguna es la heterogeneidad entre los mismos repositorios de OER, en este caso los OCW, por la razón que es muy difícil establecer, un patrón que nos permita la obtención de datos automáticamente, esto debido a que los sitios OCW, no respetan el uso de vocabularios para la descripción de los metadatos de los recursos.
- Muchos OCW, no deberían llevar ese nombre, pues están equivocados en la correcta definición de lo que implica que un sitio OCW debe ofrecer. Muchos cursos de los analizados son cursos creados solamente para los estudiantes de una determinada Universidad, puesto que piden un previo registro con el dominio de la Institución, para poder acceder a los recursos.
- La información incluida en los RSS es muy importante para recuperar información de los cursos OCW, sin embargo cuentan con muy poca información.
- Para lograr la interoperabilidad en eLearning es necesario definir una sintaxis y semántica común para la descripción de los recursos educativos (estándares, tecnologías de la SW y Linked Data como RDF que proporciona un soporte para metadatos flexible y abierto que permitirá la interconexión con otros sistemas similares).

- La heterogeneidad dentro del consumo de información es hoy en día un nudo operacional si se quiere mejorar la eficiencia.
- El problema de la interoperabilidad no es un problema 100 % tecnológico, influyen también otros aspectos como problemas de ideologías.

En lo relacionado a los vocabularios Linked Data se menciona lo siguiente:

- En base a lo analizado en este capítulo se puede concluir que la reutilización de vocabularios existentes, tiene la gran ventaja de reunir a varios dominios dentro de RDF, convirtiéndolos en más reutilizables a los datos. Gracias a la reutilización de los vocabularios, los datos ya no están aislados ni encerrados dentro de un contexto único, diseñado para un único uso.
- Lo ideal sería utilizar vocabularios estándar, aunque si no existieran se definirían ad-hoc. De esta forma se dota de semántica a la información.
- Si no utilizamos vocabularios asociados a la representación de información, esta no tiene sentido semántico.
- El vocabulario usado, está claramente especificado y es reutilizable.

En lo que respecta al aplicativo de generación de RDFS, se mencionan los siguientes puntos:

- Existe una discusión en cuanto el aplicativo admite múltiples consultas, esto se lo realizo con el objetivo de optimizar recursos y permita ejecutar varias sentencias en una misma cadena, sin embargo es importante mencionar que este atributo es importante para la seguridad, debido a que un usuario podría realizar una inyección de SQL, este parámetro está configurado con la siguiente sintaxis allowMultiQueries=true, si este parámetro es establecido en false (por defecto) no permitirá enviar en una misma cadena varias sentencias SQL.
- El aplicativo permite automatizar el proceso de generación de RDF. Actualmente en el mercado existen varias herramientas para la generación de RDF, pero no hay una que permita la generación automática a este nivel.
- El RDF generado es consiste y tiene significado definido.
- El RDF resultante, ha sido validado con el Validador de RDF recomendado por la W3C, lo que le da solvencia al resultado de la aplicación.

- Es importante mencionar que la calidad del RDF dependerá únicamente de la estructura de las fuentes de datos estructuradas.
- Para esta primera versión se ha tomado en cuenta los vocabularios más representativos, y además es importante recalcar que el aplicativo actualmente no le brinda la posibilidad al usuario que crear su propio vocabulario.
- No se ha considerado estándares de diseño o de accesibilidad. Al tratarse de una primera versión estos temas son redundantes.

En lo que respecta al visualizador, se mencionan los siguientes puntos:

- Utilizando tecnologías semánticas como RDF y SPARQL, se ha conseguido modelar, unificar y exponer los datos.
- La aplicación permite visualizar de manera sencilla datos de los OERs.
- El objetivo del aplicativo es demostrar como una gran cantidad de datos RDF puede ser visualizados como datos no RDF.
- Es importante mencionar que el aplicativo depende estrictamente de las consultas SPARQL implementadas, lo que significa que no es factible la construcción de un Visualizador de datos RDF universal al nivel del Visualizador presentado en este capítulo.
- Es importante mencionar que se ha trabajado únicamente con SPARQL, para recuperar la información presentada por el Visualizador, y se ha trabajo las cadenas para que muestre datos sin caracteres como los caracteres "_" o "#", que son muy comunes en las tripletas RDF.
- Al usuario se le debe presentar valores que pueda entender, por eso es necesario procesar los resultados de una consulta de manera que el usuario no tenga que preguntarse el porqué del formato de los resultados.
- En general, nuestra propuesta pretende ser un esfuerzo por incorporar ideas de consumo y presentación de datos RDF.

8.0. Conclusiones, Recomendaciones y Trabajos Futuros

"Y guarden estas palabras Que les digo al terminar:
En mi obra he de continuar hasta dársela concluida.
Si el ingenio o si la vida No me llegan a faltar."

José Hernández
La vuelta de Martín Fierro, Canto XXXIII

8.1. CONCLUSIONES

Según el trabajo realizado se puede concluir que:

- Se ha realizado un trabajo de investigación sobre la Evolución de la web, poniendo énfasis en la publicación de datos enlazados de recursos educativos, demostrando que se es factible la publicación de datos procesados en la web, utilizando para ellos las principales tecnologías, estándares e iniciativas de la web semántica cumpliendo así el objetivo principal del proyecto.
- En base a la investigación realizada sobre Gestores RDF, se encontró que este depende directamente del dominio de la aplicación o sistema a implementar.
- Como resultado de la investigación sobre almacenes RDF se encontró que muy pocos Gestores RDF soportan la visión de la web Semántica en cuanto a escalabilidad, operatividad en las respuestas y distribución de información; dado que procesan información sin razonar sobre ella. Es muy importante que los gestores RDF incluyan una mayor capacidad de inferencia.
- Se ha demostrado que se puede trabajar con una base de datos Relacional como MySQL para almacenar datos RDF.
- Se concluye que Jena es el mejor framework actual para trabajar con datos RDF, definiéndolo personalmente como el puente entre el almacén de datos y el Usuario, el mismo que se encuentra sostenido por pilares como Java, SPARQL, RDF, entre otras tecnologías.
- Una de las principales utilidades que se ha obtenido con el desarrollo del proyecto es dar un punto de partida para aprender y experimentar como se puede integrar tecnologías semánticas a base de datos, logrando de esta forma mayores beneficios a futuras investigaciones.
- Se encontró que la generación de un script a partir del modelo conceptual Entidad-Relación se debe hacer tanto para obtener diseños de tablas relacionales como para generar Descripción de Contenidos RDF.
- Con el aplicativo construido se ha demostrado que es factible la generación de RDF desde fuentes de datos estructuradas.
- Se concluye que la tarea más difícil en la visión de enlazar datos constituye la extracción de datos desde la web, en nuestro caso la naturaleza heterogénea de los OERs constituyo una gran barrera.
- Se concluye que la utilización de estándares como RDF, SPARQL, son los más aptos y con los cuales se obtuvo mejores resultados a la hora de acceder a la información.

- El resultado obtenido en esta investigación al generar y utilizar datos enlazados procedentes de fuentes heterogéneas como los OCW, creara nuevas oportunidades de investigación, aunque aún falta mucho camino por recorrer como trabajar en definir reglas que faciliten la homogeneidad entre Data Sources.
- Se demostró que la utilización de los principios de Linked para publicar datos sobre la web, logrará crear un espacio global sobre la Web, lo que contribuirá a incrementar el número de tripletas RDF en la Web.

8.2. RECOMENDACIONES

- En lo relacionado, los Gestores RDF, se recomienda que antes de elegir algún gestor RDF determinado, se analice a profundidad el dominio del servicio, aplicación o sistema que se dese implementar.
- Se recomienda manejar sistemas operativos con una arquitectura de 64 bits, y para realizar pruebas se recomienda manejar máquinas virtuales.
- Antes de instalar algún gestor RDF, se recomienda tener en cuenta los requerimientos mínimos expuestos por las organizaciones, además es importante instalar todas las librerías que se indican en el presente trabajo ya que esto puede ocasionar problemas para el entorno de preinstalación de los RDF Triple Store System.
- Para la configuración del entorno semántico en Oracle se deben ejecutar los scripts en el orden en el que se especifican en este trabajo, ya que algunos dependen de otros y en el caso de no hacerlo puede ocurrir un error en dicho proceso.
- En lo relacionado a la descripción de vocabularios se recomienda reutilizar vocabularios consensuados como FOAF, Dublin Core y AIISO.
- Antes de realizar un RDFS, se debe implementar un modelo de entidad relación con el objetivo de obtener una visión mucho más amplia de lo que se desea implementar.
- Se recomienda para el desarrollo de RDFS, hacerlo con la herramienta COE CMapTools, por simplicidad.
- Se recomienda crear una base de datos específica para almacenar las tripletas RDF, ya debido a que cuando se establece la conexión se recuperan todas las tablas que pertenecen a la base de datos.
- Para la validación del RDF se recomienda utilizar el servicio de validación de RDF[88] que ofrece la W3C.

[88] http://www.w3.org/RDF/Validator/

8.3. TRABAJOS FUTUROS

- Investigación y aplicación de los razonadores que se pueden utilizar desde el API Jena sobre las tripletas que están almacenadas en una base de datos.
- Consumir dantos enlazados de OERs de las principales universidades del mundo y publicarlas para el uso de los estudiantes de La Universidad Particular de Loja en primera instancia a través de su Entorno de Aprendizaje Virtual (EVA).
- Desarrollar un Buscador web a nivel macro, que permita a los estudiantes y profesores de todo el mundo encontrar cursos relacionados, temas específicos o encontrar áreas de investigación relacionadas.
- Desarrollar en base a los resultados obtenidos, un proyecto a nivel macro que forme parte de la nube de Open Linked Data.
- Desarrollar una aplicación que permita la recuperación automática de metadatos de recursos directamente desde páginas HTML.
- Mejorar la versión 1 del aplicativo RDFLoxaAdmin para poder extraer y procesar información de más fuentes de datos estructuradas, con el objetivo de convertirse en una herramienta útil a la hora de generar RDF.
- Elaborar políticas de publicación de OERs, dirigidas a todos los OCWs, con el objetivo de estandarizar la publicación de Recursos Educativos.
- Contar con un Dataset integrado y disponible como RDF y de acceso público a través de descargas e información relativa, vinculando datos de varios Dataset, con el objetivo de enriquecer la información relacionada a un recurso educativo determinado.

8.0. BIBLIOGRAFÍA

1. (Aduna B.V, 2006). Aduna B.V., Sirma AI Ltd. (2006). *User Guide for Sesame*. (OpenRDF.org) Recuperado el 26 de February de 2011, de http://www.openrdf.org/doc/sesame/users/index.html
2. (Apache, 2011) Apache (2011). *POI*. Recuperado el 03 de 04 de 2011, de http://poi.apache.org/apidocs/index.html
3. (Archives Hub, (n.d)) Archives Hub (n.d). *Archives Hub*. Recuperado el 27 de February de 2011, de http://archiveshub.ac.uk/about/
4. (Beckett, 2011 a) Beckett, D. (2011). *Raptor RDF Syntax Library*. Recuperado el 29 de June de 2011, de http://librdf.org/raptor/
5. (Beckett, 2011 b) Beckett, D. (2011). *Rasqal RDF Query Library*. Recuperado el 29 de June de 2011, de http://librdf.org/rasqal/
6. (Berners-Lee, 2007) Berners-Lee. (2007). *Hearing on the "Digital Future of the United States: Part I -- The Future of the World Wide Web"*. Recuperado el 22 de February de 2011, de Testimony of Sir Timothy Berners-Lee: http://dig.csail.mit.edu/2007/03/01-ushouse-future-of-the-web.pdf
7. (Berners-Lee, 1996) Berners-Lee, T. (1996). *The World Wide Web: Past, Present and Future*. Recuperado el 21 de February de 2011, de http://www.w3.org/People/Berners-Lee/1996/ppf.html
8. (Berners-Lee, 2000) Berners-Lee, T. (2000). *Tejiendo La Red*. (S. Xxi, Ed.) Madrid.
9. (Haslhofer, et al, 2011) Bernhard, H.; Elahed, M.; Bernhard, S.; Stefan, Z. (2011). *Europeana RDF Store Report*. Recuperado el 29 de June de 2011, de Faculty of Computer Science University of Vienna: www.europeanaconnect.eu/documents/europeana_ts_report.pdf
10. (Bizer, 2010) Bizer, C. (2010). *Linked Data*. Recuperado el 26 de February de 2011, de Deutsches Kulturerbe auf dem Weg in die Europeana: http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/research/publications/Bizer-Europeana-LinkedData-Talk.pdf
11. (W3C, 2004 c) W3C. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. Recuperado el 02 de January de 2011, de http://www.w3.org/TR/rdf-schema/
12. (Cámara, 2002) Cámara, J. C. (2002). *Learning Metadata Standards UPF*. Recuperado el 29 de June de 2011, de Learning Metadata Standards UPF: http://es.athabascau.ca/Technologies/xml_vs_rdf.htm/view

13. (Bizer, et al, 2008) Bizer, C.; Cyganiak, R.; Healt, T. (2008). *How to Publish Linked Data on the Web*. Recuperado el 23 de December de 2010, de http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/20070727/

14. (Bizer & GauB, 2007) Bizer, C., GauB, T.. (09 de January de 2007). *Disco - Hyperdata Browser*. Recuperado el 27 de February de 2011, de http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/

15. Chris Bizer, Tobias Gauß, Richard Cyganiak, Olaf Harting. (18 de November de 2009). *Semantic Web Client Library* . Recuperado el 22 de February de 2011, de Freie Universitat Berlin: http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/

16. (Bizer et al, 2011) Bizer, C.; Heath T,; Berners-Lee, T.; Hausenblas, M.; Cyganiak, R. (2011). *Linked Data on the Web Workshop (LDOW 2011)*. Recuperado el 20 de June de 2011, de WWW 2011: http://events.linkeddata.org/ldow2011/slides/ldow2011-slides-intro.pdf

17. (Bizer et al, 2009) Bizer C., Heath T., Berners-Lee, T. (2009). *Linked Data - The Story So Far*. Recuperado el 23 de February de 2011, de http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf

18. (Copac, 2011) Copac. (2011). *Copac National, Academic, and Specialist Library Catalogue*. Recuperado el 08 de March de 2011, de http://copac.ac.uk/

19. (Creative Commons, 2008) Creative Commons. (2008). *Describing Copyright in RDF*. Recuperado el 20 de June de 2011, de Creative Commons Rights Expression Language: http://creativecommons.org/ns#

20. (Cobo & Pardo, 2007), COBO C. y PARDO H.. (2007). *Planeta web 2.0 Inteligencia Colectiva o medios fast food*. Recuperado el 23 de February de 2011, de http://www.planetaweb2.net/

21. (Cueva, 2008) Cueva, S. (2008). *WEB SEMANTICA: Principios y Estandares*. Recuperado el 24 de February de 2011, de www.utpl.edu.ec/gcblog/wp-content/uploads/web-semantica.pdf

22. (Brickley & Libby, 2010), Brickley, D.; Miller, L. (2010). *FOAF Vocabulary Specification 0.98*. Recuperado el 25 de June de 2011, de http://xmlns.com/foaf/spec/

23. (Domain Counts, 2011) Domain Counts. (2011). *Domain Counts*. Recuperado el 22 de February de 2011, de Domain Counts & Internet Statistics: http://www.domaintools.com/internet-statistics/

24. (Tóvar, et al, 2011) Tóvar, E.; Cueva, S.; Rodriguez, G. (2011). *Implementation of Social and Semantic Tools into Open Educational Resources Production*. Recuperado el 29 de June de 2011, de http://www.psut.edu.jo/sites/EDUCON/program/contribution1295_b.pdf

25. (W3C, 2007) W3C (2007). Recuperado el 22 de December de 2010, de SPARQL Query Language for RDF: http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/

26. (EURECOM, 2010) European Future Internet Portal. (2010). *Future Internet Assembly*. Recuperado el 29 de June de 2011, de Linked Data in the Future Internet: http://www.future-internet.eu/home/future-internet-assembly/ghent-dec-2010/session-i-linked-open-data-i.html

27. (FOAF, 2010) FOAF. (2010). *Index*. Recuperado el 23 de February de 2011, de www.foaf.qdos.com

28. (Franz Inc, 2011 b) Franz Inc, (2011). *AllegroGraph 4.3 Introduction*. Recuperado el 29 de June de 2011, de http://www.franz.com/agraph/support/documentation/current/agraph-introduction.html

29. (Franz Inc., 2011 a) Franz Inc. (2011). *AllegroGraph RDFStore 4.2*. Recuperado el 27 de February de 2011, de AllegroGraph 4.2 Introduction : http://www.franz.com/agraph/support/documentation/v4/agraph-introduction.html#header2-52

30. (Garlik, 2009 a) Garlik. (2009). *4Store*. Recuperado el 22 de February de 2011, de Garlik: www.4store.org

31. (Garlik, 2009 b), Garlik. (2009). *Projects Using 4Store*. Recuperado el 23 de February de 2011, de www.4store.org/projects

32. (Garlik, 2010) Garlik. (2010). *DataPatrol for Individuals*. Recuperado el 23 de February de 2011, de www.garlik.com/dpindividuals.php

33. (Hildebrand, 2006) Hildebrand, M. (2006). *Facet.* Recuperado el 27 de February de 2011, de http://slashfacet.semanticweb.org/

34. (howstuffworks, 2008) howstuffworks. (2008). *Web 1.0 Defined*. Recuperado el 23 de February de 2011, de http://computer.howstuffworks.com/web-101.htm

35. (IEEE, 1991) IEEE. (1991). A Compilation of IEEE Standard Computer Glossaries. *Computer Dictionary*.

36. (Herman, 2007), Herman, I. (2007). *Introduction to the Semantic Web*. Recuperado el 01 de December de 2010, de http://www.w3.org/2007/Talks/0423-Stavanger-IH/

37. (JISC, 2011) JISC. (2011). *About JISC*. Recuperado el 27 de February de 2011, de http://www.jisc.ac.uk/aboutus.aspx

38. (LOCAH Project, 2011) LOCAH Project. (2011). *LOCAH Project*. Recuperado el 27 de February de 2011, de Linked Open Copac Archives Hub: http://blogs.ukoln.ac.uk/locah/about/

39. (sourceforge, 2008) LP Hewlett-Packard Development Compan. (2008). *Jena contributions page*. Obtenido de http://jena.sourceforge.net/contrib/contributions.html

40. (SourceForge, 2009) LP Hewlett-Packard Development Compan. (2009). *Jena – A Semantic Web Framework for Java*. Recuperado el 01 de 02 de 2011, de http://jena.sourceforge.net/

41. (OCW Consortium, 2011), MITOPENCOURSEWARE. (2011). *MITOPENCOURSEWARE*. Recuperado el 29 de June de 2011, de http://ocw.mit.edu

42. (Shabir & Clarke, 2009) Shabir, N.; Clarke, C. (2009). *Using Linked Data as a basis for a Learning Resource Recommendation System.* Recuperado el 10 de March de 2011, de http://eprints.ecs.soton.ac.uk/18053/1/semhe09_submission_5-1.pdf

43. (Newman, 2010) Newman, D. R. (2010). *The Building and Application of a Semantic e-Research Society.* Recuperado el 20 de June de 2011, de http://users.ecs.soton.ac.uk/drn/phd/Thesis.pdf

44. (Spivack, 2007) Spivack Nova. *www.novaspivack.com*. (2007). Recuperado el 25 de February de 2011, de How the WebOS Evolves: http://novaspivack.typepad.com/nova_spivacks_weblog/2007/02/steps_towards_a.html

45. (OECD, 2007) OECD. (2007). *Giving Knowledge for Free The Emergence of Open Educational Resources*. Recuperado el 15 de March de 2011, de http://www.oecd.org/dataoecd/35/7/38654317.pdf

46. (Ontology Engineer IngGroup, 2010) Ontology Engineer IngGroup. (2010). *NOR2O: a Library for Transforming Non-Ontological Resources to Ontologies*. Recuperado el 27 de February de 2011, de http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/downloads/57-nor2o

47. (OpenLink Software, 2009 a) OpenLink Software. (2009). *Implementing a SPARQL compliant RDF Triple Store using a SQL-ORDBMS*. Recuperado el 28 de February de 2011, de http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSRDFWP

48. (OpenLink Software, 2009 b) OpenLink Software (2009). *RDF Data Access and Data Management.* Recuperado el 29 de June de 2011, de SPARQL: http://docs.openlinksw.com/virtuoso/rdfsparql.html

49. (W3C, 1999) W3C (1999). *Resource Description Framework (RDF) Model and Syntax Specification*. Recuperado el 25 de December de 2010, de W3C: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

50. (ORACLE, 2009) ORACLE. (2009). *Semantic Technologies Developer's Guide 11g Realese 1 (11.1).* Recuperado el 28 de February de 2011, de http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28397.pdf

51. (ORACLE, 2010) ORACLE. (2010). *Oracle Database Semantic Technologies Overview*. Recuperado el 29 de June de 2011, de http://download.oracle.com/docs/cd/E18283_01/appdev.112/e11828/sdo_rdf_concepts.htm

52. (Erlin & Mikhailo, 2008) Erling, O.; Mikhailo, I. (2008). *RDF Support in the Virtuoso DBMS*. Recuperado el 29 de June de 2011, de http://www.openlinksw.com/dataspace/dav/wiki/Main/VOSArticleRDF/rdfdb1.pdf

53. (Corcho & Gomez, 2010) Corcho O.; Gómez A. (2010). *Mini-curso sobre Linked Data*. Recuperado el 23 de January de 2011, de http://www.slideshare.net/ocorcho/linked-data-tutorial-florianpolis

54. (Nelson & Romero, 2010), Piedra, N.; Romero, A.. (2010). *CALIDAD DE CONTENIDOS EN OCW*. Recuperado el 29 de June de 2011, de http://reposital.cuaed.unam.mx:8080/jspui/bitstream/123456789/2706/1/romero_piedra_calidad_de_contenidos_ocw.pdf

55. (Reynolds, 2010) Reynolds, D. (2010). *Jena 2 Inference support*. Recuperado el 27 de February de 2011, de http://jena.sourceforge.net/inference/

56. (Styles & Shabir, 2008) Styles, R.; Shabir, N. (2008). *Academic Institution Internal Structure Ontology (AIISO)*. Recuperado el June de 20 de 2011, de http://vocab.org/aiiso/schema

57. (Heymans, et al., 2007) Heymans, S.; Li, M.; Anicic, D. (2007). *Ontology Reasoning with Large Data Repositories*. Springer-Verlag.

58. (Semantic Web, 2009) Semantic Web. (2009). *OpenLink RDF Browser*. Recuperado el 27 de February de 2011, de http://semanticweb.org/wiki/OpenLink_RDF_Browser

59. (Sesame, 2010) Sesame. (2010). *System documentation for Sesame 2.x (DRAFT)*. Recuperado el 29 de June de 2011, de OpenRDF.org: http://www.openrdf.org/doc/sesame2/system/

60. (SMILE, 2007) SMILE. (2007). *MIT OpenCourseWare RDFizer*. Recuperado el 17 de February de 2011, de http://simile.mit.edu/wiki/OCW_RDFizer

61. (Gruber, 1993), Gruber T. (1993). *A Translation Approach to Portable Ontology Specifications*.

62. (Berners-Lee, et al, 2005) Berners-Lee, T.; Masinter, L.; Fielding, R. (2005). *Uniform Resource Identifier (URI): Generic Syntax*. Recuperado el 29 de June de 2011, de http://tools.ietf.org/html/rfc3986

63. (Berners-Lee, 2006) Berners-Lee, T. (2006). *W3C*. (W3C) Recuperado el 23 de February de 2011, de Design Issues for Linked Data: http://www.w3.org/DesignIssues/LinkedData.html

64. (Berners-Lee, et al, 2001). Berners-Lee T. , Hendler J, Lassila O. (2001). *Scientific American*. Recuperado el 15 de October de 2010, de The Semantic Web: http://www.scientificamerican.com/article.cfm?id=the-semantic-web

65. (O'Reilly, 2005) O'Reilly, T. (2005). *O'Reilly*. Recuperado el 22 de february de 2011, de What Is Web 2.0: http://oreilly.com/web2/archive/what-is-web-20.html

66. (Healh & Bizer, 2011) Heath, T.; Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Recuperado el 29 de June de 2011, de http://linkeddatabook.com/editions/1.0/

67. (Torre, 2009) Torre, P. d. (2009). *Almacenes de Datos para la Web Semantica*. Recuperado el 29 de June de 2011, de Universidad de Sevilla: www.lsi.us.es/docs/.../Torre-Moreno-Pablo-Memoria-Investigacion.pdf

68. (UNESCO, 2011) UNESCO. (2011). *Open Educational Resources*. Recuperado el 29 de June de 2011, de http://www.unesco.org/new/en/communication-and-information/access-to-knowledge/open-educational-resources/

69. (Universia, 2011) Universia. (2011). *Universia España*. Recuperado el 29 de June de 2011, de http://www.universia.es

70. (UNIVESITAT KOBLENZ- LANDAU, 2008) UNIVESITAT KOBLENZ- LANDAU. (2008). *Lena*. Recuperado el 27 de February de 2011, de http://www.uni-koblenz-landau.de/koblenz/fb4/institute/IFI/AGStaab/Research/systeme/lena

71. (Valkenburg, 20011) Valkenburg, I. W. (23 de February de 2011). *OpenCourseWare and Standards*. Recuperado el 29 de June de 2011, de http://www.imsglobal.org/meetings/Utrecht2011/Opencourseware%20Consortium.pdf

72. W3C. (06 de September de 2010). *Extensible Markup Language (XML)*. Recuperado el 23 de February de 2011, de http://www.w3.org/XML/

73. (W3C, 2011 b) W3C. (2011). *LargeTripleStores*. Recuperado el 2011 de June de 2011, de http://www.w3.org/wiki/LargeTripleStores

74. (W3C, 2011 c) W3C. (2011). *LinkedOpenData W3C SWEO Community Project*. (W3C) Recuperado el 27 de February de 2011, de http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

75. (W3C, 2011 a), W3C. (2011). *SemanticWebTools*. Recuperado el 23 de February de 2011, de http://www.w3.org/wiki/SemanticWebTools

76. (W3C, 2004 a) W3C, (2004). *Using Qualified Names (QNames) as Identifiers in XML Content*. Recuperado el 29 de June de 2011, de http://www.w3.org/2001/tag/doc/qnameids

77. (W3C, 2004 b) W3C, (2004). *RDF Primer*. Recuperado el 29 de June de 2011, de http://www.w3.org/TR/rdf-syntax/

78. (Wenk, 2010) Wenk, B. (2010). *Open educational resources (OER) inspire teaching and learning*. Recuperado el 15 de March de 2010

79. (Wu, et al, 2008) Wu, Z.; Eadon, G.; Das, Souripriya, E. (2008). *Implementing an Inference Engine for RDFS/OWL Constructs and User-Defined Rules in Oracle*. Recuperado el 29 de June de 2011, de Oracle: http://www.oracle.com/technetwork/database/options/semantic-tech/icde-2008-inf-engine-130436.pdf

9.0. GLOSARIO

A

Agente: Programa capaz de recolectar información de la Web e, imprescindiblemente, compartirla con otros agentes.

Anotaciones Semánticas: ver Sección 1.2.3.1, pág. 26.

Apache: Servidor de páginas Web desarrollado por la Apache Software Fundation, organización formada por miles de voluntarios que colaboran para la creación de software de libre distribución.

API: Interfaz de programación de aplicaciones.

B

Búsqueda sintáctica es un procedimiento de búsqueda donde el resultado son varios documentos que contienen los términos de la consulta.

D

DBMS (Sistema de gestión de base de datos): Agrupación de programas que sirven para definir, construir y manipular una base de datos.

F

Framework: Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

G

GPL: (Licencia Pública General) es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre.

Grafo: Estructura de datos utilizada en algunos lenguajes de programación, en la cual cada elemento puede tener uno o varios predecesores y uno varios sucesores.

H

HTML: Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

HTTP: Es el protocolo de la Web (WWW), usado en cada transacción.

I

Interoperabilidad: Habilidad de dos o más sistemas o componentes para intercambiar información y para usar la información que ha sido intercambiada.

J

JDBC (Java Database Connectivity): API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Java: Plataforma de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella pueda ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales.

JavaBeans: JavaBeans es un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java.

JavaScript: Lenguaje desarrollado por Netscape. Aunque es parecido a Java se diferencia de él en que los programas están incorporados en el archivo

Jena: Framework desarrollado por HP Labs para manipular metadatos desde una aplicación Java.

JSP: Java Server Pages (JSP) es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java

M

Metadatos: datos sobre datos

MySQL: MySQL es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto.

N

Namespace: Vocabulario definido dentro de un URI (Identificado de recursos Universal).

N-Triples: Documento N-Triples es una secuencia de caracteres US-ASCII y es definido por la gramática ntripleDoc.

N3: Lenguaje de programación equivalente a RDF

O

OCW: Es una publicación digital libre y abierta de alta calidad de nivel universitario, estos materiales están organizados como cursos.

ODBC(Open Database Connectivity): es un estándar de acceso a Bases de datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos almacene los datos.

OER: Recursos educativos Abiertos

Ontologías: ver Sección 1.2.3, pág. 25.

Oracle: Sistema de administración de base de datos fabricado por Oracle Corporation.

ORDBMS (Sistema de Gestión de base de datos con objetos relacionales): son aquellas que además de manejar datos numéricos de distinto tipo, strings, imágenes, etc., incluyen objetos como parte de las tablas de bases de datos, a los que maneja como otro tipo de datos.

OWL: ver Sección 1.2.3, pág. 26

P

Plugin: Programa de ordenador que interactúa con otro programa para aportarle una función o utilidad específica, generalmente muy específica.

R

RDBMS: Sistema de gestión de base de datos relacional, su enfoque consiste en que contiene bases de datos relacionales cuyos datos residen en la base de datos y se manipula de forma colectiva con las consultas en un lenguaje de consulta.

RDQL: Lenguaje de programación para manejar datos RDF.

RDF: es el estándar de la W3C para los datos interoperables legibles por máquinas,

RDF Triple Store System: Repositorios de almacenamiento para RDF.

Repositorio de objetos de aprendizaje: Repositorio que almacena recursos educativos en formato electrónico y/o sus metadatos.

S

Semántica: Se refiere al significado, o interpretación del significado de un determinado recurso.

Servlet: Objetos que corren dentro del contexto de un servidor de aplicaciones.

Shell: Parte fundamental de un sistema operativo encargada de ejecutar las órdenes básicas para el manejo del sistema.

SQL: Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

SPARQL: Es un lenguaje de recuperación web y sirve para recuperar recursos electrónicos desde la Web.

T

Tesauros documentales: Tipo de lenguaje combinatorio que consta de listas de términos que representan un ámbito científico y técnico determinado y que posee una serie de relaciones semánticas entre los términos que lo conforman.

U

URL: Acrónimo de "Universal Resource Locutor" (Localizador Universal de Recursos / Identificador Universal de Recursos). Sistema unificado de identificación de recursos en la red.

URI: Cadena corta de caracteres que identifica inequívocamente un recurso.

X

XML: Acrónimo del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web.

W

Web Semántica: ver Sección 1.2.3, pág. 23.

W3C. World Wide Web Consortium, abreviadamente W3C, es una organización que produce estándares para la World Wide Web.

ANEXOS

... Una vez que has dicho algo, ahí queda,
y debes aceptar las consecuencias.

De la Reina Roja a Alicia
Cap IX: Alicia Reina
Lewis Carroll

ANEXO 1: MANUAL DE INSTALACIÓN DE ORACLE SPATIAL 11g EN UBUNTU 10.04

1. Preparación de la instalación

1.1. Actualizar el sistema operativo con los paquetes requeridos, además hay que instalar java, en el caso de tener problemas con la instalación de java se sugiere en orígenes de software – Otro Software marcar la primera opción.

1.2. Autenticarnos como root:

- sudo –s

1.3. Descargamos los siguientes paquetes que son complementarios para la instalación:
apt-get install gcc make binutils gawk x11-utils rpm alien ksh lsb-rpm

1.4. Ubicarse en el siguiente directorio y realizar los siguiente:

- cd /bin
- ls -l /bin/sh

```
root@ecc-laptop:~# cd /bin/
root@ecc-laptop:/bin# ls -l /bin/sh
lrwxrwxrwx 1 root root 4 2010-05-11 03:55 /bin/sh -> dash
root@ecc-laptop:/bin# ln -sf /bin/bash /bin/sh
root@ecc-laptop:/bin# ls -l /bin/sh
lrwxrwxrwx 1 root root 9 2010-05-11 05:56 /bin/sh -> /bin/bash
root@ecc-laptop:/bin#
```

Figura 43 CONFIGURACIÓN DE DIRECTORIOS

1.5. Se procede a crear los grupos para la instalación en este caso son oinstall, dba y nobody.

- addgroup oinstall
- addgroup dba
- addgroup nobody

```
root@ecc-laptop:~# addgroup oinstall
Añadiendo el grupo «oinstall» (GID 1001) ...
Hecho.
root@ecc-laptop:~# addgroup dba
Añadiendo el grupo «dba» (GID 1002) ...
Hecho.
root@ecc-laptop:~# addgroup nobody
Añadiendo el grupo «nobody» (GID 1003) ...
Hecho.
```

Figura 44 CREACIÓN DE GRUPOS

1.6. Asignamos los permisos correspondientes a los grupos y se crea el usuario Oracle:

-g es primario

-G es secundario

- usermod -g nobody nobody
- useradd -g oinstall -G dba -p password -d /home/oracle -s /bin/bash oracle

```
root@ecc-laptop:~# usermod -g nobody nobody
root@ecc-laptop:~# useradd -g oinstall -G dba -p password -d /home/oracle -s /bi
n/bash oracle
```

Figura 45 ASIGNACIÓN DE PERMISOS

1.7. Creamos el directorio el siguiente directorio

- mkdir /home/oracle
- chown -R oracle:dba /home/oracle

```
root@ecc-laptop:~# mkdir /home/oracle
root@ecc-laptop:~# chown -R oracle:dba /home/oracle/
```

Figura 46 CREACIÓN DE DIRECTORIOS

1.8. Creamos los siguientes enlaces simbólicos:

- ln -s /usr/bin/awk /bin/awk
- ln -s /usr/bin/rpm /bin/rpm
- ln -s /usr/bin/basename /bin/basename
- mkdir /etc/rc.d
- for i in 0 1 2 3 4 5 6 S ; do ln -s /etc/rc$i.d /etc/rc.d/rc$i.d ; done

```
root@ecc-laptop:~# ln -s /usr/bin/awk
root@ecc-laptop:~# ln -s /usr/bin/awk /bin/awk
root@ecc-laptop:~# ln -s /usr/bin/rpm /bin/rpm
root@ecc-laptop:~# ln -s /usr/bin/basename /bin/pandora
root@ecc-laptop:~# mkdir /etc/r
rc0.d/        rc3.d/        rc6.d/        resolvconf/   rpc
rc1.d/        rc4.d/        rc.local      resolv.conf   rsyslog.conf
rc2.d/        rc5.d/        rcS.d/        rmt           rsyslog.d/
root@ecc-laptop:~# mkdir /etc/rc.d
root@ecc-laptop:~# for i in 0 1 2 3 4 5 6 S ; do ln -s /etc/rc$i.d /etc/rc.d/rc$
i.d ; done
```

Figura 47 CREACIÓN DE ENLACES SIMBÓLICOS

1.9. Creamos la carpeta en donde se instalará Oracle y asignamos como propietario al usuario Oracle:

- mkdir -p /u01/app/oracle
- chown -R oracle:dba /u01

1.10. Ahora se carga las carpetas en donde están los instaladores de la base de datos:

- mv /home/usuario/Escritorio/database/ /media/
- chown -R oracle:oinstall /media/database

1.11. Realizar los siguiente cambios en los archivos de configuración:

- cd /etc
- cp sysctl.conf sysctl.conf.original
- vi sysctl.conf
- Agregar estas líneas al final del archivo:
  - net.core.rmem_default = 4194304
  - net.core.rmem_max = 4194304
  - net.core.wmem_default = 262144
  - net.core.wmem_max = 262144

Par terminar la edición digitamos esc y colocamos en la línea de comandos: wq!

```
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
fs.file-max = 65535
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 1024 65535
net.core.rmem_default = 4194304
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 262144

root@ecc-laptop:/etc#
```

Figura 48 CONFIGURACIÓN DE ARCHIVOS DEL SISTEMA

1.12. Verificar los cambios en el archivo:

- cat sysctl.conf

1.13. Nos ubicamos en el archivo limits.conf:

- cd /etc/security/

- cp limits.conf limits.conf.original
- vi limits.conf
- Agregar las siguientes líneas al final del archivo:
  - oracle soft nproc 2047
  - oracle hard nproc 16383
  - oracle soft nofile 1023
  - oracle hard nofile 65535

1.14. Verificar los cambios en el archivo:

- cat limits.conf

1.15. Ubicarse en el siguiente directorio y editar el archivo login:

- cd /etc/pam.d
- cp login login.original
- vi login
- Agregar las siguientes líneas al final del archivo:
  - session required /lib/security/pam_limits.so
  - session required pam_limits.so

1.16. Verificar los cambios en el archivo:

- cat login

1.17. Configurar el Display primero como root:

- sudo -s
- DISPLAY=:0; export DISPLAY; xhost +;

1.18. Autenticarse como Oracle para configurar el display

- su - oracle
- DISPLAY=:0; export DISPLAY

```
ecc@ecc-laptop:~$ sudo su -
[sudo] password for ecc:
root@ecc-laptop:~#
root@ecc-laptop:~#
root@ecc-laptop:~# DISPLAY=:0; export DISPLAY; xhost +;
access control disabled, clients can connect from any host
root@ecc-laptop:~# su - oracle
oracle@ecc-laptop:~$ DISPLAY=:0; export DISPLAY
oracle@ecc-laptop:~$ xhost
access control disabled, clients can connect from any host
```

Figura 49 CONFIGURACIÓN DEL DISPLAY

1.19. Verificar que el display está configurado correctamente, (si muestra el reloj está configurado correctamente).

xclock

2. Instalación de la Base de Datos

2.1. Ubicarse en la dirección en donde están los instaladores de la base de datos:

- cd /media/database/
- ./runInstaller -ignoreSysPrereqs

2.2. Seleccionar crear y configurar base de datos.

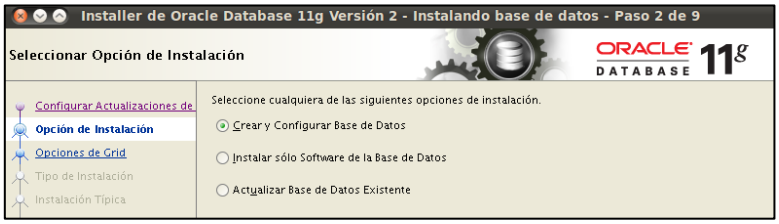


Figura 50 INSTALLER ORACLE

2.3. Se observa la configuración básica para la instalación de la base de datos en donde se muestra el directorio donde se ubica la base de datos, la ubicación de los archivos de la misma, el juego de caracteres y su nombre que es *orcl* y finalmente se añade la *contraseña* del administrador para esta se debe considerar los estándares para una clave segura debe incluir un carácter en Mayúsculas y al menos un número.

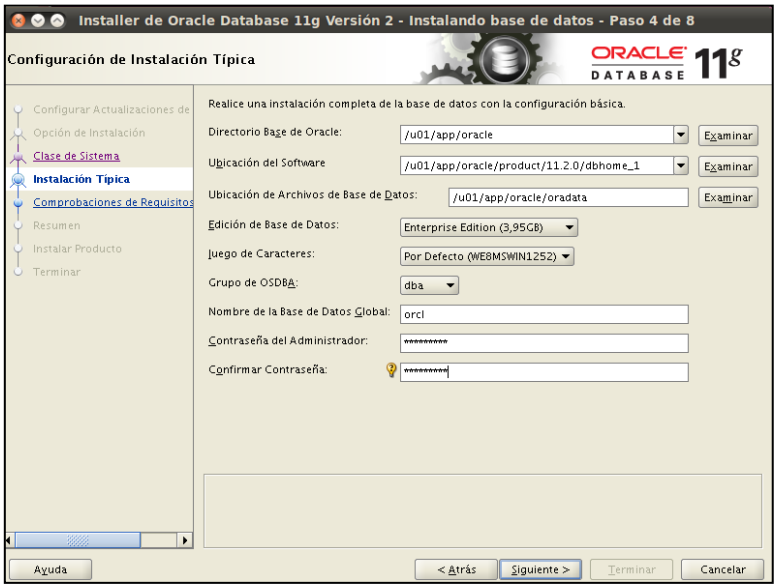


Figura 51 CONFIGURACIÓN DE VARIABLES

2.4. En este paso se realiza la comprobación de los requisitos para la instalación, nos lanzara un error para eliminar este error se debe ejecutar el archivo *runfixup.sh* para esto nos autenticamos como *root*:

- cd /tmp/CVU_11.2.0.1.0_oracle/
- ./runfixup.sh

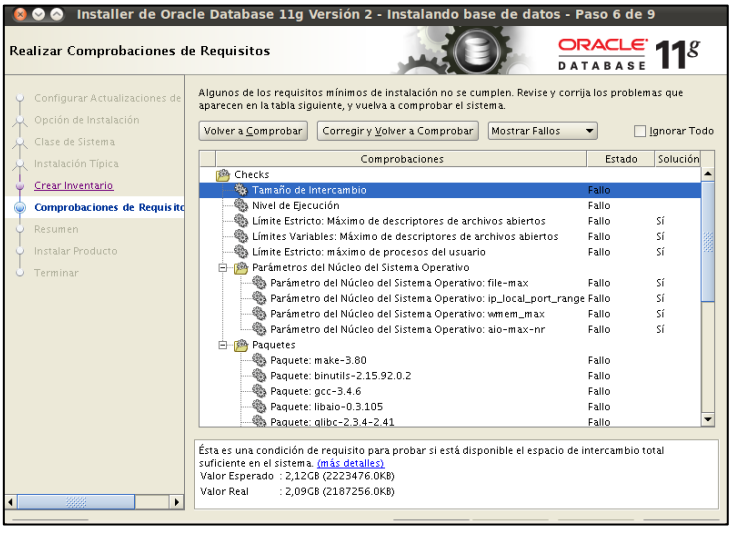


Figura 52 COMPROBACIÓN DE REQUISITOS

```
root@ecc-laptop:~# /tmp/CVU_11.2.0.1.0_oracle/runfixup.sh
Response file being used is :/tmp/CVU_11.2.0.1.0_oracle/fixup.response
Enable file being used is :/tmp/CVU_11.2.0.1.0_oracle/fixup.enable
Log file location: /tmp/CVU_11.2.0.1.0_oracle/orarun.log
Setting Kernel Parameters...
fs.file-max = 65535
fs.file-max = 6815744
The upper limit of ip_local_port range in reponse file is not greater than value
 in /etc/sysctl.conf, hence not changing it.
The upper limit of ip_local_port range in response file is not greater than valu
e for current session, hence not changing it.
net.ipv4.ip_local_port_range = 9000 65535
net.core.wmem_max = 262144
net.core.wmem_max = 1048576
fs.aio-max-nr = 1048576
uid=1001(oracle) gid=1001(oinstall) grupos=1001(oinstall),1002(dba)
oracle hard nproc 16383
oracle hard nofile 65535
oracle soft nofile 1023
root@ecc-laptop:~#
```

Figura 53 SOLUCIONAR PROBLEMAS DE REQUISITOS

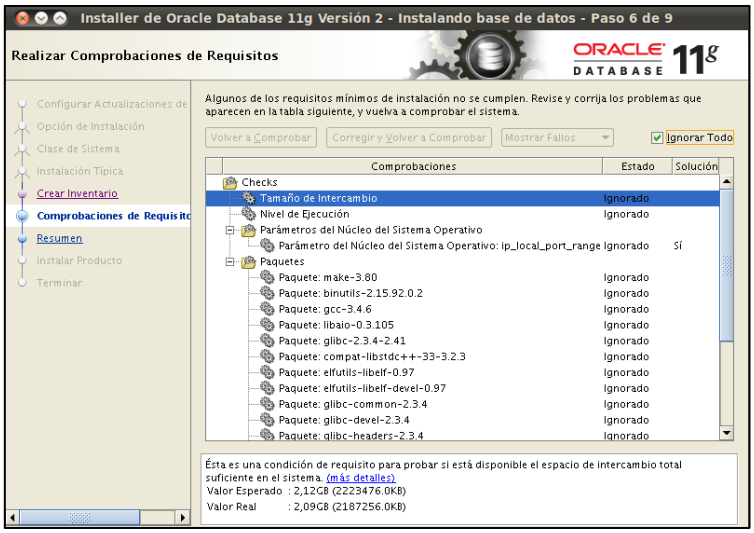


Figura 54 SOLUCIONAR REQUISITOS (SOLUCIONADO)

2.5. La instalación inicia y es posible que salga un error, que nos dice error al llamar al destino 'idg40dbc' del archivo make, yo corregí este error comentando la línea (HSODBS_LINKLINE) en el directorio /u01/app/Oracle/product/11.2.0/dbhome_1/rdbms/lib/ins_rdbms.mk:

```
$(HSDEPXA) : $(ALWAYS) $(HSDEPXA_DEPS)
        $(SILENT)$(ECHO)
        $(SILENT)$(ECHO) " - Linking HS Distributed External Procedure agent"
        $(RMF) $@
        $(HSDEPXA_LINKLINE)

$(HSOTS) : $(ALWAYS) $(HSOTS_DEPS)
        $(SILENT)$(ECHO)
        $(SILENT)$(ECHO) " - Linking HS OTS agent"
        $(RMF) $@
        $(HSOTS_LINKLINE)

$(HSODBC) : $(ALWAYS) $(HSODBC_DEPS)
        $(SILENT)$(ECHO)
        $(SILENT)$(ECHO) " - Linking $(HSDG4)odbc agent"
        $(RMF) $@
        #$(HSODBC_LINKLINE)

dg4odbc32 hsodbc32:
        $(MAKE) -f $(LINKMAKEFILE) $(HSODBC) $(REDEFINES32)

$(HSOLEFS) : $(ALWAYS) $(HSOLEFS_DEPS)
```

Figura 55 SOLUCIONAR PROBLEMA DE RDBMS

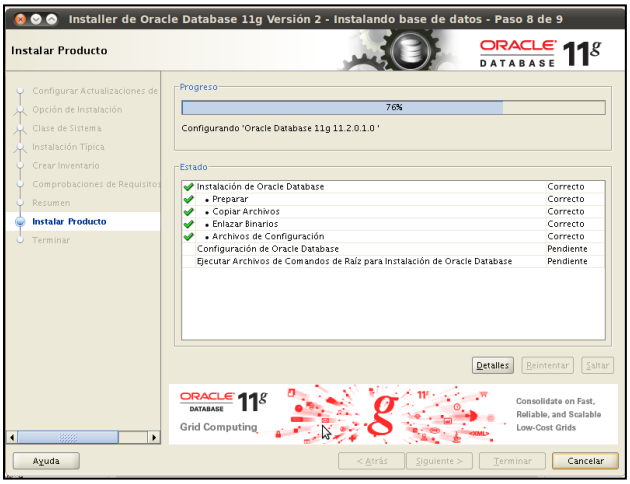


Figura 56 SOLUCIONAR PROBLEMAS DE RDBMS (SOLUCIONADO)

2.6. Ahora se realizan los siguientes pasos, para levantar el listener manualmente en caso que el instalador no lo haga automáticamente:

- ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1/
- PATH=$PATH:$ORACLE_HOME/bin
- export ORACLE_SID=orcl
- export ORACLE_HOME
- export PATH
- cd /u01/app/oracle/product/11.2.0/dbhome_1/bin/
- ./netca

Figura 57 CONFIGURACIÓN DEL LISTENER

2.7. Agregar el listener:

2.8. Escribir el nombre del listener.

2.9. Dejar el protocolo *TCP* por defecto

2.10. Ejecutar el siguiente comando para levantar el *asistente de configuración de la base de datos*:

- ./dbca

2.11. Seleccionar crear base de datos:

2.12. Seleccionar una Plantilla.

2.13. Escribir el nombre de la base de datos y su SID.

2.14. Configurar Enterprise Manager:

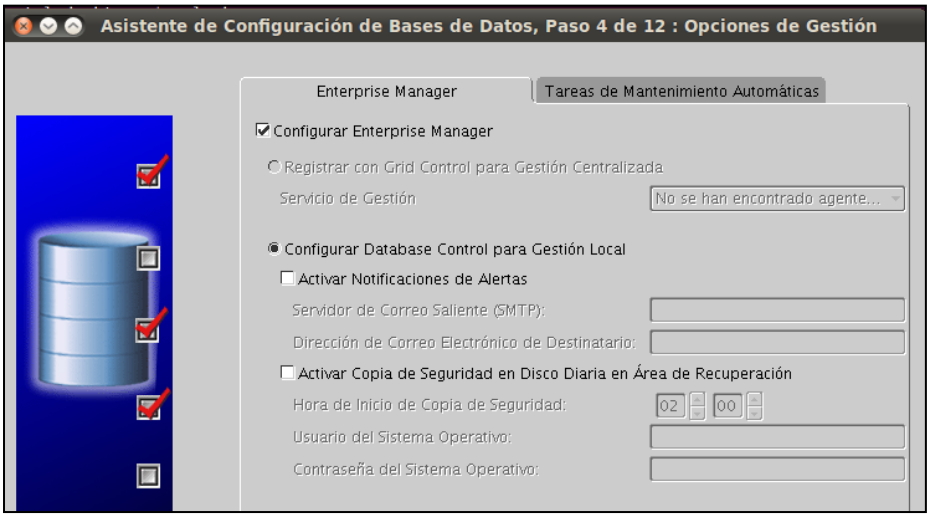


Figura 58 CONFIGURACIÓN DEL ENTERPRISE MANAGER

2.15. Configurar las credenciales para la base de datos, se puede marcar usar la misma contraseña para todas las cuentas.

2.16. Dejar las configuraciones por defecto.

2.17. Si se desea se instala los esquemas de ejemplos.

2.18. Configurar los parámetros de inicialización. Parámetros como el tamaño de memoria para el SGA(System Global Area) y el PGA(Program Global Area), se selecciona *Típica* que es la recomendada por el asistente:

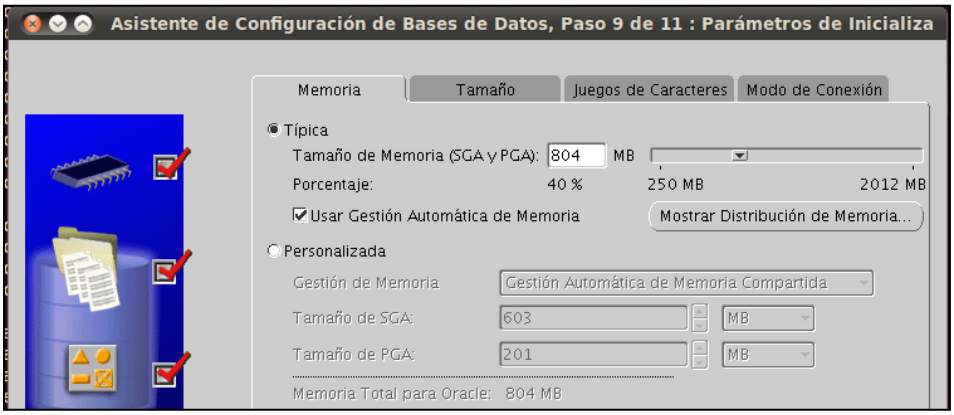


Figura 59 CONFIGURACIÓN DE LOS PARÁMETROS DE INICIALIZACIÓN

2.19. Inicia la creación de la base de datos:

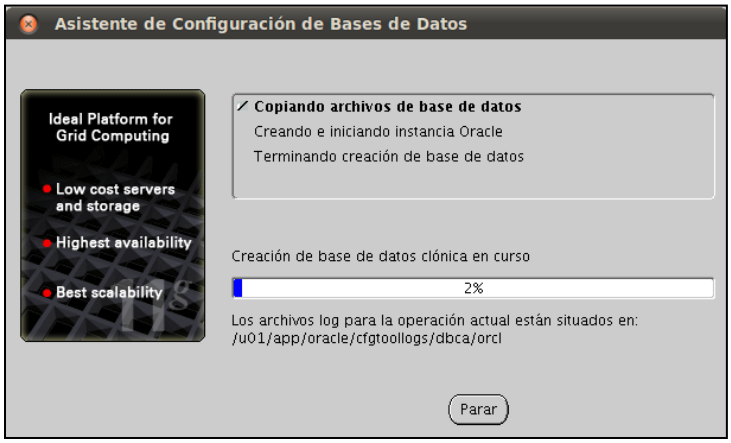


Figura 60 INICIALIZACIÓN DE LA CREACIÓN DE LA BASE DE DATOS

2.20. La creación de la base de datos a finalizado con éxito y para desbloquear la cuenta MDSYS hay que ubicarse en gestión de contraseñas y proceder activarla:

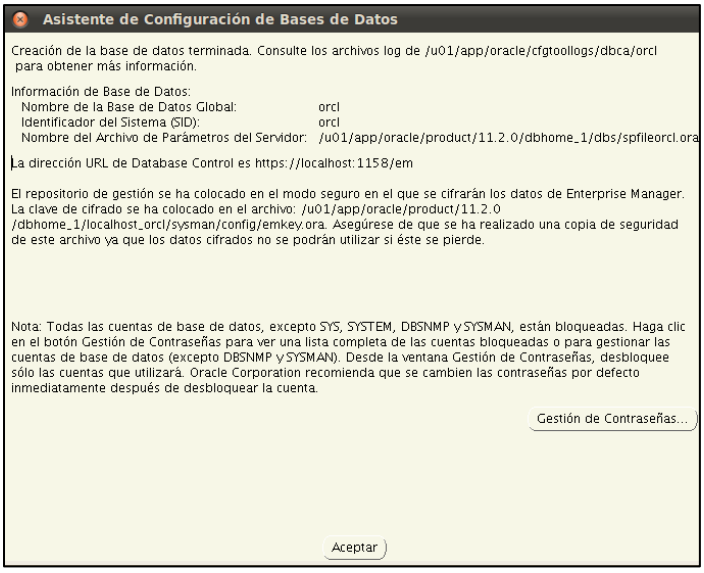


Figura 61 FINALIZACIÓN DEL ASISTENTE DE CONFIGURACIÓN DE BASE DE DATOS

2.21. Configurar permisos de OraInventory, con este paso se ha culminado la instalación.

```
root@ecc-laptop:/u01/app/oracle/product/11.2.0/dbhome_1/rdbms/lib# sudo /u01/app
/oraInventory/orainstRoot.sh
Cambiando permisos de /u01/app/oraInventory.
Agregando permisos de lectura y escritura para el grupo.
Eliminando permisos de lectura, escritura y ejecución para el público.

Cambiando el nombre de grupo de /u01/app/oraInventory a oinstall.
La ejecución del archivo de comandos ha terminado
root@ecc-laptop:/u01/app/oracle/product/11.2.0/dbhome_1/rdbms/lib# sudo /u01/app
/oracle/product/11.2.0/dbhome_1/root.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.2.0/dbhome_1

Enter the full pathname of the local bin directory: [/usr/local/bin]:
```

Figura 62 ASIGNACIÓN DE PERMISOS

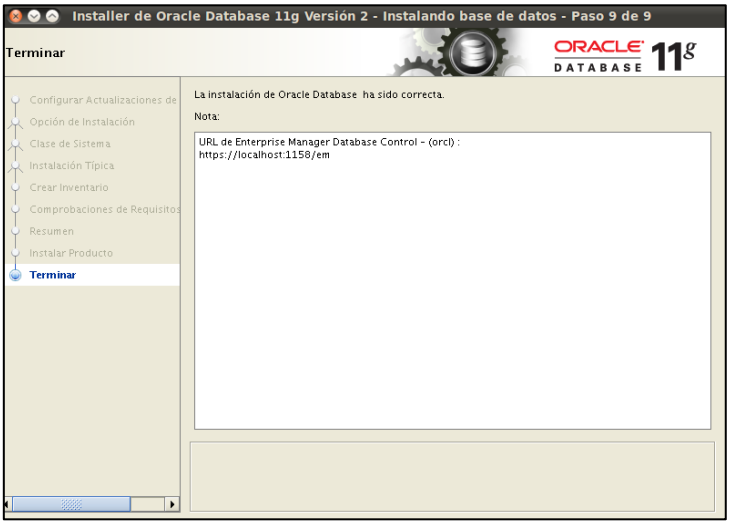


Figura 63 FINALIZACIÓN DE LA INSTALACIÓN

ANEXO 2: LEVANTAR ORACLE SPATIAL 11g EN UBUNTU 11.4

1. Logearse como usuario Oracle y establecer directorios de la siguiente manera:
    - su – oracle
    - ORACLE_HOME=/u01/app/oracle/product/11.2..0/dbhome_1/
    - PATH=$PATH:$ORACLE_HOME/bin
    - export ORACLE_SID = sid de la base de datos
    - export ORACLE_BASE
    - export PATH
    - cd $ORACLE_HOME/bin

```
root@ecc-desktop:~# su - oracle
oracle@ecc-desktop:~$ ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1/
oracle@ecc-desktop:~$ PATH=$PATH:$ORACLE_HOME/bin
oracle@ecc-desktop:~$ export ORACLE_SID=pandora
oracle@ecc-desktop:~$ export ORACLE_HOME
oracle@ecc-desktop:~$ export PATH
oracle@ecc-desktop:~$ cd $ORACLE_HOME/bin
oracle@ecc-desktop:/u01/app/oracle/product/11.2.0/dbhome_1/bin$
```

Figura 64 CONFIGURACIÓN DE DIRECTORIOS

2. Levantar listener
    - ./lsnrctl start

```
oracle@ecc-desktop:/u01/app/oracle/product/11.2.0/dbhome_1/bin$ ./lsnrctl start

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 08-MAY-2010 13:35:49

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/11.2.0/dbhome_1//bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 11.2.0.1.0 - Production
System parameter file is /u01/app/oracle/product/11.2.0/dbhome_1/network/admin/l
istener.ora
Log messages written to /u01/app/oracle/diag/tnslsnr/ecc-desktop/listener/alert/
log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date                08-MAY-2010 13:35:51
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/oracle/product/11.2.0/dbhome_1/network/admin/
listener.ora
Listener Log File         /u01/app/oracle/diag/tnslsnr/ecc-desktop/listener/aler
t/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1521)))
The listener supports no services
The command completed successfully
oracle@ecc-desktop:/u01/app/oracle/product/11.2.0/dbhome_1/bin$
```

Figura 65 START LISTENER

3. Levantar SQLPLUS y nos conectamos como usuario sysdba.
    - ./sqlplus /nolog
    - conn /as sysdba
    - Levantar una instancia de la base de datos
        - startup;

```
oracle@ecc-desktop:/u01/app/oracle/product/11.2.0/dbhome_1/bin$ ./sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Sat May 8 13:36:38 2010

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

SQL> conn /as sysdba
Connected to an idle instance.
```

Figura 66 SQLPLUS

4. Comprobar que todos los archivos de la base de datos estén correctamente, ejecutando la siguiente consulta.
    - Show sga;

```
SQL> show sga;

Total System Global Area  924860416 bytes
Fixed Size                  1340328 bytes
Variable Size             574622808 bytes
Database Buffers          343932928 bytes
Redo Buffers                4964352 bytes
SQL>
```

Figura 67 SGA

ANEXO 3: CONFIGURACIÓN DEL ENTORNO SEMÁNTICO

Por defecto el entorno que permite trabajar con RDF, RDFS u OWL no está configurado, para ello se debe ejecutar los siguientes procedimientos.

1. Se debe ejecutar los siguientes scripts para instalar el componente semántico del RDF, se incluye la arroba al inicio de la sentencia para iniciar el script:
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/catsem10i.sql
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/catsem11i.sql

```
SQL> conn /as sysdba
Connected.
SQL> @/u01/app/oracle/product/11.2.0/dbhome 1/md/admin/catsem10i.sql
```

Figura 68 USUARIO SYSDBA

```
Synonym created.

SQL> describe sem_upgrade_to_11;
PROCEDURE CREATE_ALL_MODELS_IN_11
PROCEDURE EMPTY_APP_TABS_DROP_RDF_COLS
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 DOWNGRADE                      BOOLEAN                 IN     DEFAULT
PROCEDURE LOAD_FROM_ALL_STAGING_TABLES
PROCEDURE RESTORE_10_2_RDF_NETWORK_IN_11
PROCEDURE SAVE_10_2_RDF_NETWORK_FOR_11
PROCEDURE SETUP_FOR_LOADING_IN_11
PROCEDURE UNLOAD_ALL_INTO_STAGING_TABLES
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 DOWNGRADE                      BOOLEAN                 IN     DEFAULT

SQL> set echo off
restore 10.2 RDF data in 11 ...

PL/SQL procedure successfully completed.

restore 10.2 RDF data in 11 ... DONE
```

Figura 69 EJECUCIÓN DE SCRIPTS

2. Conectarse como usuario mdsys y otorgarle permisos de DBA.

```
SQL> alter user mdsys account unlock;

User altered.

SQL> alter user mdsys identified by pandora;

User altered.

SQL> GRANT CREATE TABLE, CREATE SESSION, DBA TO MDSYS;

Grant succeeded.

SQL> GRANT SELECT ANY TABLE TO MDSYS;

Grant succeeded.
```

Figura 70 USUARIO MDSYS

3. Ejecutar los siguientes archivos para habilitar el entorno semántico de datos
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/sdordfh.sql
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/sdordfxh.sql
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/sdordfa.sql
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/sdordfb.plb
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/sdordfxb.plb
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/sdoseminfhb.plb
    - @/u01/app/oracle/product/11.2.0/dbhome_1/md/admin/sdoseminfhb.plb

Figura 71 HABILITAR ENTORNO SEMÁNTICO

Si los scripts se ejecutaron con éxito El entorno semántico se ha configurado correctamente.

4. Para la creación de modelos debemos crear un Tablespace y un Temporary Tablespace para los usuarios RDF. Creamos el Tablespace con la siguiente instrucción:

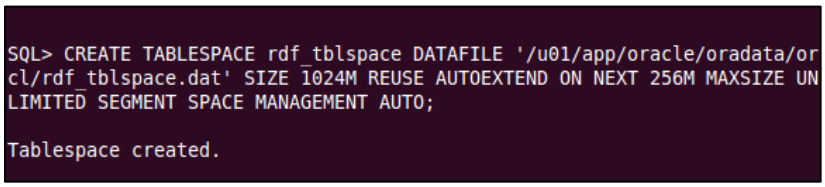


Figura 72 CREACIÓN DE TABLESPACE

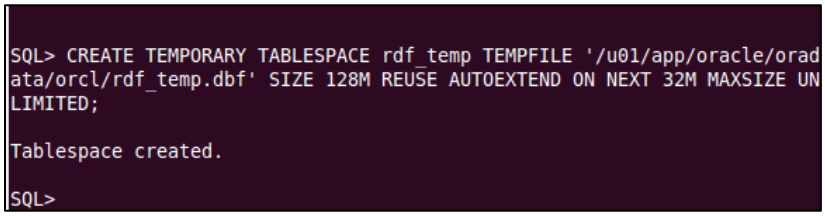


Figura 73 CREACIÓN DE TEMPORARY TABLESPACE

5. Asignar el Tablespace a la red semántica que se va a crear:

```
SQL> EXECUTE sem_apis.create_sem_network('rdf_tblspace');

PL/SQL procedure successfully completed.

SQL>
```

Figura 74 ASIGNACIÓN DEL TABLESPACE A UNA RED SEMÁNTICA

6. Crear un usuario, se define el nombre del usuario, un password y se asigna el Tablespace y el Tablespace temporal que se creó en el paso anterior y Asignar los permisos correspondientes para el usuario

```
SQL>
SQL> CREATE USER leo IDENTIFIED BY pandora DEFAULT TABLESPACE rdf_tblspa
ce TEMPORARY TABLESPACE rdf_temp;
```

Figura 75 CREACIÓN DE UN USUARIO

7. Conectarse con el usuario creado y crear la tabla para cargar las tripletas y crea el modelo en donde se van a cargar los datos e insertamos datos:

```
SQL> connect leo/pandora
Connected.
SQL> CREATE TABLE FAMILY_RDF_DATA(ID NUMBER, TRIPLE SDO_RDF_TRIPLE_S);

Table created.

SQL> EXECUTE SEM_APIS.CREATE_RDF_MODEL('FAMILY', 'FAMILY_RDF_DATA', 'TRI
PLE');

PL/SQL procedure successfully completed.

SQL> INSERT INTO family_rdf_data VALUES (1,
  2  SDO_RDF_TRIPLE_S('family',
  3  'http://www.example.org/family/John',
  4  'http://www.example.org/family/fatherOf',
  5  'http://www.example.org/family/Suzie'));

1 row created.
```

Figura 76 ALMACENAMIENTO DE DATOS RDF.

ANEXO 4: INSTALACIÓN DE ORACLE 10G EN MAC OS X SERVER 10.5.4

Requerimientos:

Oracle 10g	Está disponible en: http://www.oracle.com/technology/software/products/database/index.html
Sistema Operativo Mac OS X Server 10.5.4	Oracle oficialmente no es soportada en otra versión de Mac, pero se hará referencia como a instalar en Mac OS X Leopard Snow.
Capacidad	Al menos 1GB de RAM y 1.2 GB de Disco,
Swap	La Cantidad de espacio swap debe ser como mínimo, la misma que la de la RAM física.
Xcode tools 3.0	O superior, esta aplicación viene el CD de instalación de Mac OS X Server en Utilidades.
JDK de java	Para saber si está instalado y que versión ingrese el comando java –versión, si no está instalado no devolverá nada en la pantalla para descargarlo puede ir a Software Update de su Mac y Actualizar automáticamente se descargara e instalara.

Tabla XIII REQUERIMIENTOS DE ORACLE PARA MAC OS X

Pasos:

1. Abrir un Terminal (/Applications/Utilities/Terminal) Logearse como usuario root
   - sudo sh e ingresa el password.
2. Para saber la cantidad física de RAM ingresar el siguiente comando en un terminal.
   - /usr/sbin/system_profiler SPHardwareDataType | grep Memory.
3. Para determinar la cantidad de espacio swap ingresar el comando:
   - df –h
4. Para determinar si la arquitectura del sistema puede ejecutar el software introduzca el comando:

- uname –p Esto debe retornar un valor i386

5. Para determinar la versión del sistema operativo ingresar el siguiente comando:
   - sw_vers
6. Vamos a crear usuarios y grupos Oracle, por ejemplo para crear grupo oinstall y el usuario Oracle he usado el número de usuario y grupo 1025 para asegurarse de que no choquen con los grupos existentes y los usuarios, se puede usar Workgroup Manager (/Applications/Server/Workgroup Manager) para crear usuarios y grupos en mi caso lo hago directamente por comandos.

   - dscl . -create /groups/oinstall dscl . -append /groups/oinstall gid 1025
   - dscl . -append /groups/oinstall passwd "*"
   - dscl . -create /users/oracle dscl . -append /users/oracle uid 1025
   - dscl . -append /users/oracle gid 600
   - dscl . -append /users/oracle shell /bin/bash
   - dscl . -append /users/oracle home /Users/oracle
   - dscl . -append /users/oracle realname "Oracle software owner"
   - mkdir /Users/oracle
   - chown oracle:oinstall /Users/oracle

7. Luego de crear los usuarios se creael directorio /Users/oracle, y se otorga permisos al grupo oinstall. Adicionalmente se puede grupos como dba, oper estos son opcionales

```
sh-3.2# dscl . -create /groups/oinstall
sh-3.2# dscl . -append /groups/oinstall gid 1025
sh-3.2# dscl . -append /groups/oinstall passwd "*"
sh-3.2# dscl . -create /groups/dba
sh-3.2# dscl . -append /groups/dba gid 1026
sh-3.2# dscl . -append /groups/dba passwd "*"
sh-3.2# dscl . -create /groups/oper
sh-3.2# dscl . -append /groups/oper gid 1027
sh-3.2# dscl . -append /groups/oper passwd "*"
sh-3.2# id oracle
id: oracle: no such user
sh-3.2# dscl . -create /users/oracle
sh-3.2# dscl . -append /users/oracle uid 1025
sh-3.2# dscl . -append /users/oracle gid 1025
sh-3.2# dscl . -append /users/oracle shell /bin/bash
sh-3.2# dscl . -append /users/oracle home /Users/oracle
sh-3.2# dscl . -append /users/oracle realname "Oracle software owner"
sh-3.2# mkdir /Users/oracle
sh-3.2# chown oracle:oinstall /Users/oracle
sh-3.2# passwd oracle
Changing password for oracle.
New password:
Retype new password:
sh-3.2#
```

Figura 77 CREACIÓN DE USUARIO PARA ORACLE

Si usamos el Workgroup Manager debemos crear el grupo oinstall.

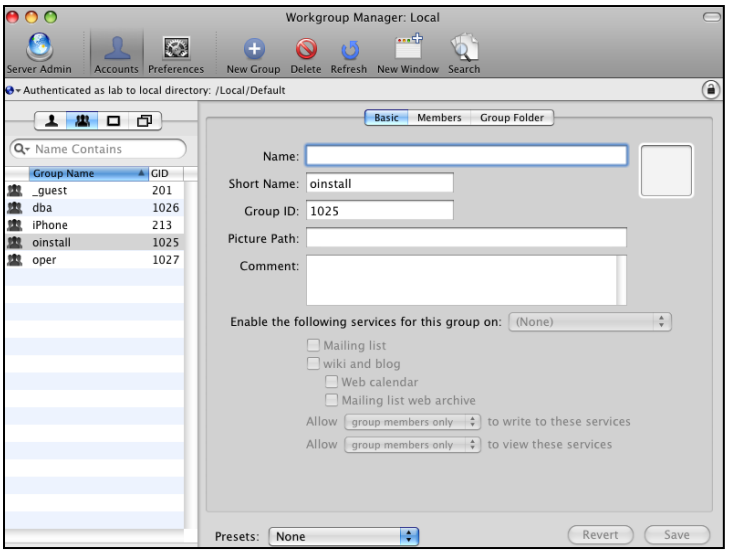


Figura 78 CREACIÓN DE USUARIO PARA ORACLE CON WORKGROUP MANAGER

Cuando creamos el usuario Oracle debemos crearlo con las siguientes características.

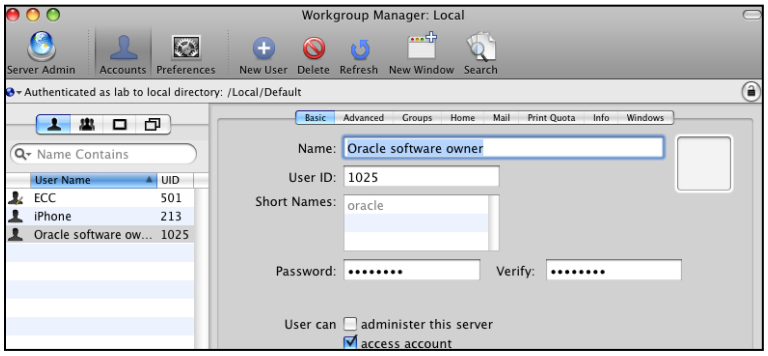


Figura 79 CREACIÓN DE USUARIO ORACLE OWNER CON WORKGROUP MANAGER

Debemos asegurarnos que primary Group id es el del oinstall

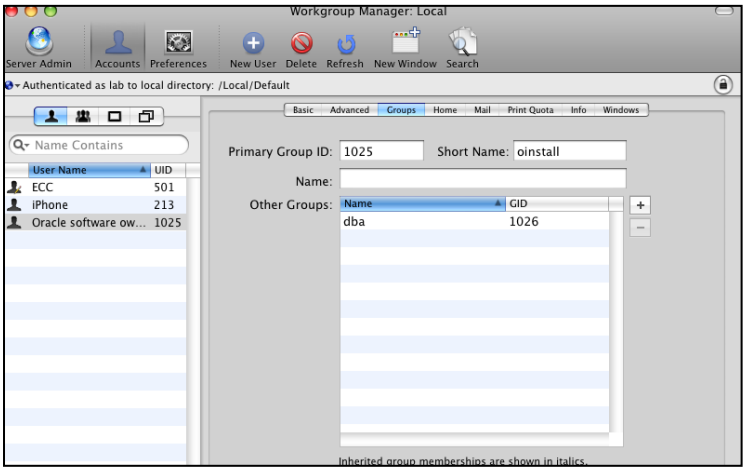


Figura 80 GRUPO OINSTALL

8. Debemos cambiar los parámetros del Kernel por defecto, para la producción sistemas de base de datos, Oracle recomienda que estos valores para optimizar el rendimiento del sistema, para cambiar los estos valores por defecto ingresar el comando.

   - vi /etc/sysctl.conf

   Luego de hacer los cambios presione ESC luego :wq! Para guardar y cerrar el archivo

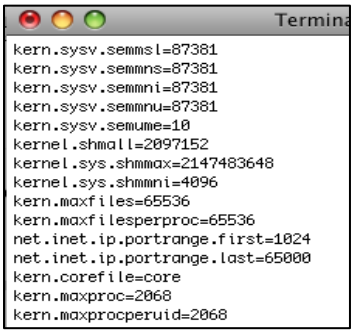


Figura 81 CAMBIO DE PARÁMETROS DEL KERNEL

9. Para esta versión de Mac no es necesario hacer el siguiente paso para la versión Snow Leopard el script Oracle DB de instalación tienen referencia a la versión 1.4.2 de Java que no está presente en Snow Leopard. La forma más fácil de solucionarlo es crear un enlace simbólico a la versión más reciente de Java:

    - sudo ln -s /System/Library/Frameworks/JavaVM.framework/Versions/1.5.0 /System/Library/Frameworks/JavaVM.framework/Version/1.4.2

10. Se debe rreiniciar el equipo para que los cambios de los parámetros del Kernel tengan efecto.
11. Logearse como usuario oracle ingresando el comando:
    - su – oracle
12. Ingresar el comando **pwd** para saber dónde está ubicado es necesario que este bajo el directorio **/Users/Oracle** si no está hay que cambiar de directorio con el comando:
    - cd /Users/oracle
13. Abrimos o creamos el archivo bash_profile para ello ingresamos el comando:
    - vi .bash_profile

    Esto en el caso de que estemos bajo el shell para el usuario Oracle el cual por defecto en Mac es bash. Para saber el shell con el que estamos trabajando ingresamos el comando:
    - echo $SHELL

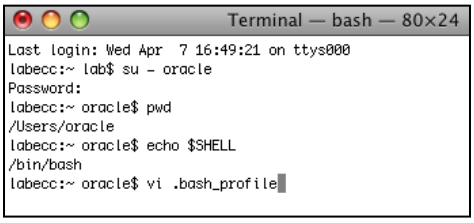


Figura 82 SHELL MAC OS X

14. Añadimos las siguientes líneas al archivo bash_profile
    - export DISPLAY=:0.0
    - export ORACLE_BASE=$HOME
    - umask 022
    - ulimit -Hn 65536
    - ulimit -Sn 65536

    En el display podemos poner nuestra dirección ip o en su defecto localhost, luego establecimos el ORACLE_BASE y el comando umask 022 nos sirve para que se asignen otros permisos distintos a los que vienen por defecto en la distribución. Cuando terminemos de añadir las líneas al archivo debemos presionar la tecla ESC y dos puntos más wq! para poder guardar y cerrar.
15. Ejecutamos el archivo .bash_profile para ello ingresamos el comando
    - . ./.bash_profile

16. Ahora tenemos que crear los directorios para la instalación y debemos otorgarle los permisos necesarios, para ello abrimos una nueva ventana de terminal y nos logearnos como usuario root y nos aseguramos estar bajo el directorio /Users/oracle, e ingresamos los comandos:
    - sudo sh
    - pwd
    - cd /Users/oracle
    - mkdir /oracle
    - mkdir /oraInventory
    - chown oracle:oinstall /oraInventory/
    - chown oracle:oinstall /oracle/
    - chmod 775 /oracle

- chmod 775 /oraInventory

El commando Mkdir nos permite crear el directorio Oracle en el cual se instalara el software y el directorio oraInventory, el comando chown cambia los permisos de usuario:grupo de ficheros/directorios, luego cambiamos los permisos a los directorios.

17. Ahora vamos a crear un directorio para el software donde lo vamos a descargar o copiar en caso que ya lo tengamos, para ello vamos a ingresar los siguientes comandos:
    - mkdir /OraDb10g
      Nos cambiamos a este directorio
    - cd /OraDb10g
      Ahora copiamos el archivo a este directorio
    - unzip db.zip

    Nos cambiamos al siguiente directorio bajo el usuario oracle.
    - su – oracle
    - cd /OraDb10g/db/Disk1
18. Ahora empezaremos la instalación para ello ejecutaremos el archivo runInstaller bajo el usuario oracle para ello ingresamos el comando
    - ./runInstaller

    Esto iniciará una ventana gráfica con el wizard de instalación.

    Nota: En el caso de estar instalando bajo Mac OS X Snow debemos ingresar el comando:
    - ./runInstaller -J-d32

    Esto es para forzar a correr Java en modo 32 bits

```
labecc:~ oracle$ cd /OraDb10g/db/Disk1/
labecc:Disk1 oracle$ ./runInstaller
Starting Oracle Universal Installer...
```

Figura 83 ORACLE INSTALLER

19. Seleccionamos Instalación Avanzada, ya que esta nos permite establecer diferentes passwords para las cuentas SYS, SYSTEM, SYSMAN y DBSNMP, de igual manera nos permite elegir el character set, entre otras.

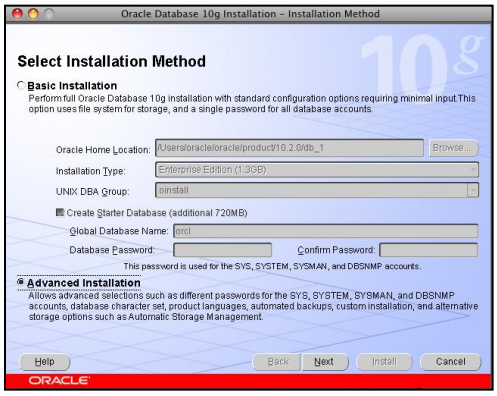


Figura 84 SELECCIÓN DE MÉTODO DE INSTALACIÓN

20. podemos seleccionar el tipo de instalación, elegimos la edición Enterprise la cual es la más alta del rango en las versiones de Bases de Datos oracle.
21. Al establecer los directorios nos aparecerán automáticamente, el Path es la ruta donde se instalara la base de datos, por defecto es /oracle/oracle/product/10.2.0/db_1
22. Seleccionamos las opciones de configuración y elegiremos create database.
23. Elegimos el tipo de base de datos que deseamos instalar, elegiremos General Purpose
24. Especificamos la configuración de la base de datos, como el nombre de la base de datos, el character set, el nombre por defecto es orcl, pero se puede especificar cualquier nombre, si desea que se cree la base de datos con esquemas de ejemplo seleccione el check box.

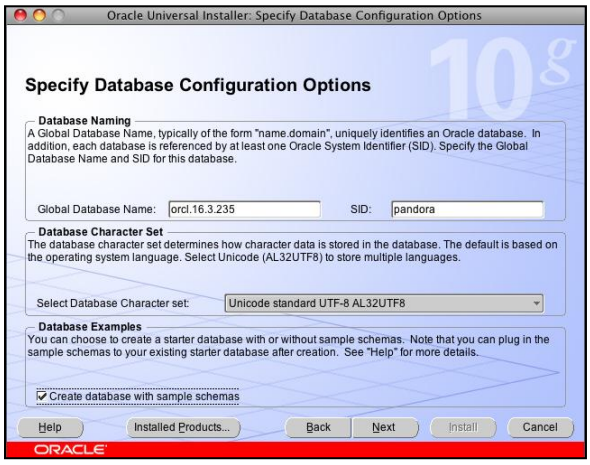


Figura 85 CONFIGURACIÓN DE PARÁMETROS DE LA BASE DE DATOS

25. Seleccionar archivos de sistemas y especificamos el directorio por defecto oradata, Oracle recomienda que para una mejor organización y funcionamiento de la base de datos instalar los data files y el software Oracle en diferentes directorios.
26. Especificamos los passwords, elegiremos usar el mismo password para todas las cuentas.
27. Nos presentara un resumen con los paramentos que hemos establecido anteriormente y Comienza la instalación este proceso tomara varios minutos.

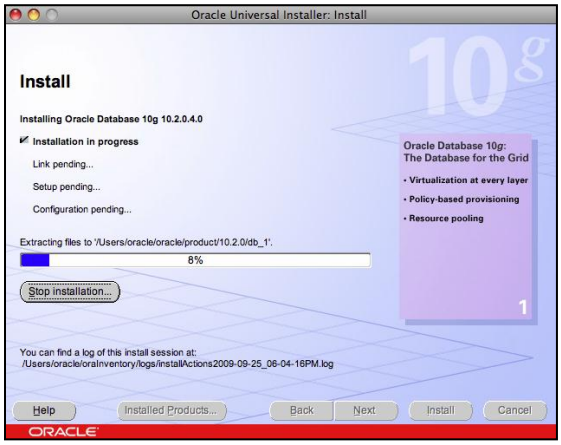


Figura 86 ORACLE UNIVERSAL INSTALLER

Nota: Si se está instalando en Leopard Snow en medio de la instalación aparecerá el siguiente

- "Error in invoking target 'all_no_orcl ipc_g ihsodbc32′ …"

No presionar nada y abrir un terminal en el cual vamos a ingresar en comando

- cd Users/oracle/oracle/product/10.2.0/db_1/rdbms/lib vi ins_rdbms.mk

Se abrirá un archivo y aquí buscamos la línea que contenga HSODBC_LINKLINE y comentamos esta línea

- # $(HSODBC_LINKLINE)

Presionamos ESC y :wq! para guardar y cerrar el archivo, y presionamos Retry en el Oracle Universal Installer.

De este modo desactivamos la compilación de librerías, que no es necesario para instalación de Oracle.

Nota: En la versión Mac OS X Server 10.5.4 no tenemos el error mencionado anteriormente.

28. El Oracle Universal Installer nos pedirá que ejecutemos los siguientes scripts.

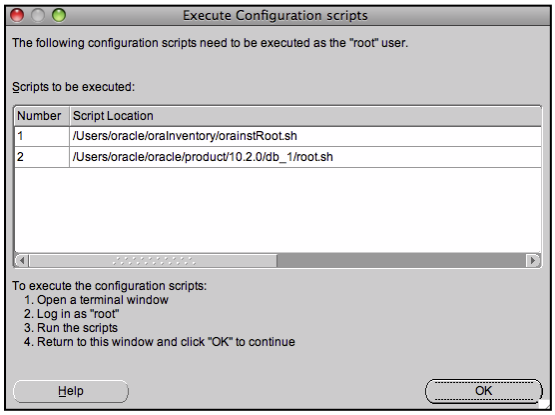


Figura 87 EXECUTE CONFIGURACIÓN SCRIPTS

29. Abrimos un nuevo terminal y nos logeamos como usuario root e ingresamos los siguientes comandos:
    - sudo /Users/oracle/oraInventory/orainstRoot.sh
    - sudo /Users/oracle/product/10.2.0/db_1/root.sh

    Aceptamos los valores por defecto del pathname del directorio local bin [/usr/local/bin]: dando enter.
    Presionamos ok en la ventana de ejecución de scripts de configuración.
30. Nos mostrara una ventana de OUI informándonos que la instalación ha finalizado.
31. El software se ha instalado pero no se ha configurado la base de datos ni el listener se ha creado, para ello debemos cambiar de directorio a /Users/oracle/ bajo el usuario Oracle Modificamos el bash_profile., con el comando vi .bash_profile lo vamos modificar y agregamos entradas como muestra la Figura para establecer el Oracle_home, para guardar y salir presionar la tecla ESC y :wq! estos comandos sirven para guardar y salir.

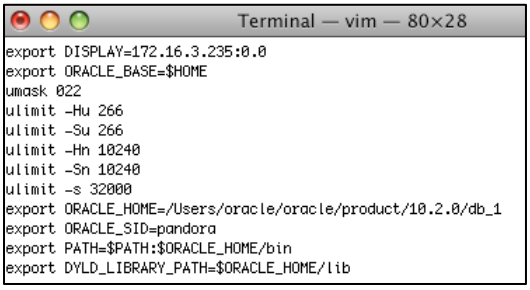


Figura 88 MODIFICACIÓN DE BASH_PROFILE

32. Ejecutamos el archivo con . ./.bash_profile.

33. Cambiamos de directorio cd $ORACLE_HOME/bin que es donde se encuentran todos los archivos ejecutables de Oracle e ingresamos el comando. /netca y se debe levantar el asistente de configuración de Red, en el anexo 1 se muestra el Asistente de Configuración de Red.
34. Ingresamos el comando ./dbca para levantar el asistente de configuración de la base de datos, este creara archivos como redologs, controlfiles y datafiles. En el anexo 1 se muestra el Asistente de configuración de la base de datos.
35. Levantar la base datos.
36. Logearse como usuario oracle su – oracle.
37. Ejecutamos el archivo bash_profile que guarda la configuración de las variables de entorno de la base de datos.
    - . ./.bash_profile.
39. Cambiamos de directorio cd $ORACLE_HOME/bin.
40. Ejecutamos ./sqlplus /nolog para no logearnos.
41. Nos conectamos como usuario sysdba o usuario system en caso que se desee conectarse como usuario system se debe poner connect system/contraseña-establecida.
42. Levantamos la base con el comando startup.

```
labecc:~ oracle$ . ./.bash_profile
labecc:~ oracle$ cd $ORACLE_HOME/bin
labecc:bin oracle$ ./sqlplus /nolog

SQL*Plus: Release 10.2.0.4.0 - Production on Wed Apr 7 10:49:53 2010

Copyright (c) 1982, 2007, Oracle.  All Rights Reserved.

SQL> conn /as sysdba
Connected to an idle instance.
SQL>
SQL> startup
ORACLE instance started.

Total System Global Area  645922816 bytes
Fixed Size                  2086128 bytes
Variable Size             251661072 bytes
Database Buffers          385875968 bytes
Redo Buffers                6299648 bytes
Database mounted.
Database opened.
SQL>
```

Figura 89 LEVANTAR INSTANCIA DE ORACLE

43. La base de datos Oracle esta lista para usarse para ver el estado de la base de datos se puede consultar la vista V$instance:
    - Select status, database_status from v$instance;

ANEXO 5: CONFIGURACIÓN DEL ENTORNO SEMÁNTICO EN MAC OS X SERVER 10.5.4

Los paquetes SDO_RDF son los que permiten trabajar con tripletas RDF dentro de Oracle. Esto paquetes van a permitir crear tablas donde van a ir las tripletas y el modelo al cual van a pertenecer.

PASOS:

1. Debemos crear un Tablespace para los usuarios RDF pero antes nos conectamos como sys y con el rol de sysdba con el siguiente comando conn /as sysdba y creamos el tablespace con la siguiente instrucción:

```
SQL> CREATE TABLESPACE rdf_tblspace DATAFILE '/Users/oracle/oradata/pandora/rdf_
tblspace.dat' SIZE 1024M REUSE AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED SEGMENT
 SPACE MANAGEMENT AUTO;

Tablespace created.

SQL>
```

Figura 90 CREACIÓN DEL TABLESPACE

2. Creamos una red semántica para habilitar el manejo de los datos, El procedimiento SDO_RDF.CREATE_RDF_NETWORK va ser el encargado de dar soporte RDF a nuestra base de datos en Oracle. Este procedimiento es el encargado de crear las tablas y objetos necesarios para trabajar con RDF dentro de Oracle.

```
SQL> EXECUTE SDO_RDF.CREATE_RDF_NETWORK('rdf_tblspace');

PL/SQL procedure successfully completed.
```

Figura 91 EJECUCIÓN DE PROCEDIMIENTO

3. Creamos un usuario ya que Oracle nos recomienda no trabajar bajo el usuario sys o system ya que podemos modificar o dañar algún componente de Oracle.

```
SQL> CREATE USER LEO IDENTIFIED BY oracle DEFAULT TABLESPACE USERS TEMPORARY TAB
LESPACE TEMP QUOTA 256M ON USERS;

User created.
```

Figura 92 CREACIÓN DE USUARIOS

4. A este nuevo usuario se le dan los respectivos privilegios para poder ejecutar sentencias como Insert, update, delete, etc:

```
SQL> GRANT EXECUTE ON MDSYS.RDF_APIS_INTERNAL TO LEO;

Grant succeeded.
```

```
SQL> GRANT CREATE SESSION, DBA, CREATE TABLE TO LEO;

Grant succeeded.
```

Figura 93 OTORGACIÓN DE PRIVILEGIOS

5. Nos Conectamos con el nuevo usuario que hemos creado.
6. Creamos una tabla para almacenar referencias a los datos RDF.

```
SQL> CREATE TABLE FAMILY_RDF_DATA (id NUMBER, triple SDO_RDF_TRIPLE_S);

Table created.
```

Figura 94 CREACIÓN DE OBJETOS PARA RDF

7. Creamos un modelo RDF.

```
SQL> EXECUTE SDO_RDF.CREATE_RDF_MODEL('FAMILY', 'FAMILY_RDF_DATA', 'TRIPLE');

PL/SQL procedure successfully completed.
```

Figura 95 CREACIÓN DE MODELO RDF

8. Insertamos datos.

```
SQL> INSERT INTO family_rdf_data VALUES (1,
  2  SDO_RDF_TRIPLE_S('family',
  3  'http://www.example.org/family/John',
  4  'http://www.example.org/family/fatherOf',
  5  'http://www.example.org/family/Suzie'));

1 row created.
```

Figura 96 INSERCIÓN DE TRIPLETAS

9. Ejemplos.

```
SQL> BEGIN
  2  SDO_RDF_INFERENCE.CREATE_RULES_INDEX(
  3  'rdfs_rix_family',
  4  SDO_RDF_Models('family'),
  5  SDO_RDF_Rulebases('RDFS'));
  6  END;
  7  /

PL/SQL procedure successfully completed.

SQL> SELECT m
  2  FROM TABLE(SDO_RDF_MATCH(
  3  '(?m rdf:type :Male)',
  4  SDO_RDF_Models('family'),
  5  null,
  6  SDO_RDF_Aliases(SDO_RDF_Alias('','http://www.example.org/family/')),
  7  null));

M
--------------------------------------------------------------------------------
http://www.example.org/family/Jack
http://www.example.org/family/Tom

SQL>
```

Figura 97 CONSULTA DE DATOS RDF

ANEXO 6: INSTALACIÓN DE VIRTUOSO SOBRE OPEN-SOURCE EN UBUNTU 10.4

Para Instalar esta herramienta debemos tener instalado los siguientes paquetes en nuestro equipo, las versiones son importantes aunque yo lo he instalado con los paquetes en su versión más nueva y no me dio ningún problema.

Package	Version	From
autoconf	2.57	http://www.gnu.org/software/autoconf/
automake	1.9	http://www.gnu.org/software/automake/
libtool	1.5.16	http://www.gnu.org/software/libtool/
flex	2.5.33	http://www.gnu.org/software/non-gnu/flex/
bison	2.3	http://www.gnu.org/software/bison/
gperf	2.7.2	http://www.gnu.org/software/gperf/
gawk	3.1.1	http://www.gnu.org/software/gawk/
m4	1.4.1	http://www.gnu.org/software/m4/
make	3.79.1	http://www.gnu.org/software/make/
OpenSSL	0.9.7i	http://www.openssl.org/

Figura 98 REQUERIMIENTOS PREVIOS

1. Instalar los paquetes con el comando apt-get install [nombre del paqute] por ejemplo.

```
apt-get install libtool
```

Figura 99 EJEMPLO PARA INSTALAR UN PAQUETE

2. Descargar virtuoso open-source y descomprimirlo.
3. Ubicarse en la ruta donde está el virtuoso descomprimido y ejecutamos el archivo autogen.sh

```
./autogen.sh
```

Figura 100 EJECUCIÓN DE AUTOGEN.SH

4. Establecer variables de entorno con los siguientes comandos, dependiendo de la arquitectura del computador por ejemplo si es de 64 se hace lo siguiente.

```
CFLAGS="-0.2 -m64"
export CFLAGS
```

Figura 101 ESTABLECIMIENTO DE VARIABLES DE ENTORNO

5. Ejecutar el archivo configure con:

- ./configure y luego ejecutamos el comando
- make
- make install

6. Se acepta valores por defecto y con esto concluye la instalación. Para ejecutarlo se realiza lo siguiente:
   - cd /usr/local/virtuoso-opensource/var/lib/virtuoso/db
7. Ejecutamos la siguiente línea:
   - /usr/local/virtuoso-opensource/bin/virtuoso-t -f
8. En un navegador ingresamos con localhost y con el puerto 8890.

Figura 102 OPEN LINK VIRTUOSO

ANEXO 7: INSTALACIÓN DE ALLEGRO GRAPH SOBRE UBUNTU 11.4

Instalación De La Máquina Virtual

1. Descargar la Imagen de la Máquina Virtual de Allegro Graph (2GB).
2. Descargar e Instalar el VMware Player dependiendo para Mac, Windows o Linux
3. Ejecutar el VMware Player, la versión 3.1 está habilitada, sin embargo yo he instalado la versión 2.1[89]
4. Haga doble clic en **VMware-Fusion-*x.x.x-xxxxxx*.dmg** o **VMware-Fusion-*x.x.x-xxxxxx*-light.dmg** para montarlo.
5. Haga doble clic en el icono **Instalar VMware Fusion** para abrir el Asistente de instalación y siga las instrucciones.

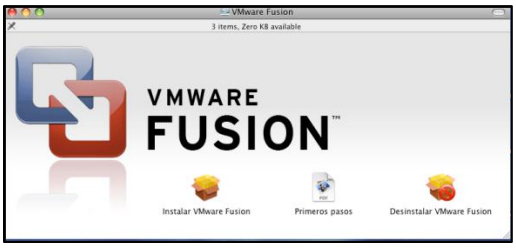


Figura 103 VMWARE INSTALL

6. Introduzca su número de serie. Si no sabe el número de serie, haga clic en **Obtener número de serie** para ir al sitio web de VMware, donde podrá adquirir un número de serie VMwareFusion.
7. Ejecute la VMware Fusion, por defecto esta en Aplications y Click on "Open a Virtual Machine" navegamos hasta donde descomprimimos el archivo y buscamos AllegroGraph v4 Virtual Machine.vmx.
8. Inicie la máquina virtual, cuando le pregunte si fue movido o copiado, seleccione copiado.

[89] http://downloads.vmware.com/d/details/fus_310/ZGpkYmQqQGRiZHR0cA

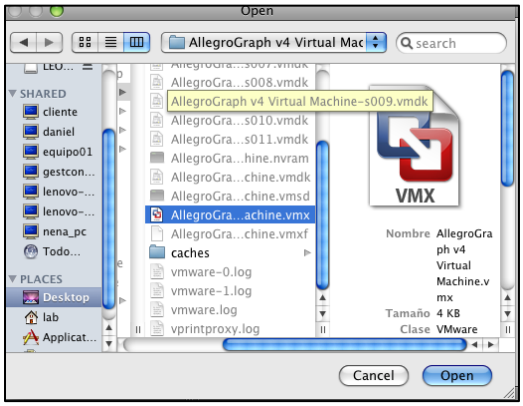


Figura 104 MONTAR ALLEGRO GRAPH

9. Logeese en la máquina virtual de Linux
   - Username: franz
   - Password: allegrograph

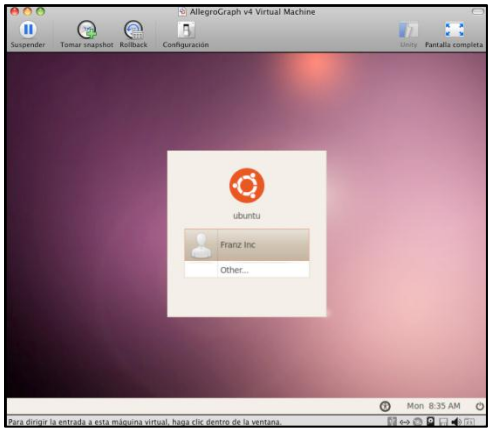


Figura 105 ALLEGRO GRAPH VIRTUAL MACHINE

10. Para iniciar AllegroGraph dar doble click sobre agstart que se encuentra el Desktop y seleccionamos Run in Terminal Window", otra alternativa es abrir una ventana de terminal y tipear agstart.
11. Ejecutar Firefox y click en AGWebView" en la barra de tareas o ingrese http://localhost:10035
12. Logearse en AllegroGraph
    - Username: test
    - Password: xyzzy

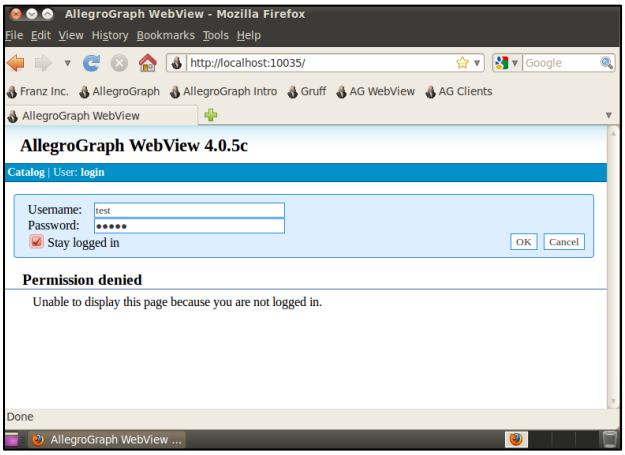


Figura 106 ALLEGRO GRAPH WEBVIEW

13. Para detener AllegroGraph dar doble click sobre agstop que se encuentra el Desktop y seleccionamos Run in Terminal Window", otra alternativa es abrir una ventana de terminal y tipear agstop.

AllegroGraph Virtual Machine

14. Ejecutar Firefox y click en AGWebView" en la barra de tareas o ingrese http://localhost:10035
15. Seleccionamos File>New Triple Store y aceptamos el Puerto por defecto 10035.

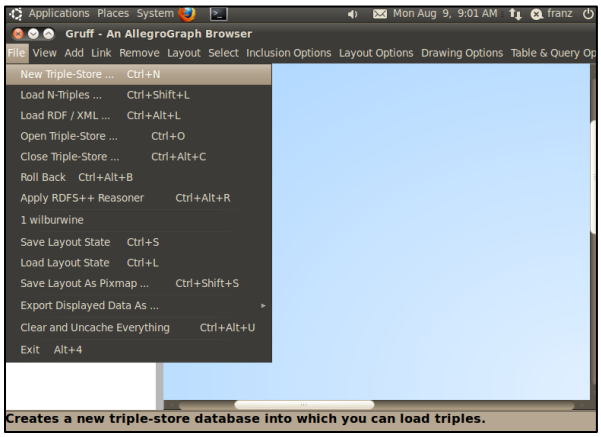


Figura 107 CREACIÓN DE UN TRIPLE STORE

16. Especificamos algún nombre
17. Seleccionamos menu File>Load>N-Triples y seleccionamos uno de los archivos de Home>Franz>AG Triple Data y elegimos Abrir.

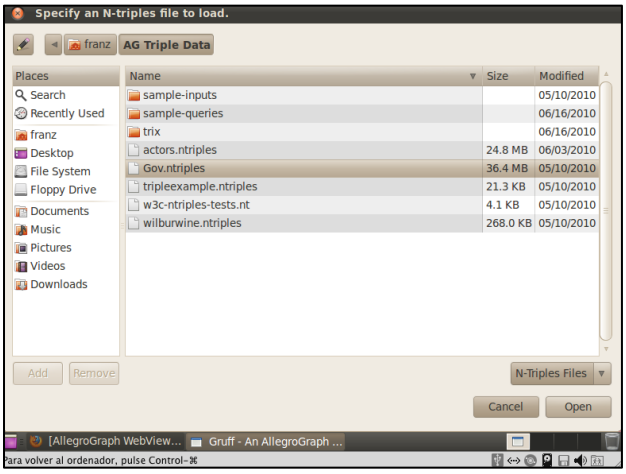


Figura 108 ABRIR AG TRIPLE DATA

18. Esto creara un store y cargara los datos RDF seleccionados, ahora se pueden consultar estos datos.
19. Se puede crear usuarios y asignarles privilegios.

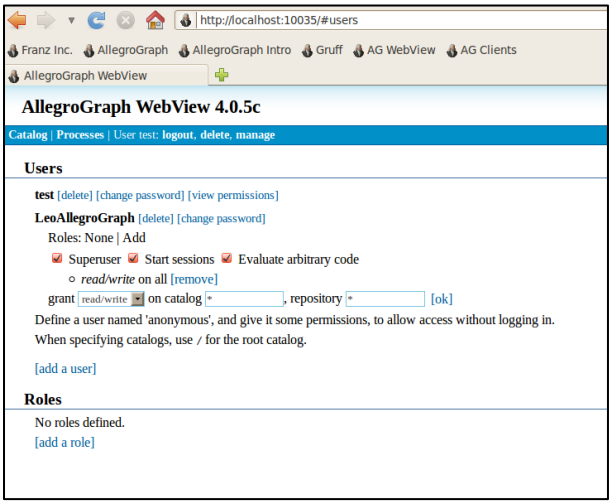


Figura 109 CREACIÓN DE USUARIOS Y ROLES

20. Podemos crear repositorios.

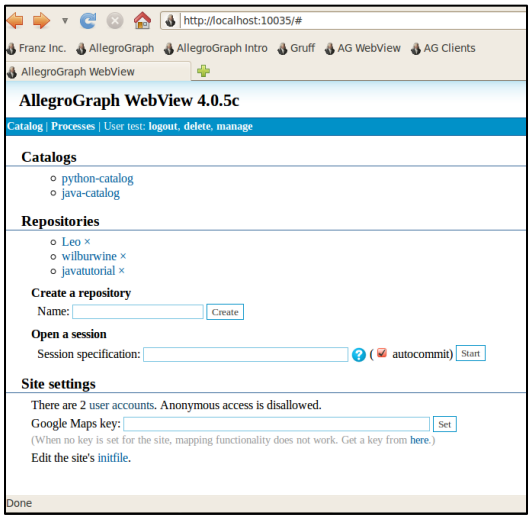


Figura 110 CREACIÓN DE REPOSITORIOS

21. Podemos subir cargar archivos

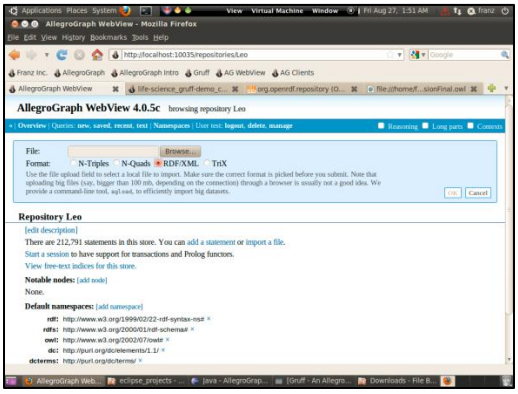


Figura 111 LOAD FILES

22. Realización de consultas utilizando SPARQL.

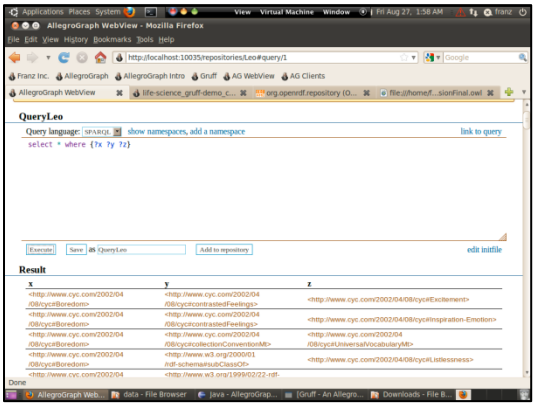


Figura 112 CONSULTAS SPARQL

Eclipse para Java y Jena

23. Asegurarse que AllegroGraph ha sido inicializado
24. Abrir Eclipse desde el Desktop, es importante mencionar que Allegro Graph cuenta con excelentes tutoriales para programar con Jena y Java.

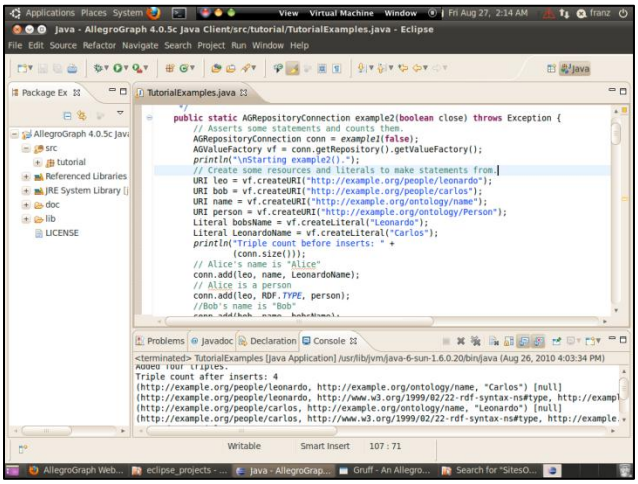


Figura 113 ECLIPSE (JAVA Y JENA)

ANEXO 8: INSTALACIÓN DE 4STORE SOBRE UBUNTU 10.4 32 bits

Pasos para la Instalación:

1. 4store requiere algunas librerías[90] para trabajar entre esas están:
   - raptor (4store 1.0.x requiere raptor 1, y rasqal 0.9.21 o una versión más reciente)
   - rasqal

   Las demás librerías dependen de la distribución en la que se quiera instalar, en nuestro caso es la distribución de Ubuntu. En /etc/apt/Sources.list pegamos las siguientes líneas para poder descargar los paquetes necesarios.

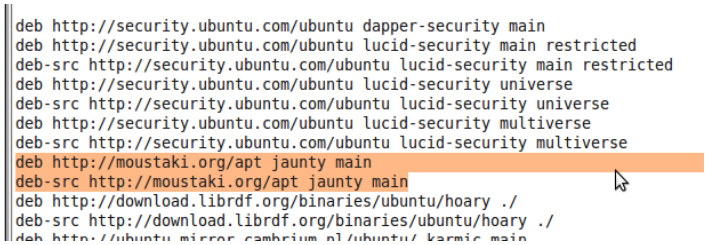


Figura 114 CONFIGURACIÓN DE SOURCES.LIST

2. En un terminal ejecutamos **apt-get update y apt-get install**, esto instalara los paquetes necesarios si tienes algún problema con la instalación de algunos paquetes, estos se pueden descargar manualmente con **apt-get install build-essential libpcre3-dev librasqal2-dev libraptor1-dev libglib2.0-dev ncurses-dev libreadline-dev**. En mi caso tuve problemas para instalar varias librerías, para lo cual las instale manualmente descargando los paquete desde:

   - http://packages.ubuntu.com/karmic/i386/librasqal1/download
   - http://packages.ubuntu.com/da/natty/libdevel/
   - http://security.ubuntu.com/ubuntu/pool/main/p/pcre3/

   Si existen problemas con el **gtk-doc**, se debe instalar manualmente, es importante que también se instalen las dependencias del **gtk-doc**[91], la mayoría de librerías que instale manualmente se las instala con **./configure --prefix=/usr make y make install**.

[90] http://4store.org/trac/wiki/Dependencies
[91] http://www.linuxfromscratch.org/blfs/view/svn/general/gtk-doc.html

3. Instalar **raptor2-2.0.2,**

   - **cd …./raptor2-2.0.2/**
   - **./autogen.sh** (verifica que todas las dependencias estén instaladas y sus versiones. Si existe algún error de versión o de que falta alguna librería se debe buscarla e instalarla manualmente, solo si las dependencias son satisfechas, se podrá ejecutar el comando **make**)

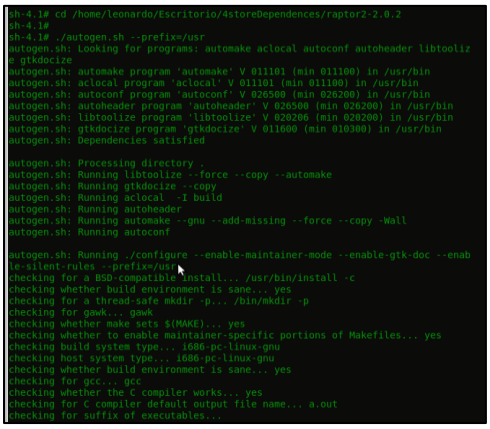


Figura 115 CONFIGURACIÓN DE INSTALACIÓN DE RAPTOR

4. Ejecutar los siguientes comandos:

   - ./configure --prefix=/usr (autogen.sh ya lo hace automáticamente)

Figura 116 INSTALACIÓN DE RAPTOR

5. Ejecutar los siguientes comandos:

   - make
   - make install

6. Instalar rasqal-0.9.25

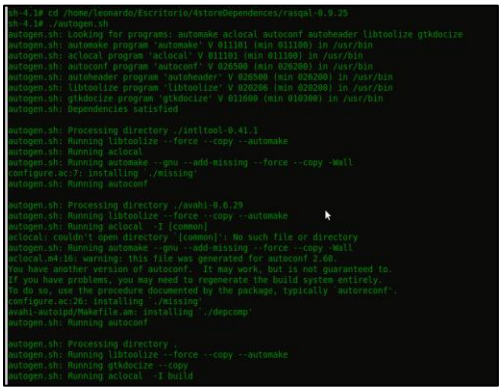


Figura 117 INSTALACIÓN DE RASQAL

7. Ejecutar los siguientes comandos
    - ./configure --prefix=/usr (autogen.sh ya lo hace automáticamente)
    - make
    - make install
8. Instalar 4store, se lo puede hacer utilizando el comando **apt-get install 4store,** pero esto instala la versión 2.0, o también se lo puede descargar al archivo y realizar los mismos pasos que se utilizó para instalar **rapptor** y **rasqal** esta es la forma recomendada.
9. Para crear una base de datos se puede seguir la documentación[92].

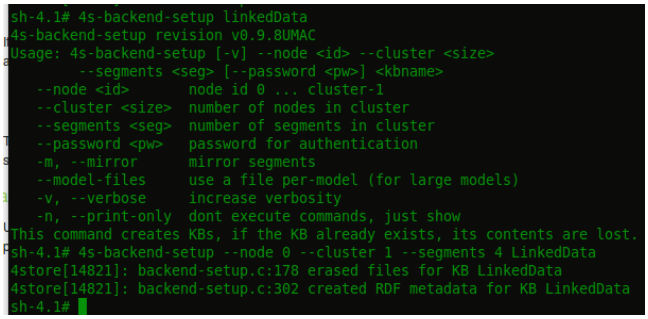


Figura 118 CREACIÓN DE UNA BASE DE DATOS CON 4STORE

La ubicación física de los archivos los podemos encontrar en **/var/lib/4store**

[92] http://4store.org/trac/wiki/Documentation

Nota: La instalación de 4store sobre Ubuntu de 32 bits es mucho más complicada que en la versión de 64 bits, por esta razón se documentó la instalación sobre esta arquitectura, aunque también fue testeada sobre Ubuntu de 64 bits.

ANEXO 9: INCORPORAR LIBRERÍA DE JENA A NETBEANS

Pasos:

1. Descargar el paquete de Jena[93]
2. Extraer los archivos dentro de un directorio.
3. Se crea una biblioteca con las librerías de Jena para ello en NetBeans Herramientas-Bibliotecas- Agregar Archivo JAR/Carpeta

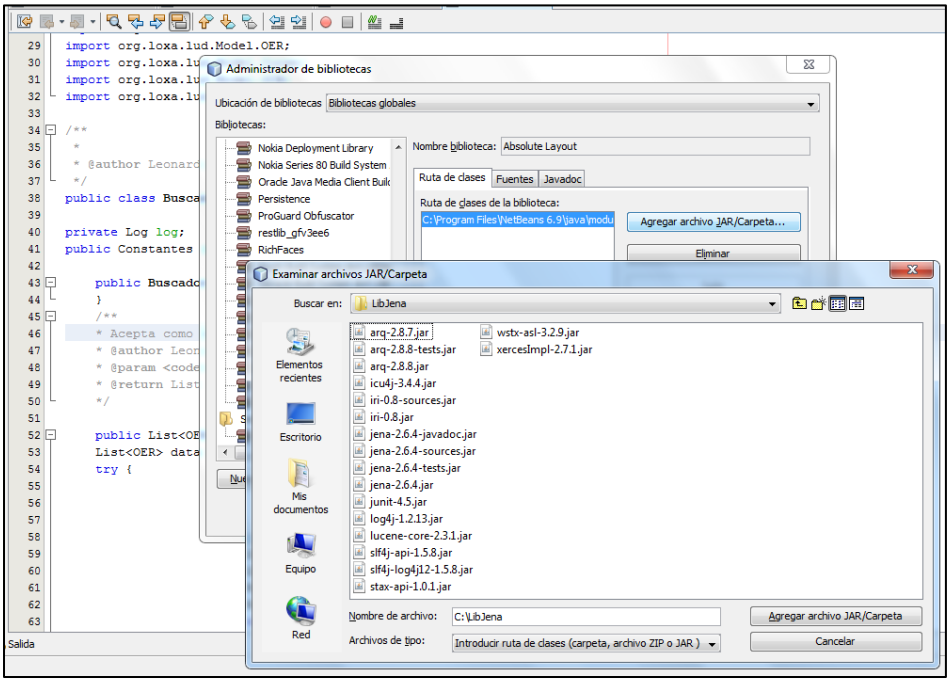


Figura 119 CREACIÓN DE BIBLIOTECA JENA

4. Añadir biblioteca Jena a un proyecto, para ello en un proyecto previamente creado click derecho en librerías y agregar biblioteca y escogemos la biblioteca creada de Jena.

[93] http://jena.sourceforge.net/downloads.html

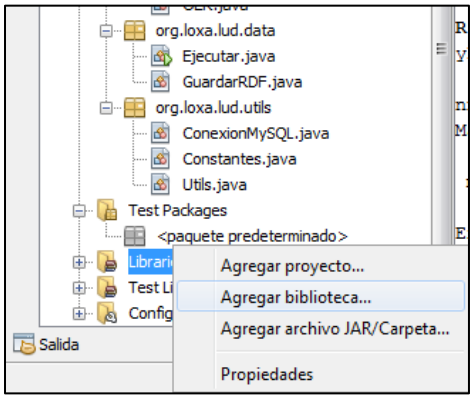


Figura 120 AGREGAR BIBLIOTECA JENA

ANEXO 10: PAPER CIENTÍFICO

Publicación de datos universitarios observando los principios de Linked Data

Universidad Técnica Particular de Loja

Ing. Nelson Piedra
Instituto de Investigacion de Ciencias de la Computacion-UTPL
Loja, Ecuador
nopiedra@utpl.edu.ec

Leonardo Sarango Romero
Instituto de Investigacion de Ciencias de la Computacion-UTPL
Loja-Ecuador
dlsarango@utpl.edu.ec

Resumen—**La web ha evolucionado rápidamente desde su creación hace 15 años aproximadamente, evolucionando desde páginas de texto estáticas, que permitían únicamente la obtención de información unidireccional, pasando por una web social o web 2.0, a una web semántica caracterizada por que la información tiene un significado bien definido, pero la evolución no se detiene en la web semántica debido a que no es suficiente con publicar datos en la web, si no lograr que estos estén vinculados a otros, en lo que se conoce como web de datos.**

La web de datos se ha convertido en un potencial considerable para varios sectores entre ellos el sector educativo, permitiendo que los datos educativos se conecten con más datos relacionados. Entornos de información interconectados, producirán ambientes de aprendizaje más ricos y el aumento de oportunidades para estudiantes y docentes, convirtiéndose la web de datos en una poderosa herramienta para el aprendizaje independiente.

El presente proyecto tiene por objetivo lograr que los dataset de los OCW (OpenCourseWare) de las principales Universidades de Iberoamérica y los dataset de los OER(Open Educational Resource) estén disponibles como RDF, tomando en cuenta los principios de Linked Data delineados por (Tim Berners-Lee, 2006), con el objetivo de mejorar la experiencia de uso de los OCW y OERs, analizando para ello las principales tecnologías semánticas, componentes, estándares, motores de representación RDF y herramientas de visualización y consumo.

Keywords-component: *Semántica, RDF, metadatos, ontología, RDF Schema, Web Semántica, Linked Data, SPARQL, Anotación semántica, URI, HTML, DBPEDIA, XML, HTTP, Motores de*

Representación de RDF, Jena, Sesame, SPARQL end point. Java, Prefix, Tripletas, Recursos, Propiedades y Objetos.

I. INTRODUCCIÓN

La cantidad de datos vinculados publicados en la web ha experimentado un gran crecimiento, y la lista de recursos disponibles en Linked Data crece día a día, sin embargo la mayoría de datos se encuentran sueltos en la web convirtiéndose en datos relativamente pobres, controlados por aplicaciones, es por esto que los Datos enlazados es la forma que tiene la web semántica para vincular todos estos datos sueltos en la web, de forma que se referencien como lo hacen los enlaces de las páginas web con el objetivo de facilitar la búsqueda de información, ofreciendo de esta manera una mejor accesibilidad y brindar mayor significado a los datos.

Actualmente en la web de datos encontramos, alguna información vinculada relacionada a OCWs, esta información puede ser enlazada y consumida desde una fuente de datos basada en OERs (Open Educational Resource) y OCW, añadiendo de esta manera conocimiento y contexto a los OERs.

La solución que se plantea es contar con un Dataset rico, abierto e integrado de RDF de las principales organizaciones OCWs de Iberoamérica, tomando como clave la desambiguación de los OERs, con identificadores únicos en este caso URIs, a través de un vocabulario consensuado, teniendo como resultado final la obtención de una serie de metadatos involucrados a un curso OCW en particular.

Los resultados obtenidos permitieron corroborar los objetivos del presente trabajo, por consiguiente se establece que la utilización de los principios de Linked Data pueden ser usados para publicar datos enlazados que sirvan a estudiantes y profesores a auto educarse, además se demuestra que Linked Data permite la interoperabilidad entre sitios OCW heterogéneos. Para finalizar también se pudo establecer algunos trabajos futuros que pueden partir a raíz de este trabajo, dejando claro en que aplicaciones puede ser factible esta tendencia.

II. LINKED DATA

TimBerness-Lee en su artículo DesingIssues (Berners-Lee, 2006), menciona que la web semántica no solo significa poner datos en la web, también implica enlazar datos, para que una persona o una máquina pueda explorar la web de datos, permitiendo encontrar datos relacionados, por ejemplo si una organización publica datos como su nombre, tamaño, área, etc. Es posible que desee indicar información acerca de su localización, gracias a que en la web existen bases de datos geográficas, la organización puede hacer referencia a los datos geográficos de esa fuente externa, enriqueciendo de esta manera sus datos.

i. Principios de Linked Data

En el mismo artículo (Berners-Lee, 2006), se delinean 4 principios para la publicación de datos en la web.

1. *Utilizar URIs (UniformResourceIdentifier) como nombres únicos para los recursos:* Si se utiliza los URIs para nombrar conceptos se consigue una abstracción del lenguaje natural y evitamos ambigüedades, por ejemplo si deseamos referirnos a un país por su topónimo este puede variar en función de su representación e idioma (EEUU, United States, United States of América, Estados Unidos).
2. *Utilizar el protocolo HTTP para nombrar y resolver la ubicación de los datos identificados mediante esas URIs:* los URIs no son solo direcciones son identificadores de los recursos, por ejemplo http://dbpedia.org/page/United_States.
3. *Representar los datos en RDF y utilizar SPARQL, como lenguaje de consulta de dichos datos:* URIs y HTTP son complementadas por el uso del estándar RDF, una tecnología que es crítica en la web de datos bebido a que provee un modelo genérico de grafos basado en datos para estructurar y vincular datos que describan cosas en el mundo (Bizer, et al, 2008).
4. *Incluir enlaces a otras URIs para permitir la localización de más datos enlazados:* los datos no deben quedar aislados, se deben enlazar a fuentes externas.

ii. Publicación de Datos con RDF

Existen varios mecanismos para publicar datos vinculados en la web, mecanismos que tienen en común el uso de RDF, esto debido a que RDF provee un modelo de datos que es extremadamente simple por un lado, pero a la vez es estrictamente dura hacia la arquitectura de la Web Semántica, permitiendo así que los usuarios puedan buscar todos los URIs en un

grafo RDF a través del Internet para obtener información adicional (Bizer, et al, 2008).

En el artículo mundo (Bizer, et al, 2008), se mencionan los principios básicos de Linked Data para la publicación de datos:

1. Asignar URIs a las entidades descritas por el conjunto de datos.
2. Usar link RDF para interconectar datos desde diferentes fuentes de datos en la web.
3. Proporcionar metadatos sobre los datos publicados.

En (Corcho & Gómez, 2010), se menciona que el proceso de publicación de datos enlazados en la web incluye los pasos citados a continuación:

1. *Identificación de fuentes de datos.*
2. *Desarrollo de vocabularios.*
3. *Generación de datos RDF.*
4. *Publicación de datos RDF.*
5. *Desambiguación y cleaning de datos.*
6. *Enlazar datos RDF con otras fuentes en la Web de datos.*
7. *Habilitar un descubrimiento efectivo.*

iii. Tecnologías

Existen varias tecnologías que son muy útiles a la hora de vincular y publicar datos entre las más importantes se mencionan las siguientes.

- *URIS*: Una URI puede identificar cualquier cosa, en otras palabras provee un simple y un extensible significado para identificar un recurso. (Berners-Lee, et al, 2005).
- *RDFS*: Extensión semántica de RDF que consiste en un lenguaje primitivo de ontologías que proporciona los elementos básicos para la descripción de vocabularios (Brickley & Libby, 2010).
- *RDF*: (Marco de Descripción de Recursos) es el estándar de la W3C para los datos interoperables legibles por máquinas.
- *SPARQL*: Es una tecnología que hace posible la recuperación de RDF y RDFS como lo mencionan (W3C, 2007), esto debido a que se basa en la comparación de patrones gráficos los cuales están basados en tripletas como las usadas en RDF.
- *Vocabulario para Linked Data*: permiten la descripción de datos los cuales deben reutilizarse y no reinventar, para garantizar que los datos puedan ser consumidos por aplicaciones en la web facilitando la interoperabilidad, entre los más importantes y utilizados en el desarrollo de esta investigación se mencionan los siguientes:
  - FOAF
  - DUBLIN CORE
  - AIISO
  - XSD
  - CCREAL
- *Motores de Representación (RDF Data Store):* Actualmente existen varias herramientas que se utilizan como repositorios de almacenamiento para RDF llamados RDF Triple Store System según la (W3C, 2011), son sistemas que nos permiten almacenar e interactuar con tripletas, entre los más importantes y los que se han instalado, configurado y usado se mencionan los siguientes:
  - 4Store
  - OpenLinkVirtuoso
  - Sesame
  - Oracle Spatial 11g
  - AllegroGraph

En la tabla I se realiza una comparativa entre los RDF triple Store System.

III. DESARROLLO

i. Identificación y Selección de Fuentes de Datos.

La identificación y selección de fuentes de datos, obedece al primer paso que se propone en (Corcho & Gómez, 2010). En esta investigación las fuentes de datos son básicamente metadatos de OCW (OpenCourseWare) y OER (Open Educational Resource).

Una cantidad significativa de información está expuesta en páginas web públicas de varias instituciones OCW alrededor de todo el mundo, esta información puede ser publicada en RDF como Linked Data, término que en este contexto se convierte en la mejor practica para exponer, compartir, conectar e interoperar con OCW/OER, ayudando a usuarios e instituciones a romper barreras limitantes de la web tradicional; sin embargo el problema radica en que estos datos no están enlazados y su recuperación es bastante tediosa, esto debido a la heterogeneidad de los diferentes repositorios OCW. Sin embargo hasta el momento el movimiento Open Linked Data ha realizado un excelente trabajo y ha demostrado que la publicación de datos enlazados abiertos en la web es técnicamente posible, esto según lo mencionado en (Shabir & Clarke, 2009).

El proceso de obtención de datos se lo presenta en la siguiente Figura y se simplifica en los siguientes pasos nombrados a continuación.

1. Los metadatos obtenidos fueron de las páginas principales de varios cursos OCW, debido a que en la presentación de un curso, encontramos metadatos como autores, título del curso, fecha de creación del curso, objetivos, área de conocimiento entre otros, metadatos.
2. Almacenamiento de datos en procesadores de texto
3. Desambiguación de datos.
4. Procesamiento de datos con una aplicación construida para tal propósito utilizando la librería POI[94].

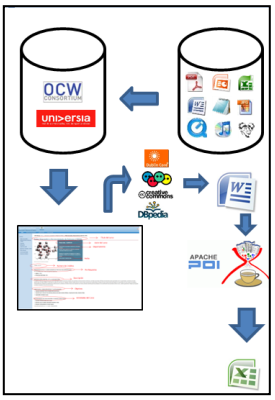


Figura 121 Proceso de obtención de datos

ii. Descripción de Vocabularios.

La descripción de vocabularios, obedece al tercer paso para la publicación de datos en la web, según lo mencionado en (Corcho & Gómez, 2010), en el proceso de publicación de datos enlazados en la web, además es una recomendación según se lo menciona en (Healh & Bizer, 2011). Se reutilizo vocabularios consensuados como FOAF (para describir los autores de los OER), Dublin Core (para describir las propiedades genéricas de un OER), AIISO (para describir las propiedades de las instituciones OCW) y Creative Commons (para describir el tipo de licencias asociadas a un determinado OER). En pocas palabras en base a las fuentes de datos seleccionadas, se reutilizó vocabularios existentes que se adapten a las necesidades del proyecto para posteriormente poder generar el RDFS y el RDF.

Según lo mencionado en (Healh & Bizer, 2011), en un ambiente de Linked Data frecuentemente es suficiente expresar vocabularios en RDFS, aunque también es frecuente la utilización de RDF++, una combinación de RDFS y ciertas primitivas de OWL como sameAs, para afirmar que dos URIs identifican a un mismo recurso.

[94] http://poi.apache.org/

Es importante mencionar que debido, a que no todos los vocabularios, se adaptan a las necesidades de nuestro dominio, fue necesario la creación un vocabulario hibrido que nos permita, contar con un esquema de RDF para, posteriormente en base a este esquema poder generar el RDF en base los metadatos obtenidos de OCW/OER La Figura a continuación muestra el modelo relación de lo mencionado.

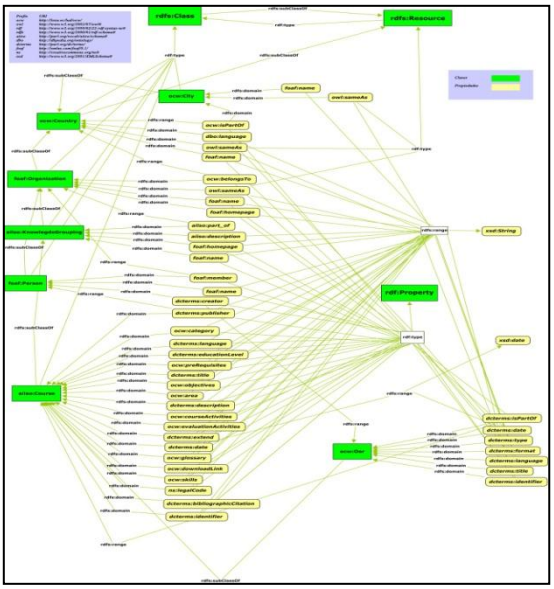


Figura 122 VOCABULARY OCW/OER/LICENCE

iii. Generación de RDFS y RDF

Generación de RDFS y RDF, el tercer paso para el proceso de publicación de datos enlazados en la web según lo mencionado en (Corcho & Gómez, 2010). Los datos son generados a partir de los datos obtenidos y de la reutilización de los vocabularios consensuados. Básicamente se desarrolló una aplicación web en JAVA que permite la generación automática de los datos RDFS y RDF a partir de una cadena de conexión a una base de datos MySQL. La Arquitectura de la aplicación (LOXA VOCABRDFADMIN), se ilustra en la siguiente Figura.

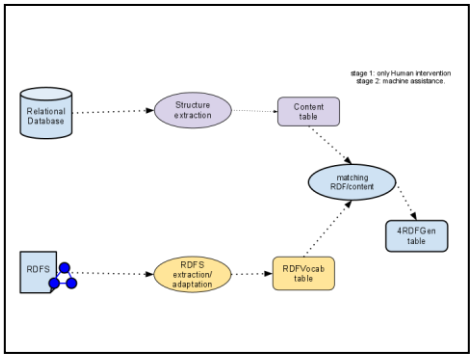


Figura 123 LOXA VOCABRDFADMIN

Para el desarrollo de esta aplicación se utilizó tecnologías como MySQL (repositorio de información), JAVA (lenguaje de programación), RichFaces (framework de diseño) y Jena (manipulación y almacenamiento de grafos RDF).

Jena interviene en la aplicación para usar sus vocabularios ya establecidos y para generar el RDF. Para esto Jena recibe valores recuperados directamente desde la base de datos, los mismos que el usuario va formando interactivamente. Jena almacena los valores como un modelo, este modelo es creado es a su vez por ModelFactory sentencia que crea un modelo por defecto. Esto se lo consigue utilizando la siguiente sentencia. Model model = ModelFactory.createDefaultModel(). Este modelo permite que Jena pueda leer los datos que se le envían para que una vez que el modelo sea poblado Jena pueda realizar un conjunto de operaciones.

Definido y creado el Modelo se trabaja con Recursos (Resource) y Propiedades (Property) que son provistas por unas constantes que se manejan, dichas constantes se procesan en listas y son enviadas a una tabla llamada Vocabulary en la base de datos. Con los datos que retorna

MySQL y los datos recuperados con la interacción del usuario se forman tripletas (Statements).

Una vez que se cuentan con los datos almacenados y cargados en un modelo, resulta sencillo escribir un archivo o crear una tabla en la base de datos con el RDF generado. Para este propósito se utiliza la función: model.write con un formato de abreviatura "RDF/XML-ABBREV".

iv. Resultados

Los resultados obtenidos, son medidos en cuanto al RDF generado con la aplicación, resultados muy satisfactorios para la primera versión de esta aplicación web que se pretende llegué a ser considerada una herramienta muy útil a la hora de Generar RDF. La Figura a continuación muestra la interacción del usuario con la aplicación.

La interfaz principal presenta unos cuadros de texto que tendrán que ser llenados con información correspondiente a la conexión a la base de datos como la IP de la maquina (en caso de ser local será localhost), puerto por el cual se conecta la base de datos por defecto es 3306, usuario de la base de datos, su contraseña y el nombre del modelo o base de datos. Luego de haber llenado el formulario con los datos correspondientes es necesario seleccionar el botón enviar datos, esto levantará una serie de procesos a nivel de código java para interactuar con la base de datos. Posteriormente el usuario podrá formar las tripletas interactivamente, para este proceso se le presenta al usuario una serie de componentes que le permitirán realizar esta tarea, e inclusive el usuario podrá subir su propio vocabulario o cualquier archivo RDF.

Por último Se genera el RDFS a nivel de ID de las tablas únicamente, caso contrario se genera el RDF a nivel de los datos de las columnas de las tablas de la base de datos, y el resultado se presenta en un cuadro de texto.

.

En las Figuras a continuación se ilustran los resultados obtenidos.

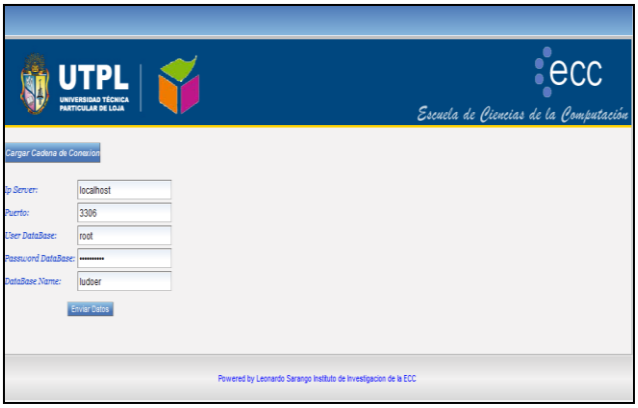


Figura 124 CARGAR ARCHIVO DE CONEXIÓN

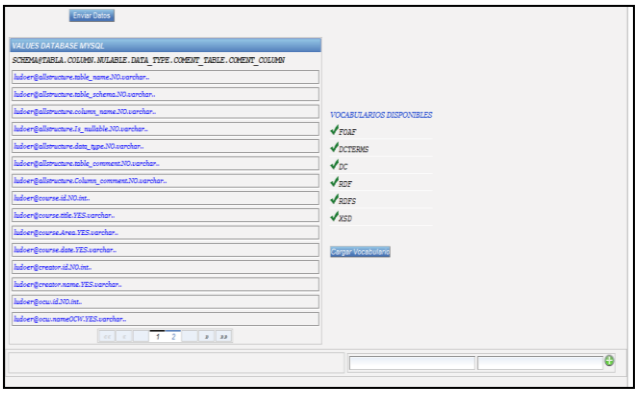


Figura 125 TABLAS RECUPERADAS DE LA BASE DE DATOS

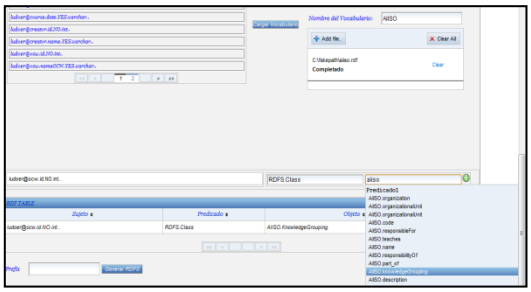


Figura 126 COMPONENTE AUTOCOMPLETE

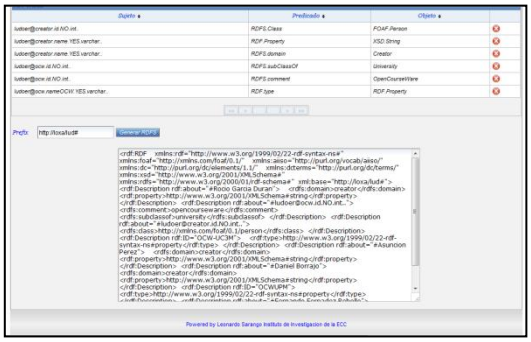


Figura 127 RDFS/RDF GENERADO

Si bien es cierto que el proceso de generación RDFS y RDF es un proceso tedioso, el aplicativo actualmente permite automatizar el proceso de generación. Actualmente en el mercado existen varias herramientas para la generación de RDF, pero no hay una que permita la generación automática a este nivel. No obstante la calidad del RDF dependerá únicamente de la estructura de las fuentes de datos estructuradas, y del conocimiento de los usuarios en temas de datos RDFS y RDF.

v. Implementación de un modelo de visualización para RDF Data del Modelo.

Se implementó un módulo que permite la visualización de datos RDF y búsqueda de Recursos, a partir de los datos generados con la aplicación Loxa VocabRDFAdmin, es importante mencionar que el dominio de la aplicación solamente funciona con un modelo de datos RDF especifico, esto significa que no es una aplicación que permite la visualización de cualquier dato RDF. Es importante mencionar que es muy complicado realizar una aplicación que muestre al nivel aquí presentado datos RDF, debido a que esta depende íntegramente de las consultas SPARQL aplicadas al modelo de datos RDF.

La aplicación permite recuperar y analizar un archivo RDF que contiene un grafo o una colección de grafos, en nuestro dominio los grafos corresponden a los OER, Cursos OCW, Autores y Organizaciones, estos datos son almacenados en una base de datos MySQL, es importante mencionar que Jena almacena directamente los datos en la base de datos mediante un método java que recibe la ubicación del archivo RDF, y el nombre del modelo en la base de datos. Además Se implementó métodos para acceder a los datos almacenados en formas de tripletas por Jena en MySQL, y poder consumirlos utilizando para ello SPARQL. Los resultados obtenidos se muestran en una interfaz web, para ello se utilizó los componentes de RichFaces.

vi. Resultados

Los resultados obtenidos en base a las consultas SPARQL, previamente definidas en el código, fueron satisfactorios y suficientes para demostrar el consumo y enlace de datos. Es importante mencionar que se ha trabajado únicamente con SPARQL, para recuperar la información presentada por el visualizador, y se ha trabajo las cadenas para que muestre datos sin caracteres especiales como "_" o "#", que son muy comunes en las tripletas RDF, esto con el objetivo de presentar una información al usuario totalmente transparente.

En las Figuras a continuación se ilustran los resultados obtenidos.

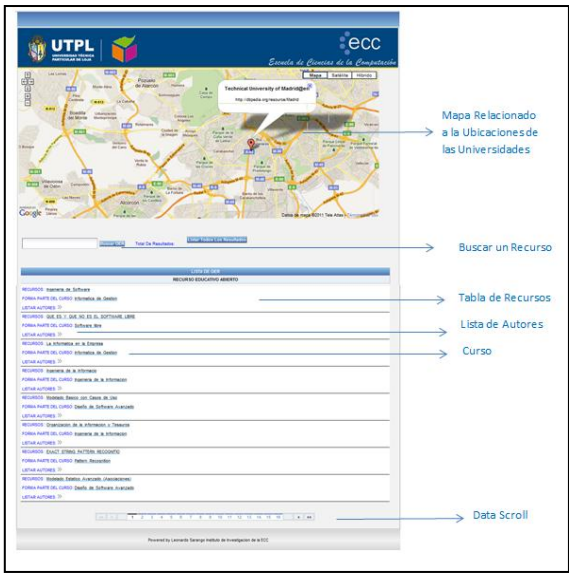


Figura 128 4RDFVISUALIZADOR

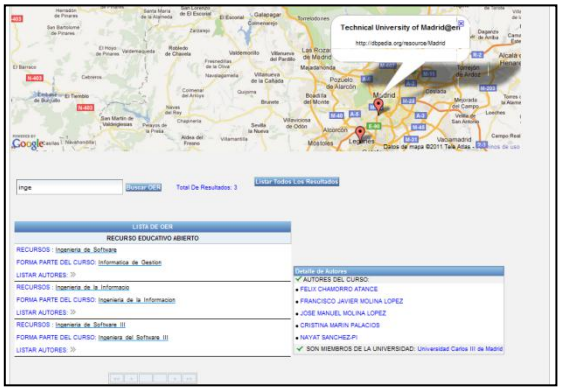


Figura 129 BUSCADOR DE RECURSOS

IV. CONCLUSIONES

- Se ha realizado un trabajo de investigación sobre la Evolución de la web, poniendo énfasis en la publicación de datos enlazados de recursos educativos, demostrando que se es factible la publicación de datos procesados en la web, utilizando para ellos las principales tecnologías, estándares e iniciativas de la web semántica cumpliendo así el objetivo principal del proyecto.

- Como resultado de la investigación sobre almacenes RDF se encontró que muy pocos Gestores RDF soportan la visión de la web Semántica en cuanto a escalabilidad, operatividad en las respuestas y distribución de información; dado que procesan información sin razonar sobre ella. Es muy importante que los gestores RDF incluyan una mayor capacidad de inferencia.

- Se concluye que Jena es el mejor framework actual para trabajar con datos RDF, definiéndolo personalmente como el puente entre el almacén de datos y el Usuario, el mismo que se encuentra sostenido por pilares como Java, SPARQL, RDF, entre otras tecnologías.

- Una de las principales utilidades que se ha obtenido con el desarrollo del proyecto es dar un punto de partida para aprender y experimentar como se puede integrar tecnologías semánticas a base de datos, logrando de esta forma mayores beneficios a futuras investigaciones.

- Se encontró que la generación de un script a partir del modelo conceptual Entidad-Relación se debe hacer tanto para obtener diseños de tablas relacionales como para generar Descripción de Contenidos RDF.

- Con el aplicativo construido se ha demostrado que es factible la generación de RDF desde fuentes de datos estructuradas.

- Se concluye que la tarea más difícil en la visión de enlazar datos constituye la extracción de datos desde la web, en nuestro caso la naturaleza heterogénea de los OERs constituyo una gran barrera.

- Se concluye que la utilización de estándares como RDF, SPARQL, son los más aptos y con los cuales se obtuvo mejores resultados a la hora de acceder a la información.

- El resultado obtenido en esta investigación al generar y utilizar datos enlazados procedentes de fuentes heterogéneas como los OCW, creara nuevas oportunidades de investigación, aunque aún falta mucho camino por recorrer como trabajar en definir reglas que faciliten la homogeneidad entre Data Sources.

- Se demostró que la utilización de los principios de Linked para publicar datos sobre la web, logrará crear un espacio global sobre la Web, lo que contribuirá a incrementar el número de tripletas RDF en la Web.

V. RECOMENDACIONES

- En lo relacionado, los Gestores RDF, se recomienda que antes de elegir algún gestor RDF determinado, se analice a profundidad el dominio del servicio, aplicación o sistema que se dese implementar.

- Para la configuración del entorno semántico en Oracle se deben ejecutar los scripts en el orden en el que se especifican en este trabajo, ya que algunos dependen de otros y en el caso de no hacerlo puede ocurrir un error en dicho proceso.

- En lo relacionado a la descripción de vocabularios se recomienda reutilizar vocabularios consensuados como FOAF, Dublin Core.

- Antes de realizar un RDFS, se debe implementar un modelo de entidad relación con el objetivo de obtener una visión mucho más amplia de lo que se desea implementar.

- Para la validación del RDF se recomienda utilizar el servicio de validación de RDF que ofrece la W3C.

VI. TRABAJOS FUTUROS

- Investigación y aplicación de los razonadores que se pueden utilizar desde el API Jena sobre las tripletas que están almacenadas en una base de datos.

- Consumir dantos enlazados de OERs de las principales universidades del mundo y publicarlas para el uso de los estudiantes de La Universidad Particular de Loja en primera instancia a través de su Entorno de Aprendizaje Virtual (EVA).

- Desarrollar un Buscador web a nivel macro, que permita a los estudiantes y profesores de todo el mundo encontrar cursos relacionados, temas específicos o encontrar áreas de investigación relacionadas.

- Desarrollar en base a los resultados obtenidos, un proyecto a nivel macro que forme parte de la nube de Open Linked Data.

- Desarrollar una aplicación que permita la recuperación automática de metadatos de recursos directamente desde páginas HTML.

- Mejorar la versión 1 del aplicativo RDFLoxaAdmin para poder extraer y procesar información de más fuentes de datos estructuradas, con el objetivo de convertirse en una herramienta útil a la hora de generar RDF.

- Elaborar políticas de publicación de OERs, dirigidas a todos los OCWs, con el objetivo de estandarizar la publicación de Recursos Educativos.

- Contar con un Dataset integrado y disponible como RDF y de acceso público a través de descargas e información relativa, vinculando datos de varios Dataset, con el objetivo de enriquecer la información relacionada a un recurso educativo determinado.

VII. BIBLIOGRAFÍA

1. (Berners-Lee, 2006) Berners-Lee, T. (2006). *W3C*. (W3C) Recuperado el 23 de February de 2011, de Design Issues for Linked Data: http://www.w3.org/DesignIssues/LinkedData.html
2. (Bizer, et al, 2008) Bizer, C.; Cyganiak, R.; Healt, T. (2008). *How to Publish Linked Data on the Web*. Recuperado el 23 de December de 2010, de http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/20070727/
3. (Corcho & Gomez, 2010) Corcho O.; Gómez A. (2010). *Mini-curso sobre Linked Data*. Recuperado el 23 de January de 2011, de http://www.slideshare.net/ocorcho/linked-data-tutorial-florianpolis
4. (Berners-Lee, et al, 2005) Berners-Lee, T.; Masinter, L.; Fielding, R. (2005). *Uniform Resource Identifier (URI): Generic Syntax*. Recuperado el 29 de June de 2011, de http://tools.ietf.org/html/rfc3986
5. (Brickley & Libby, 2010), Brickley, D.; Miller, L. (2010). *FOAF Vocabulary Specification 0.98*. Recuperado el 25 de June de 2011, de http://xmlns.com/foaf/spec/
6. (W3C, 2007) W3C (2007). Recuperado el 22 de December de 2010, de SPARQL Query Language for RDF: http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/
7. (W3C, 2011), W3C. (2011). *SemanticWebTools*. Recuperado el 23 de February de 2011, de http://www.w3.org/wiki/SemanticWebTools
8. (Shabir & Clarke, 2009) Shabir, N.; Clarke, C. (2009). *Using Linked Data as a basis for a Learning Resource Recommendation System.* Recuperado el 10 de March de 2011, de http://eprints.ecs.soton.ac.uk/18053/1/semhe09_submission_5-1.pdf
9. (Healh & Bizer, 2011) Heath, T.; Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Recuperado el 29 de June de 2011, de http://linkeddatabook.com/editions/1.0/